

FEED CAPE FEAR-EMPOWERING COMMUNITIES THROUGH
COMPREHENSIVE ENGAGEMENT AND FOOD DISTRIBUTION

Madhava Reddy Muli

A Capstone Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Congdon School of Supply Chain, Business Analytics, and Information Systems

University of North Carolina Wilmington

2024

Approved by

Advisory Committee

Dr. Curry Guinn
Committee Member 1

Dr. Avinash Geda
Committee Member 2

Dr. Kevin Matthews
Committee Chair

Accepted By

Dean, Graduate School

TABLE OF CONTENTS

	Page
ABSTRACT.....	iv
Chapter 1: Introduction	8
Chapter 2: Review of Literature	13
2.1. Community Engagement and Technology	13
2.2. Food Distribution and Localized Solutions	14
2.3. Corporate Social Responsibility Integration	15
2.4. Existing Solutions and Problems	16
2.4.1. VolunteerMatch.....	16
2.4.2. GivePulse	17
2.4.3. NextDoor.....	18
2.4.4. MealConnect.....	18
Chapter 3: Description of Proposed Solution	20
3.1. Overcoming Existing Challenges	20
3.2. User Interaction	23
3.3. Goals for Feed Cape Fear	28
3.3.1. Localized Platform Development.....	29
3.3.2. Comprehensive Dashboard	29
3.3.3. Real-Time Communication Tools	30
3.3.4. Volunteer Interest Tracking	30
3.3.5. Localized Emergency Response	31
3.3.6. Multi-Stakeholder Collaboration	31
3.4. Target Audiences	31
3.5. Technology Stack	32
3.5.1. Frontend Development.....	33
3.5.2. Backend Development.....	34
3.5.3. Database Management	35
3.5.4. Package Management	37
3.6. Analyzing the Activities and Processes	38
Chapter 4: Methodology	41
4.1. Analysis of use cases.....	42
4.2. Prototype and design.....	44
4.3. Database design.....	45

4.4.	Security	47
Chapter 5:	Development	50
5.1.	Initial Setup for the system.....	50
5.1.1.	Installing Visual Studio Code.....	50
5.1.2.	Setting up React for Front-End Development	51
5.1.3.	Setting up Express for Backend Development.....	51
5.1.4.	Setting up MYSQL for database	52
5.2.	User Registration and authentication.....	52
5.3.	User Dashboard and Emergency/normal posts.....	55
5.4.	Groups/Organization page	59
5.5.	Volunteering opportunities/ events page	61
5.6.	Logout	65
Chapter 6:	Testing.....	66
6.1.	MANUAL testing	66
6.2.	Integration testing	77
6.3.	User Feedback.....	79
Chapter 7:	Deployment	80
7.1.	Initial Setup and preparation	80
7.2.	File transfer and setup	81
7.3.	Dependency Installation and database migration.....	81
7.4.	Challenges faced and solutions.....	82
Chapter 8:	Future works.....	83
Chapter 9:	Lessons Learnt	86
	REFERENCES.....	88
	APPENDIX A: Use of Artificial Intelligence Tools.....	90

ABSTRACT

Feed Cape Fear – Empowering Communities Through Comprehensive Engagement and Food Distribution. Madhava Reddy Muli, 2024. Capstone Paper, University of North Carolina Wilmington.

In the active Cape Fear region where a strong community encounters technology innovation, "Feed Cape Fear" is aimed at developing an online platform for essential food distribution and community connection in the Cape Fear area. This online platform goes beyond the conventional models by combining all three facets of food distribution, multistakeholder collaboration, and volunteerism in a dynamic virtual environment. Powered by the desire to make certain that no person in the community remains without food, this project utilizes geolocation features, real-time communication tools, and intuitive user experience. The idea behind "Feed Cape Fear" is a community where local organizations, corporations and even individuals come together to meet the challenges, providing not only sustenance but also a true sense of home. The platform allows collaboration, communication, and coordination among these stakeholders to meet the supply demand of food resources through integration of corporate social responsibility initiatives as well as providing emergency response capabilities. The platform also improves the user experience and volunteerism as the users can post and view volunteering opportunities using pictures or videos, so that they are visually engaging ways of interacting with the community to make some impact. This abstract gives a glimpse of the strategic development and technological approach of the "Feed Cape Fear," so that the reader would continue to study the comprehensive proposal to

gain a deeper understanding for this transformative initiative which will shape the community involvement and eliminate hunger in the Cape Fear region.

LIST OF TABLES

Table	Page
1. Table 1: Different Models in Software Development Lifecycle (SDLC).....	8
2. Table 2: Brief explanation about different phases of “Feed Cape Fear”	11
3. Table 3: Pros and Cons of existing solutions.....	21
4. Table 4: Questions asked during User Interaction.....	23
5. Table 5: List of activities and their descriptions.....	39
6. Table 6: Use cases description for the User actor.....	41
7. Table 7: Use cases description for the Organization/Admin actor.	41
8. Table 8: Manual testing for user registration and login.....	68
9. Table 9: Manual testing for user dashboard.....	69
10. Table 10: Manual testing for Groups/Organizations page.....	70
11. Table 11: Manual testing for Events page.	72
12. Table 12: Manual testing on admin controls.....	74
13. Table 13: Manual testing on Real-Time Communication.....	75
14. Table 14: Integration testing for Feed Cape Fear.	77

LIST OF FIGURES

Figure	Page
1. Figure 1: Flowchart for different phases of “Feed Cape Fear”	10
2. Figure 2: User Journey Diagram.....	38
3. Figure 3: Use Case Diagram for “Feed Cape Fear”	40
4. Figure 4: User Dashboard prototype	42
5. Figure 5: Timeline Chart for all the phases of project.....	44
6. Figure 6: Entity Relationship Diagram for Feed Cape Fear.....	47
7. Figure 7: User Registration page.....	53
8. Figure 8: User Login page.....	54
9. Figure 9: Admin Dashboard Page.....	55
10. Figure 10: User Dashboard home page.....	57
11. Figure 11: Post Creation window.....	58
12. Figure 12: Groups/Organizations page.....	61
13. Figure 13: Volunteering page showing the create event option.....	63
14. Figure 14: Volunteering/Event Page.....	64
15. Figure 15: Displaying the list of participants interested in the event or opportunity..	65

CHAPTER 1: INTRODUCTION

In the heart of the Cape Fear region, where the pulse of community life is felt in shared experiences and mutual support, the "Feed Cape Fear" project unfolds as a transformative response to critical community needs. This capstone initiative stands at the intersection of addressing hunger, fostering community engagement, and amplifying volunteerism. Recognizing the interconnected nature of these elements, the project endeavors to enhance community well-being by integrating technology, collaboration, essential food distribution, and a dedicated focus on volunteering opportunities.

"Feed Cape Fear" is a pioneering web-based platform designed to bridge the gap between non-profit organizations, corporate entities, and individuals striving to make a difference in their community. Beyond its technological innovation, the project has a central and profound objective: to ensure that no member of the Cape Fear community goes hungry. This proposal outlines the strategic development and implementation plan for "Feed Cape Fear," emphasizing a localized approach to food distribution, multi-stakeholder engagement, and the integration of corporate social responsibility.

This project aligns with the Software Development Lifecycle (SDLC) approach and emphasizes structured and systematic methodology to ensure the successful development of this platform. There are several models in SDLC like waterfall, agile, spiral, and iterative. Unlike the waterfall model, agile, spiral, and iterative follow iterative and incremental development processes, which can provide some flexibility throughout the project lifecycle (GeeksforGeeks, 2024). Table 1 discusses what are these models of SDLC briefly.

Models of SDLC	Description
Waterfall	<ul style="list-style-type: none"> • Linear and sequential approach • Requirement analysis, design, implementation, testing, and maintenance are the phases of this model. • Suitable for projects with well-defined and stable requirements.
Agile	<ul style="list-style-type: none"> • Iterative and incremental • Emphasizes flexibility and adaptability. • Prioritizes working software in short iterations, allowing for rapid response to changes in requirements.
Spiral	<ul style="list-style-type: none"> • Combination of both waterfall and iterative models. • Multiple cycles of prototyping, risk analysis, and development, with each cycle building on the previous one. • Well suited for large and complex projects.
Iterative	<ul style="list-style-type: none"> • Involves repeating cycles of development, testing, and feedback gathering. • Each iteration adds new features or improvements. • Suitable for projects with evolving requirements and uncertain environments.

Table 1: Different Models in Software Development Lifecycle (SDLC).

“Feed Cape Fear” approach is to address specific issues in the Cape Fear region, where requirements are expected to be relatively stable. The involvement of stakeholders early in the project is critical for understanding the needs and the expectations on the “Feed Cape Fear” platform. This project has certain sensitive aspects like emergency response and food distribution which requires a sequential nature for better risk management. The waterfall model was chosen for this project due to its suitability for well-defined requirements, early stakeholder involvement, and effective risk management capabilities. This project is divided based on the various phases of waterfall model as mentioned in Table 1.

Beginning with the requirements gathering phase, this project meticulously identifies and analyzes the specific needs of users within the cape fear region. In this initial phase all the requirements are gathered, examples include discussion among

different stakeholders, conducting user interviews and doing external research. This is followed by the design phase, where comprehensive user interfaces and functionalities are crafted to meet the requirements. Subsequently the development phase involves the actual coding and implementation of the platform, adhering to industry best practices and standards. In the next phase, testing and quality assurance procedures are then conducted to validate the platform's functionality and ensure a seamless user experience. Finally, the deployment phase marks the launch of the platform, and as of now a maintenance plan is yet to be finalized for this platform but there will be continued support to address any user feedback during this procedure. Figure 1 summarizes the SDLC chosen for this project. Table 2 summarizes the types of tasks that will be completed in each phase for the SDLC.

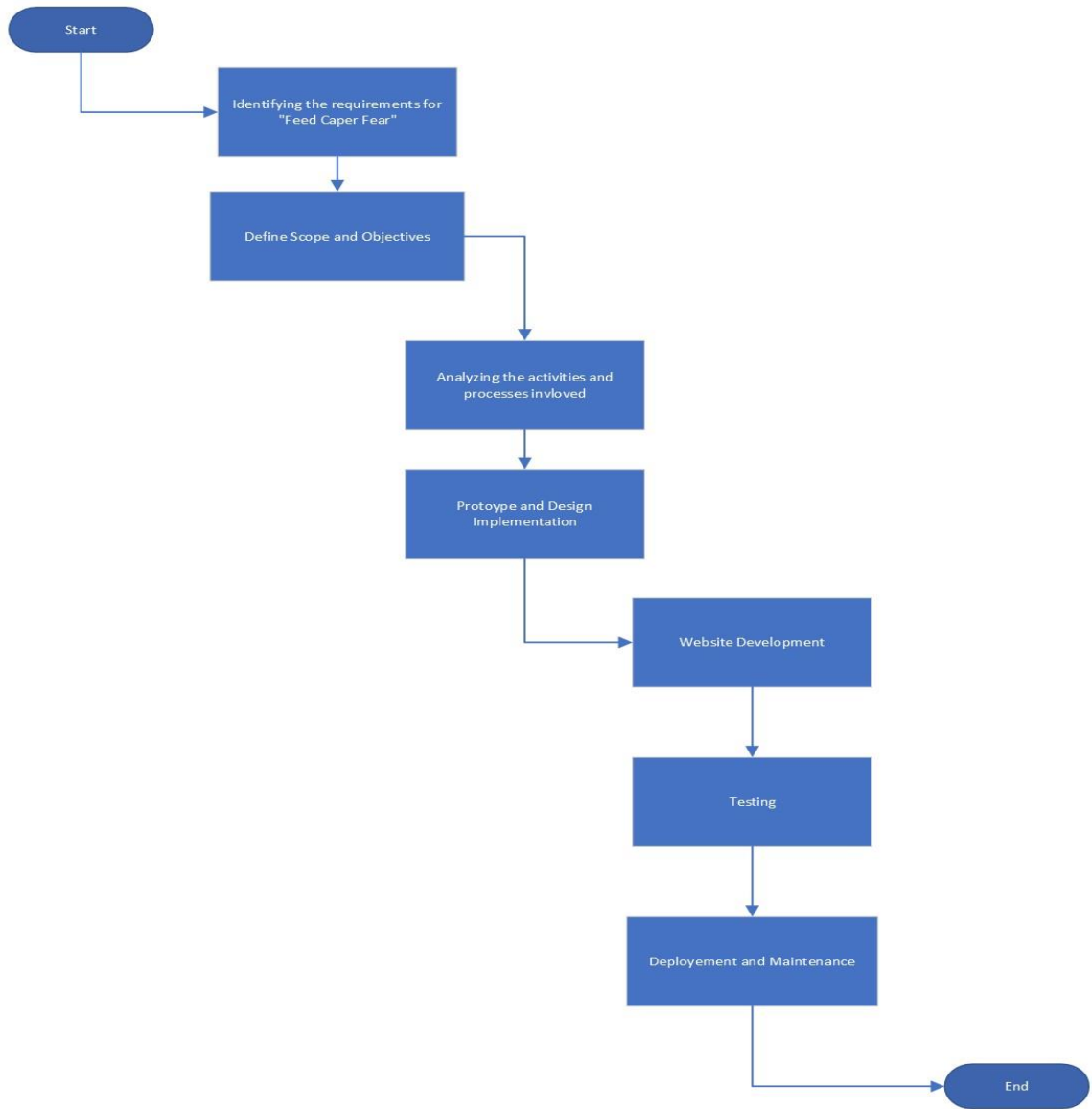


Figure 1: Flowchart for different phases of “Feed Cape Fear”

Development Process	Description
1. Identifying System Requirements	<ul style="list-style-type: none"> • Researching existing solutions. • Conduct user interviews with multiple stakeholders. • Brainstorming techniques and observation to understand user needs and preferences.
2. Defining Scope and Objectives	<ul style="list-style-type: none"> • Including primary objectives, target audience • Developing key project goals and document stakeholder expectations
3. Analyzing activities and processes	<ul style="list-style-type: none"> • Using activity diagrams to depict the flow of activities and processes within the Feed Cape Fear Platform. • Identifying primary actors (users, organizations, administrators) and their interaction with the system. • Develop use case diagrams to illustrate various user scenarios, system functionalities and interactions.
4. Prototype and Design	<ul style="list-style-type: none"> • Develop prototypes or wireframes to visualize the user interface and system functionality. • Gather feedback from stakeholders through prototype testing sessions. • Designing database, data flow of the system.
5. Website Development	<ul style="list-style-type: none"> • Implementing the design by writing the code for the web-based platform. • Using required programming languages, frameworks, and tools that are suitable for the project. • Testing and debugging the code as it is being developed.
6. Testing	<ul style="list-style-type: none"> • Verifying and validating the platform by conducting various types of testing. • Using testing tools, methods, and criteria which are relevant for the project. • Evaluating the quality, functionality, and usability of the web-based platform.
7. Deployment and Maintenance	<ul style="list-style-type: none"> • Launch of platform on a server • Continued support to address any user feedback.

Table 2: Brief explanation about different phases of “Feed Cape Fear”

CHAPTER 2: REVIEW OF LITERATURE

In the active and interconnected community engagement technology-led efforts, as well as in the hunger relief field, "Feed Cape Fear" emerges as a comprehensive solution to the particular challenges faced by the Cape Fear region. In the next three sections of this literature review, each ingredient of the project's foundation is analyzed under different interdisciplinary angles to offer a multi-dimensional view on every aspect in its broad background. In the later sections we analyze the already existing solutions used by many people around the world for the goals which builds "Feed Cape Fear".

2.1. COMMUNITY ENGAGEMENT AND TECHNOLOGY

Community engagement is the collaborative process of addressing issues affecting the well-being of individuals within a community through partnerships and coalitions (McCloskey, 2011). It mobilizes resources, influences systems, and catalyzes changes in policies, programs, and practices to promote environmental and behavioral improvements for the community and its members (CDC, 1997, p.9). This literature on community engagement highlights a need for platforms that move beyond transactional exchanges, creating meaningful relationships between various stakeholders. Often traditional approaches prioritize one-off transactions or exchanges, which lacks engagement and collaboration. Recognizing this critical need for platforms fostering meaningful relationships beyond transactional exchanges, this concept served as one of the main inspirations for "Feed Cape Fear".

The technological innovations in community development as portrayed through platforms like VolunteerMatch, GivePulse, and Nextdoor reveal the potential of digital platforms to function as agents of change. "Feed Cape Fear" acknowledges the

significance of such insights with real-time communication tools, visual engagement for volunteerism and a user-friendly interface that allows smooth usage by users.

On the surface, the combination of community involvement and technology changes how people, groups, firms operate as well as cooperate to advance society. “Feed Cape Fear” uses these insights strategically to design a platform that not only meets the immediate needs of the community but also creates a resilient, inter-connected and involved community in the Cape Fear region.

2.2. FOOD DISTRIBUTION AND LOCALIZED SOLUTIONS

Although hunger relief is the general aim, the focus shifts to the critical aspect of food distribution. Hunger in the Cape Fear region must be solved through localized measures that aim beyond immediate relief. These measures not only eliminate hunger in the short-term but also empower individuals and families to break the cycle of food insecurity, promoting self-sufficiency and resilience within the community (Hope C.O, 2023).

The findings from blog post about MealConnect platform and other Crisis Response Platforms (Crisis Cleanup, n.d) underscore the importance of establishing efficient processes for quickly connecting surplus food with non-profit organizations (Morello, 2023; Greene, L, 2024). Rapid linkage ensures that excess food is effectively redirected to those in need, minimizing food waste and addressing hunger more promptly. By prioritizing efficiency in food redistribution efforts, communities can maximize the impact of food donation and better support individuals facing food insecurity.

“Feed Cape Fear” understands food distribution as a lifeline and centralizes the purpose to guarantee on-time, fair & local provision of necessary nutrition

sources. Through opening exclusive sections for food dispensation and partnerships with local organizations, this project seeks to develop an efficient strategic framework that targets people directly. This local focus on food relief, however, is not just about providing immediate provision but establishing networks that become a sustainable response to the continuing challenge of hunger in the Cape Fear region.

2.3. CORPORATE SOCIAL RESPONSIBILITY INTEGRATION

Corporate Social Responsibility (CSR) refers to the integration of ethical principles and sustainable management practices into the business operations of corporations or firms. CSR involves the implementation of strategies aimed at addressing the interests and concerns of various stakeholders, including employees, consumers, local communities, and the environment. By actively engaging in CSR activities, corporations aim to enhance their reputation as responsible corporate citizens and gain competitive advantage, while also meeting the expectations and demands of stakeholders in an increasingly socially conscious environment (Masum. A, 2020).

The integration of CSR into the “Feed Cape Fear” project is a strategic fit between business and social objectives, which demonstrates the critical importance of corporations in community service. In an article on CSR, Masum (2020) emphasizes the dynamic character of CSR, outgrowing philanthropy to active and meaningful involvement in addressing social issues.

CSR initiatives integrated into the platform enable corporations to demonstrate their social responsibility in an open and transparent manner. This is in line with what has been recommended Aziz (2020); they recommend that companies should ensure better communication of their CSR projects which builds trust and credibility. However, the

most effective CSR initiatives are those that address community needs in a particular context. The inclusion of CSR in “Feed Cape Fear” is an indicator that contributions from corporate bodies have been channeled towards programs aimed at addressing hunger, essential food distribution and community wellbeing.

2.4. EXISTING SOLUTIONS AND PROBLEMS

This part of the review elaborates on probable existing solutions which are already been used by many people around the globe for the goals which shape up Feed Cape Fear. It also contains the problems associated with these existing solutions and in the following section it discusses how “Feed Cape Fear” as a platform having all these features can eliminate these problems.

2.4.1. VolunteerMatch

VolunteerMatch (VolunteerMatch, n.d) is a global online tool that empowers voluntary workers with non-profit entities through its free platform to maximize the results from their operations. It facilitates the search for volunteering opportunities depending on user’s interests and location. Organizations can even post events on the portal, and volunteers can find opportunities and register according to their interests. This platform promotes community involvement and social change by serving as a centralized repository for individuals to meet and partner with organizations who can facilitate efforts in charity, education, environment preservation listed among other projects.

The challenge with this platform is that it has limited localized focus. Since this operates on a global scale this may lead to a lack of specific focus on local community needs. Also, this platform has a lot of generic event listings which makes it challenging

for users to find opportunities tailored to their specific interests or causes.

VolunteerMatch may not have dedicated sections for corporate entities to showcase their CSR activities or engage employees effectively.

2.4.2. GivePulse

GivePulse (GivePulse, n.d) is an online volunteer matchmaking company. It enables users to find and participate in volunteer activities according to the interests they may have as well as their geographical locations. Organizations can create events, volunteer portal management, and community involvement tracking. GivePulse aims at building a comprehensive center of community service where people, non-profit organizations and other organizations can come together to work for the development of society. The site stresses inclusiveness, a place where users can interact, participate in causes they shared in addition to showing how they are making an impact in the community by volunteering and performing some community work. For this project, user interviews were primarily conducted with individuals who were frequent users of GivePulse, providing valuable insights and feedback on their experiences with the platform. This part of the user interaction will be discussed in detail in the later sections.

The challenge with GivePulse is that it lacks a robust community engagement feature such as real-time communication, group activities and interactive forums. It does not have robust system for localized emergency response coordination, making it challenging to disseminate urgent information and organize community efforts during crises.

2.4.3. NextDoor

NextDoor (NextDoor, n.d) is a local community based social networking, which enables communication between residents of neighborhoods. It acts as a network within a community composed of neighbors that can help each other by giving recommendations, tips and even the events that would be held in their area. People can talk about the local problems, shop, and trade with each other on the same platform and interact with the people who are living around them. In its design, NextDoor aims at promoting the ties of neighborhoods by encouraging communication and cooperation among neighbors who can interact virtually but locally.

This platform being a neighborhood based social platform may lack dedicated features for organizing and promoting volunteer opportunities. It also lacks comprehensive tools for tracking individual and community wide volunteer efforts making it challenging for users to quantify their impact. It also does not have special features for promoting volunteering initiatives and broader civic management.

2.4.4. MealConnect

MealConnect (America, F, n.d) is a food rescue platform developed by Feeding America, involving food businesses with extra food to local nonprofit firms that need it. The excess foods by the businesses are easy to donate, and the nonprofits can utilize the MealConnect platform on receiving and distributing these efforts to those suffering from hunger. The case of the platform is that it saves time addressing the issue of food wastage by saving it for those who are in need. The mission of MealConnect is to establish a nation-wide re-distribution system that helps to efficiently redistribute surplus food, further contribute to the battle against hunger and support sustainability.

The issue with MealConnect is that there may be a communication gap between food donors, nonprofit organizations and volunteers which lead to inefficiencies in the coordination and distribution process. This platform does not offer features for engaging and coordinating volunteers specifically for food distribution efforts. This platform lacks features for emergency food distribution during crisis and disasters.

CHAPTER 3: DESCRIPTION OF PROPOSED SOLUTION

3.1. OVERCOMING EXISTING CHALLENGES

“Feed Cape Fear” aims to create a more tailored, engaging, and community focused platform which can effectively meet the unique needs of the Cape Fear Region. On analyzing the challenges in the current available platforms mentioned in the previous section, Feed Cape Fear is developed in a way that it can overcome all the challenges mentioned in the previous section. Below are the methods by which Feed Cape Fear is trying to overcome these challenges.

- **Localized Platform Development:** Feed Cape Fear is specifically designed to address the needs of the Cape Fear Region which ensures a localized focus on issues, events, and other volunteering opportunities. This can help users engage with opportunities and any emergencies which are directly relevant to their own community and give a stronger sense of connection and impact.
- **Comprehensive Dashboard and Community Engagement:** This platform provides a comprehensive dashboard where the emergency and immediate attention posts are classified and pinned at the top of the dashboard. It also integrates real-time communication tools, group activities, and collaborative initiatives to enhance community engagement. The main impact of this is that the users can actively participate, share insights, and coordinate efforts, creating a vibrant and connected community within Feed Cape Fear.
- **Real-Time Communication and Community Engagement:** Feed Cape Fear integrates real-time communication tools, group activities and collaborative initiatives to enhance community engagement. It also includes sharing multimedia while on a chat with any other user of the platform. In this way it is made sure

that users can actively participate in discussions and coordinate efforts which create a vibrant environment connecting the Cape Fear Community.

- **Localized Emergency Response Features:** Feed Cape Fear incorporates the features for emergency response. This feature includes user-generated emergency posts, geolocation tagging, and community coordination during any crisis. This feature will have a larger impact when there is any urgent information to be shared, or to coordinate responses and mobilize any resources during crisis.

Table 3 summarizes the existing solutions, their pros and cons, and how “Feed Cape Fear” overcomes these cons.

Existing Solution	Description	Pros	Cons	How Feed Cape Fear overcomes the cons?
VolunteerMatch	It is a platform to connect volunteers with non-profit organizations globally.	It is as broad platform which allows users to find volunteering opportunities that align with their interests, thereby increasing the engagement within communities.	It has a limited localized focus, leading to difficulties in addressing specific community needs and events.	Its focus is on Cape Fear region, ensuring more relevant platform for community engagement.
GivePulse	This platform connects individuals, organizations, and educational institutions for community engagement	It is very diverse platform where individuals, organizations and communities can come together and make a social impact.	This platform lacks robust community engagement features, which hinders effective communication and collaboration among users and organizations.	This platform integrates the dashboard and communication tools for active participation and collaboration within the community.
MealConnect	It is a program by feeding America, connects surplus food from local businesses with non-profit organizations.	This platform is an efficient food redistribution system by connecting businesses with surplus food to local nonprofits.	It has a potential communication gap between food donors, non-profit organizations, and volunteers leading to inefficiencies in coordinating and food distribution.	It provides real time communication features which ensures efficient distribution of resources for food donation.
NextDoor	It is a neighborhood based social networking platform	It can foster communication and cooperation among neighbors, facilitating the exchange of recommendations and local information.	It has a limited utility during crisis, as it does not offer features for emergency response coordination.	It provides an emergency section where anything related to emergency is pinned to the top of dashboard to get any immediate attention.

Table 3: Pros and Cons of existing solutions

3.2. USER INTERACTION

As a part of research phase for the Feed Cape Fear project, user interactions were conducted with individuals familiar with volunteer engagement platforms, particularly GivePulse users. The aim was to gather insights about how they were familiar with GivePulse, how much they were involved in community engagement and services and how much they were satisfied with the application which they were using for community engagement and other services according to their requirements. The goal was to simultaneously gain more insight into the perceived goals of the project and to get feedback and preferences about the design and development of Feed Cape Fear. A total of four one-to-one interviews were conducted with users representing diverse backgrounds: a non-profit administrator, a director and an assistant director of student community engagement, and a community organizer. Table 4 discusses on the users with whom the interview is conducted and the questions asked to them as a part of interaction which will be used for the development of “Feed Cape Fear Project”.

User	Questions
<p>Director and Assistant Director of student community engagement UNCW</p> <p>Date of Interaction: 12/04/2023</p>	<ul style="list-style-type: none"> • How do you currently participate in community engagement activities within the Cape Fear Region? • What challenges have you faced in connecting with local community initiatives or organizations? • How does your organization currently engage in CSR initiatives? • Which platform do you use for these activities you have mentioned and how often do you use this platform? • Can you share your overall experience with the platform (GivePulse)? What initially drew you to the platform? • Can you identify any features on GivePulse that have had significant impact on your community engagement experience and why are they valuable? • Have you encountered any challenges while using GivePulse? How did you overcome those? • What expectations do you have from a community engagement platform tailored to the Cape Fear Region? Are there any specific features or capabilities you would like to see? • If you had the opportunity to improve the platform (GivePulse) you have been using, what specific changes or additions would you suggest? • Are you using any other community engagement platforms beside GivePulse? How do they compare in terms of meeting your needs and expectations? • If you had a vision of an ideal community engagement platform, what features and functionalities would you want it to include?

User	Questions
<p>Director and administrator of One Christian Network</p> <p>Date of Interaction: 12/18/2023</p>	<ul style="list-style-type: none"> • How do you currently participate in community engagement activities within the Cape Fear Region? • What challenges have you faced in connecting with local community initiatives or organizations? • How do you currently access or contribute to food distribution efforts in the Cape Fear Region? • What support or resources would be helpful for expanding CSR efforts within your organization? • How frequently do you utilize online platforms for community engagement or volunteer work? Which platform do you use more frequently? • What features did you find the most useful in the platform you are using for community engagement? • Have you experienced any difficulties in finding relevant volunteering opportunities or connecting with local organizations through existing platforms? • If you had the opportunity to improve the platform you have been using, what specific changes or additions would you suggest? • In your opinion, what role should technology play in addressing community needs and fostering social impact? • What specific features or functionalities would you like to see in a platform aimed at enhancing community engagement?
<p>Research and Development Consultant and community organizer.</p> <p>Date of Interaction: 01/09/2024</p>	<ul style="list-style-type: none"> • How do you currently participate in community engagement activities within the Cape Fear Region? • What challenges have you faced in connecting with local community initiatives or organizations? • How do you believe technology can better facilitate community engagement and collaboration? • What features did you find the most useful in the platform you are using for community engagement? • Have you encountered any challenges while using any platform? How did you overcome those? • If you had the opportunity to improve the platform you have been using, what specific changes or additions would you suggest?

Table 4: Questions asked during User Interaction

During user interactions with individuals familiar with GivePulse, participants expressed appreciation for several key features that enhanced their volunteer engagement experience. They appreciated the platform's clean design, clear layout, and

straightforward registration process, which minimized barriers to entry and enhanced the overall user experience. They emphasized the platform's event scheduling, RSVP tracking, and attendance management features, which streamlined the planning and execution of volunteer activities. They also mentioned how it underscores its value as a platform for facilitating volunteerism and social impact. They also had a few issues with the platform where the new users initially found it difficult to navigate through the tool. Also, it had many features which made it a bit difficult to navigate through various features offered by the platform.

During the interaction, two of the users had some ideas about other platforms like VolunteerMatch and NextDoor but they could not remember much about their experience with these platforms. They did mention that VolunteerMatch was the first choice for their organizations in the past if they were in search of volunteers or any event. In the later part of user interaction, the idea of Feed Cape Fear was explained to them, and the valuable insights gained from the conversation regarding the requirements of Feed Cape Fear are summarized below:

- **User-Friendly Platform:** participants expressed a strong preference for a user - friendly platform that simplifies the process of discovering and participating in volunteering opportunities. They emphasized clear navigation and simple design.
- **Interactive Features:** Participants highlighted the need for real-time communication as one of the significant features for any community engagement platform. They also emphasized how fast and efficient communication makes it easy to increase involvement of users.
- **Priority for Volunteering:** Participants expressed their need on a focused volunteering page in the platform, to which the users can easily navigate to and

search for volunteering opportunities. They mentioned that it can foster community engagement and facilitate meaningful connections.

- **Community Engagement Features:** Participants appreciated Givepulse's emphasis on community engagement and collaboration. They mentioned having community forums and group facilities on the platform. They did highlight this would foster a sense of belonging, facilitating knowledge sharing and mobilizing collective action within their organization or group.

In the next part of user interaction, it was purely on gaining some feedback on the Feed Cape Fear Platform which will be later discussed in the upcoming sections of the document. Using the information gathered after doing user interaction, the system requirements can be summarized as follows:

- **User Registration and Authentication:** Users should be able to register securely with their personal information. The system should authenticate users during login to ensure secure access.
- **Activity Feeds and Posts:** The system should support activity feeds where users can view and interact with posts from other users, organizations, and groups. Users should be able to create, edit, and delete posts, including text, images, and videos.
- **Groups and Organizations:** Group and organization pages should provide features for posting announcements, events, and volunteer opportunities.
- **Volunteer Opportunities:** The system should facilitate posting and discovering volunteer opportunities by organizations and individuals. Users should be able to

register volunteer opportunities based on criteria such as location, date, and type of service.

- **Real-Time Communication:** The platform should include personal and group chat options for users to communicate in real-time. Users should be able to share text, images, and videos within chat conversations.
- **Geolocation and Tagging:** The system should support geolocation tagging for posts, events, and volunteer opportunities to indicate their location.
- **Emergency Response Features:** The platform should include features for users to post and share emergency alerts and requests for assistance.
- **Volunteer Interest Tracking:** The system should track users' interest in volunteer opportunities and provide options for users to express interest or RSVP for events.
- **Scalability and Performance:** The system should be scalable to accommodate a growing user base and increasing data volume. It should maintain high performance and responsiveness even during peak usage periods.

3.3. GOALS FOR FEED CAPE FEAR

"Feed Cape Fear" aims to combine elements mentioned in the previous sections into a single, comprehensive platform tailored for the unique needs of the smaller region. Its potential lies in creating a centralized hub that not only connects volunteers with opportunities but also integrates localized food distribution, engages corporate entities in CSR initiatives, and provides a dynamic space for multi-stakeholder collaboration, ultimately fostering a stronger and more resilient community. It aims to represent an

innovative approach that combines essential features from various platforms, catering specifically to the challenges and opportunities within the Cape Fear region.

3.3.1. Localized Platform Development

The localized development of the platform represents one of the strategic approaches to focusing on developing and aligning Feed Cape Fear with unique needs, advantages, features contexts as well for that region. To achieve this, some of the characteristics that must be considered include socioeconomics, culture, and geographical aspects to ensure that it becomes an actual useful tool for people in community, non-profit organizations, and corporations as well as volunteers. Implementing the geolocation features can provide localized content and services specific to the Cape Fear region.

3.3.2. Comprehensive Dashboard

The dashboard is also supposed to focus on the users, where it will prompt active participation and allow easy navigation through all the functionalities offered by Feed Cape Fear. It functions as a hub through which the user will search relevant information, reach out to connections, and update themselves with community happenings.

The main interface of Feed Cape Fear is the comprehensive dashboard, which serves as a one-stop shop for all necessary details concerning activities and opportunities presented through it. The dashboard is meant to improve the user experience and engagement, hence having a critical role in keeping community members informed and linked.

3.3.3. Real-Time Communication Tools

The tools for real-time communication are indeed highly important in the development of a vivid and interactive community within Feed Cape Fear enabling them to collaborate, coordinate, and even support each other right on time (What is Community Engagement in a digital world? n.d). These elements add to the success of this platform in promoting community participation and social impact. Feed Cape Fear platform includes personal and group chat options, multimedia sharing, and accessibility across devices to facilitate seamless communication between users of the platform and take actions at the right time if there are any emergency situations in the Cape Fear region.

3.3.4. Volunteer Interest Tracking

Volunteering with Feed Cape Fear is not merely an activity, but a force for good. The emphasis on inclusivity, community-driven initiatives, and easy coordination of efforts makes this platform a powerful instrument for individuals, organizations, and businesses that want to change the living conditions in the Cape Fear region. The goal is to develop a community-based platform that allows users to actively participate in the region's growth and welfare. The platform aims to provide a convenient and easy-to-use volunteering structure that motivates people to join various initiatives, events, and community service activities. Volunteering opportunities posting, User interest tracking and Feedback and Reviews are some features aimed to be developed in the platform.

3.3.5. Localized Emergency Response

The goal of localized emergency response in Feed Cape Fear is to promote a proactive and community-oriented crisis management system allowing users to react quickly and effectively on emergencies while fostering collaboration, information sharing, and community resilience. This functionality aims to leverage technology to enhance community resilience and improve coordination during the time of emergency situations. The functionalities aimed to be included in the platform are immediate posting of emergency posts which gives top priority for display in the user dashboard. It also includes geolocation tagging which helps in providing accurate information about the location of the crisis or urgent situation.

3.3.6. Multi-Stakeholder Collaboration

Feed Cape Fear is based on multistakeholder alignment, and it needs collaborative effort of different stakeholders. Strategic multistakeholder collaboration is aimed within the Feed Cape Fear platform where non-profit organizations, corporate companies, community members as well as local authorities come together. This partnership model is aimed at multiplying the effect of community participation initiatives, developing cooperative problem solving and creating synergistic environment where different needs are addressed.

3.4. TARGET AUDIENCES

The main target audiences for “Feed Cape Fear” are:

- Non-profit organizations: These are the organizations which are involved in community service, humanitarian aid, and social welfare programs within the Cape Fear region.
- Corporate Entities: These are the businesses that are interested in corporate social responsibility initiatives, volunteer engagement programs, and community outreach efforts in Cape Fear region.
- Volunteers: These are the individuals seeking opportunities to contribute to community service, participate in volunteering activities, and make a positive impact within the Cape Fear region.
- Community members: Residents of the Cape Fear region who are a part of any organization in Cape Fear region and interested in accessing resources, participating in local events, and staying informed about community initiatives and emergency efforts.
- Local authorities and emergency services: These are the agencies which are responsible for public safety, emergency management, and disaster response in the Cape Fear region, who may utilize the platform for communication and coordination during crisis.

3.5. TECHNOLOGY STACK

The development of Feed Cape Fear will ensure scalability, flexibility, and responsiveness. In the below subsections various alternatives for technologies are discussed and one technology is chosen based on the requirements a technology is chosen for the development of this project.

3.5.1. Frontend Development

- **Bootstrap:** Bootstrap is a CSS framework that provides pre-designed templates and components for building responsive and mobile-first websites. It is widely used for its use and extensive documentation. Bootstrap enables quick development of responsive websites and ensures consistency across browsers and devices, simplifying the development process and ensuring a uniform user experience. On the other hand, websites developed using bootstrap may appear generic due to reliance on default styles and components. Also including the entire bootstrap framework can result in larger file sizes and increased load times.
- **Vue.js:** It is a progressive JavaScript framework that is known for its simplicity and flexibility. Developers can incrementally adapt to its features and integrate smoothly with existing projects. It offers reactive data binding and a component-based architecture like react (Karczewski, 2022). Vue.js has a limited ecosystem and community support compared to Angular and React. It also has less mature tooling and documentation.
- **React:** React is a free and open-source front-end JavaScript library for building fast and interactive user interfaces for web or Mobile Applications. Any user can access the source code, modify it, and can use it. In Model, View, Controller architecture, react is the View which is responsible for how the app looks like and feels. It follows a component-based architecture which makes it easier for building user interfaces. Some disadvantages of using React are they require additional libraries for routing which increases the complexity.

Based on the project's requirements and considerations, the chosen technology for Front-end Development of "Feed Cape Fear" platform are React and Angular. Because of its component-based architecture and reusability React has been chosen for this project. Angular is also considered because of its two-way binding which makes it suitable for building complex user interfaces.

3.5.2. Backend Development

- Django: Django is a Python web framework which is famous for its scalability, security, and the ability to develop fast (Klien, M, 2021). The framework follows the Model-View-Controller (MVC) architectural pattern, and it includes some built-in features such as an Object-Relational Mapping (ORM) system, an admin interface, and URL routing. The merits of Django are its great documentation and the vast ecosystem of reusable components, plugins, and third-party libraries. Nevertheless, Django's learning curve is rather steep for the beginners who encounter its large feature set, and its "batteries-included" philosophy can result in overhead in the cases when the lightweight frameworks are needed.
- Flask: Flask is a Python web framework appreciated for its simplicity and ease of use, being lightweight and flexible. It gives the developers a set of tools that are required for the development of web applications and APIs fast and easy (Klien, M, 2021). The advantages of Flask are that it follows a minimalistic approach which means that the developer gets more control over the system, and it also has many extensions that can be used to add extra functionality. While Flask's minimalist approach might be a drawback for larger projects as it means developers would have to handle additional configuration and integration tasks

manually. Furthermore, Flask may not have as many prebuilt features and features that are easier to use as the bigger frameworks Django.

- **Express:** Express.js is a lightweight, flexible Node.js web application framework that streamlines backend development. It provides a rich set of features for construction of web servers and APIs such as middleware support, routing, and template engines. A strong point of Express.js is the fact that it is simple, making it easy for developers to set up and configure web applications. Additionally, it has an expansive ecosystem and a huge user base, which is a great benefit. Express.js may not come with some of the built-in features that these more opinionated frameworks have, so developers may have to use third-party middleware or create their own solutions instead.

Express.js will be employed as the backend framework for building RESTful APIs and handling server-side logic. Its lightweight and minimalist nature make it ideal for creating scalable backend services. Further, Express.js integration with Node.js ecosystem and rich middleware support cater well to the project's requirements of real-time communication, geolocation tagging, and community engagement features.

3.5.3. Database Management

- **PostgreSQL:** PostgreSQL is a highly popular free and open-source relational database management system (RDBMS) capable of providing a strong level of data integrity, flexibility in data modeling, and compliance with SQL standards. It provides advanced functionality including support for complicated queries, JSON data types, and full-text search and so on, which is suitable for a great variety of

applications (Kirpichny, A, 2024). PostgreSQL offers high data integrity, concurrency control, and transactional capabilities as well, resulting in reliability and data consistency. The initial setup and configuration of this database may be complicated as compared to other databases, and its performance may degrade when handling write-intensive workloads.

- **MongoDB:** MongoDB is a popular NoSQL database management system, notable for its flexibility and scalability. It can store data in flexible, JSON-like documents and thus is suitable for working with unstructured or semi-structured data. MongoDB's document-oriented model is schema-less, ensuring easy schema evolution and horizontal scaling, making applications adaptable to changing requirements and able to handle growing datasets (Kirpichny,A, 2024). One benefit of MongoDB is its built-in support for sharding and replication, which makes it easier to implement high availability and fault tolerance. Also, the powerful query language and indexing in MongoDB provide for quick data retrieval and analysis. However, in terms of storage space, MongoDB may consume more space because of denormalization, and its eventual consistency model could bring data consistency issues in the distributed environment.
- **MYSQL:** MySQL is one of the most popular open-source relational database management systems (RDBMS), which is appreciated for its performance, reliability, and ease of use. It provides the complete set of features for operating structured data like transactions, indexes, and stored procedures. A benefit of MySQL is its scalability which makes it convenient for both small-scale applications and large-scale enterprise systems. Moreover, MySQL has a vast and lively community, giving abundant documentation, support, and third-party tools.

MySQL might be slower when working with huge sizes of data or complex queries than some other RDBMS.

MYSQL will be the chosen relational database management system (RDBMS) for storing and managing data related to users, organizations, events, and volunteer opportunities. It is easy to integrate with other technologies which are commonly used in web development. It offers a mature ecosystem with extensive documentation, community support, and a wide range of tools and frameworks making it a reliable choice for this project.

3.5.4. Package Management

- **Yarn:** Yarn is a JavaScript package manager tool which is faster and reliable when it comes to dependency management as compared to NPM. It was created by Facebook together with other tech giants to solve some problems of NPM. The offline package installation, deterministic dependency resolution, and parallel downloads provided by Yarn can greatly accelerate the development process. Yarn provides a lock file (yarn.lock) that guarantees identical installations in different systems. Nevertheless, Yarn is relatively larger in terms of file size than NPM and could have compatibility issues with some NPM packages.
- **NPM (Node Package management):** NPM is a popular package manager for JavaScript, and it is used primarily in the Node.js environment. It lets developers install, publish, and maintain packages and their dependencies for the purposes of

web app development. The huge number of packages in NPM is the major benefit of it, which amounts to over a million packages with diverse functionalities.

NPM has a clean CLI for package installation and version management, which eases the development process. The dependency tree of NPM may be too complex and vulnerable to version conflicts that may cause issues with package compatibility and resolution of dependencies.

NPM will be used for package management, dependency resolution, and script execution, streamlining the development workflow. NPM possesses an enormous package repository, coupled with its advanced dependency management features, which enables developers to have access to a wide variety of JavaScript packages, thus making project development and deployment processes smooth.

3.6. ANALYZING THE ACTIVITIES AND PROCESSES

The main goal of this section is to gain a comprehensive understanding of the activities and processes involved in this project. Figure 2 below describes the journey of a user while exploring the activities in the “Feed Cape Fear” platform.

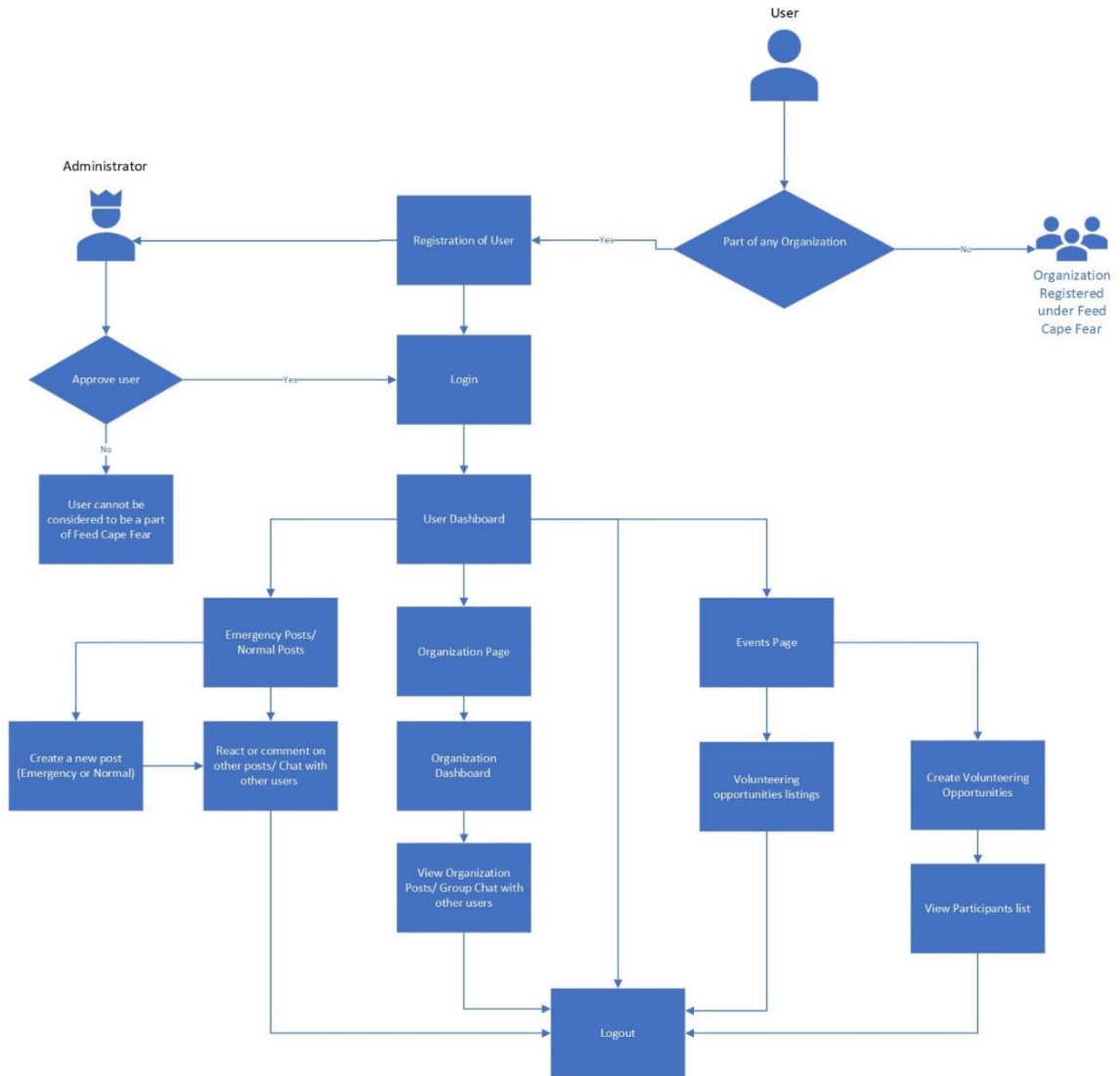


Figure 2: User journey diagram

All the rectangle boxes in Figure 2 depict the activities which the user can perform on the “Feed Cape Fear” platform. Table 5 below mentions the details of important activities and what the user can do at that block of the user journey diagram.

Activity	Description
Registration of user or User Registration	Allows users to create accounts and register on platform
Login or User login	Once registration is approved, user can login through ID provided automatically by platform and user created password.
Create Emergency Post	Enable users to post and view emergency alerts or urgent community needs. (these posts are pinned to the top of dashboard for immediate attention)
Create New Post	Enables users to post any community need or share any achievement of individual and let other users on the platform know about it. (this can be achieved through multimedia posts as well)
Organization Dashboard	Separate page for each organization, like a group page in any social media application. Users can share any information specific to individual organization.
React or comment on any post (user/organization dashboard)	Enables user to react to any post by other users in the platform and can comment on any post in the user dashboard.
Create Event listing or Volunteering opportunities	Enables user to create a volunteering opportunity or event by posting it in the events page.
View participants list	Users who created the opportunity or event can view the list of participants interested in the event or volunteering opportunity.

Table 5: List of activities and their descriptions

CHAPTER 4: METHODOLOGY

This chapter dives into the structured approach adopted in developing Feed Cape Fear, encompassing key stages like the Software Development life Cycle application, comprehensive analysis of user journeys and use cases, prototyping and design processes, and the design of the database architecture. This chapter is a route map, which displays the methodical process of figuring out the support system, designing the features as a whole and implementation of the platform's functionality. It presents the elements involved in strategic decision-making, collaboration, and the continuous fine-tuning done to ensure the successful development and deployment of Feed Cape Fear in line with project objectives and stakeholder requirements.

Software Development Life Cycle process was always an integral part at each stage of Feed Cape Fear project. With the help of SDLC clear requirements and objectives for the platform were defined and made sure the project goals were aligned with the stakeholders goal. Detailed analysis and planning were achievable through facilitating SDLC, which includes studying feasibility and resource allocation. Also made sure the project's scope, timeline and resources were well defined and achievable. Moving onto the design phase here, the focus is to design the system architecture, database, and user interface layouts. The below subsections explain the analysis made on designing the user experience, user interface and database.

4.1. ANALYSIS OF USE CASES

We use a use case diagram to analyze the activities of “Feed Cape Fear” by providing visual representation of the interactions between users and the system. Firstly, we identify the actors or users involved in the system. Later we understand the use case representing a specific user goal or activity within the system. Using these use cases, we can get a high-level overview of the systems functionality from a user’s perspective.

Figure 3 shows the use case diagram used to guide this project while Tables 6 and 7 give the use case descriptions for the User and Organization/Admin actors, respectively.



Figure 3: Use Case Diagram for “Feed Cape Fear”

Use Case	Description
User Registration	This use case allows the user to register to the platform by giving the required credentials.
User Login	This use case allows the user to access the system by entering their userID and password.
Emergency Post	This use case allows the user to view or create an emergency post which is relevant to any of the current problems in the locality.
Volunteering Opportunities	This use case allows the user to create any volunteering opportunities or show interest towards any other opportunity in the volunteering page.
View participation List	This use case allows the user to view the list of users who are interested in the event/ opportunity posted on the platform.
Group/Organization Post	This use case allows the user to create, view and interact with posts from groups or organizations that they follow or join on the system.
Real-Time Communication	This use case allows the user to communicate with other users or organizations through private or group messages either through plain text, image, or video.

Table 6: Use cases description for the User actor.

Use Case	Description
User Registration	This use case allows to approve/decline the registration of user after doing a background check by the administrator.
User Login	This use case allows the admin to login to the platform by entering their unique ID and password.
Approve User	This use case allows the admin to approve the registration request by user after a manual background check.
Decline User	This use case allows the admin to decline the registration request by user after a manual background check.

Table 7: Use Cases description for the Organization/Admin actor.

4.2. PROTOTYPE AND DESIGN

User Dashboard View:

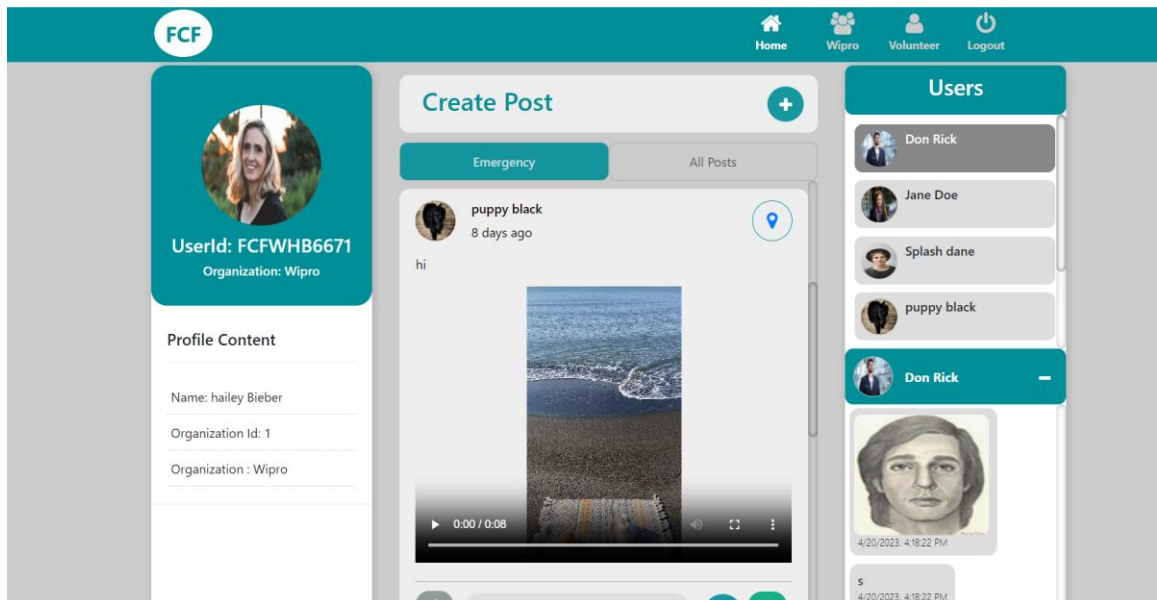


Figure 4: User Dashboard prototype

This is a user interface of the platform “Feed Cape Fear” which links non-profit organizations, corporate bodies, and individuals who have a common goal of no member of the Cape Fear community experiencing hunger. The interface enables users to create posts, view posts categorized under "Emergency" and "All Posts" and see a list of users. Users can also chat with each other. The interface is light teal green background with whitish panels for content. The user’s profile section displays their profile photo, User ID, and company. The Post section contains an example of a post with a picture and text. The user list part displays names and avatars of other users. The prototype is developed with the intention of improving user experience and volunteerism by enabling users to post and view volunteering opportunities using pictures or videos.

Admin Dashboard View:

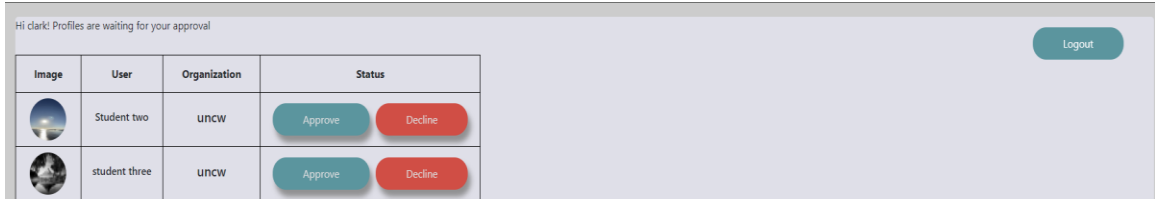


Figure 5: Group/Organization Prototype

This is the user interface on the admin side where the admin can only view the list of users who finished registering to the platform and are waiting for approval from the admin. This Interface is designed to streamline the process of managing member access within an organization. This page includes a table containing the information of registered users. Each user profile includes a photo, name, and organization they belong to, which allows for quick visual identification. Then comes the “Approve” or “Decline” options associated with each profile, where the admin can click approve so that the user can login or else decline to reject the user registration and decline their request. Lastly the “Logout” button will let the admin log out of the platform and redirect to login screen.

4.3. DATABASE DESIGN

The database structure is represented using the Entity Relationship Diagram. It depicts the entities, their attributes, and the relationship between them. In the context of Feed Cape Fear, the entities are the users, organizations, events or volunteers, and their attributes such as names, IDs, quantities, and locations. Relationships between these entities, like users belonging to organizations, organizations distributing food items, and users interacting with food items, would be clearly defined. The ER diagram aids in understanding the structure of the database by visually organizing entities and their

relationships. It helps in identifying the key components of the system, defining data attributes accurately, and establishing the rules governing data interaction. This clarity ensures data integrity, reduces redundancy, and facilitates efficient database management. The ER diagram serves as a blueprint for designing a well-structured and functional database system, aligning data architecture with the requirements and goals of Feed Cape Fear.

The Entities are:

- Users
- User_details
- Chat
- Post
- Comments
- Group_chat
- Organization_lookup
- Events
- Interested_event

Figure 6 shows the database design for Feed Cape Fear which includes all the mentioned entities and required attributes.

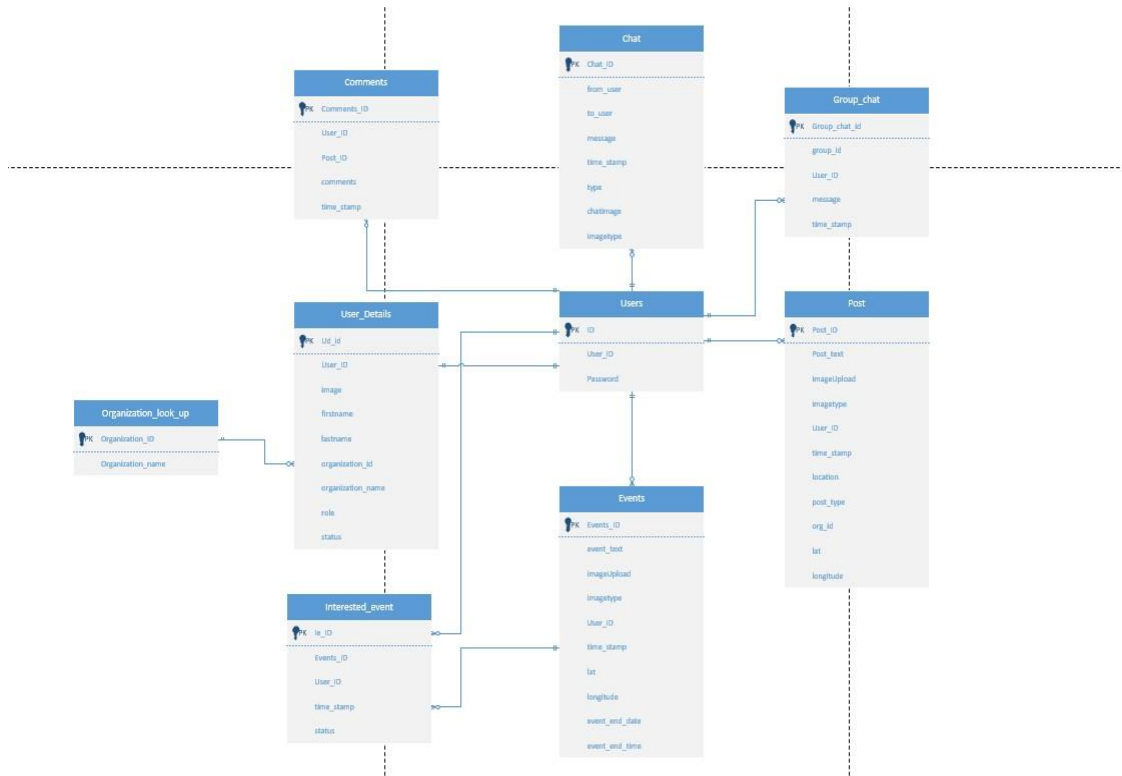


Figure 6: Entity Relationship Diagram for Feed Cape Fear

4.4. SECURITY

The registration and login process in the Feed Cape Fear application is designed with a strong emphasis on security to protect user data and ensure a safe user experience. This section outlines the security measures and procedures involved in user registration, admin approval, and user login.

The pre-requisites needed for any user to be on Feed Cape Fear Platform are:

- Users should be a part of any organization or corporate entity or emergency services in the Cape Fear region.
- Users should have a decent history during their tenure with the organization.

Users with any record of negative works or anything which is against the good

intentions of Feed Cape Fear during their tenure in the organization, then the admin must decline their registration request.

- The system is designed to have a single admin, controlling the entry of users into the system based on their background check which is done manually by the admin.

Secure Registration Process:

- **Data collection:** During registration users are required to provide essential information like mentioned in previous chapter i.e., First Name, Last Name, organization they are from, their profile photo and finally the password they want to set for their account. The input is validated to prevent incorrect or malicious data entry.
- **Password Security:** User passwords are hashed using an encryption algorithm before they are stored in the database. This ensures that they are protected from unauthorized access.

Admin approval process:

- **Pending Status:** Once the user provides all the details required during the registration process, a unique ID is generated according to the user details provided during their registration. Later, the admin page, a new user approval request pops up. In the background the admin must do the background check of the user and upon meeting the above-mentioned pre-requisites, admin hits the approve button and later the user can login to the system. If the user does not meet the required pre-requisites, then the admin prevents the user from getting into the system.

- Admin screen: The admin screen is very much different than that of what the user watches when they are approved to login to the system. The admin screen only consists of the list of users which are to be approved or declined.

User Login Process:

- Upon successful completion of registration process, the user has to login to the system using the unique ID generated along with the password which the user has provided during the registration process.
- The login process involves session management to validate user credentials and grant access only to the authorized users.
- User access and permissions are managed through a role-based access control approach, ensuring that users have access only to the resources and functionalities based on their assigned roles.

These security measures which are implemented in the registration and login processes of Feed Cape Fear application aim to safeguard user data, prevent unauthorized access, and ensure a secure environment for user interactions.

CHAPTER 5: DEVELOPMENT

5.1. INITIAL SETUP FOR THE SYSTEM

The initial setup for “Feed Cape Fear” involves configuring the development environment and setting up the necessary tools and frameworks to start building the platform. This includes installing the Visual Studio Code as the primary Integrated Development Environment (IDE) for the editing and management of the codes, for instance. JavaScript is based on React framework for presentation layer development, while Node.js is used for the internal logic and back-end part of the application. One of the great features of NPM is that it allows for the easy installation of node.js packages and their dependencies. Also, MYSQL is the Relational Database Management Systems (RDBMS) which stores the data in a relational manner. Establishing this structure will serve as a basis for constructing a system, that will be able to do the work required to help the population of Cape Fear Region that is experiencing hunger and provide a venue for community engagement.

5.1.1. Installing Visual Studio Code

Visual Studio Code is used as code editor for Feed Cape Fear platform. Visual studio code is a lightweight code editor with all the capabilities of a full development stack which includes debugging, task running, and version control. Its main goal is to provide a tool according to developers needs for a rapid code-build-debug loop. In this project for the initial setup having visual studio code is a huge advantage which includes all necessary tools for smooth development of the

platform. We can add a vast library of extensions which can be installed to enhance the functionality. We can use it for specific frameworks, languages, or tools. It has an integrated terminal which enables to execute, run scripts, and manage the dependencies without leaving the editor. Extensions like live server can be used to launch a local development server directly from VS code, providing real-time updates and previews of changes made to Feed Cape Fear.

5.1.2. Setting up React for Front-End Development

The user-friendly interface of visual studio code, the extensive plugin ecosystem, and built-in features such as IntelliSense, debugging tools, and version control integration make it an ideal choice for front-end development with React. Before setting up React on visual studio code we need to make sure we have Node.js installed on our system. Then to set up React in VS Code, we use the “create-react-app” command to generate a new React project, navigating to the project directory, then we start a development server and begin coding. Working with VS Code’s intuitive interface and powerful tools, working on React projects becomes streamlined and efficient, enhancing productivity and the code quality.

5.1.3. Setting up Express for Backend Development

Express.js is a Node.js framework. So had to be sure about the Node.js previously installed. Creating a new project folder and initializing a Node.js Project using the command “npm init -y”. We then use the command “npm install express” to install Express.js and then we create a file for our application. This process

enables efficient backend development using express.js in VS Code, allowing to create a robust server-side application with ease.

5.1.4. Setting up MYSQL for database

For the database we had to install MYSQL server and client on my machine.

Later had to start the MYSQL server and create a new server for Feed Cape Fear.

Later had to connect the Backend development with MYSQL so a separate “.js” file was created to handle the MYSQL connection.

5.2. USER REGISTRATION AND AUTHENTICATION

According to the user requirements, the main goal is to allow users to register on the platform, providing essential information and generating a unique ID for Login. Fields are included for basic information and account creation. The information taken for user registration are:

- Profile photo of the user (recommended to be professional)
- First name of user
- Last name of user
- Which organization they are belonging to
- Password which they want to set for their account.

After registration of user, user is assigned a unique key to log into the system in case if the user has been approved to get into the platform by the admin. For this platform there will be a single admin who has the power to let any user into the application after

manually doing the background check for the user according to their security norms.

Figure 7 shows the screen for the user if they want to register into the platform.

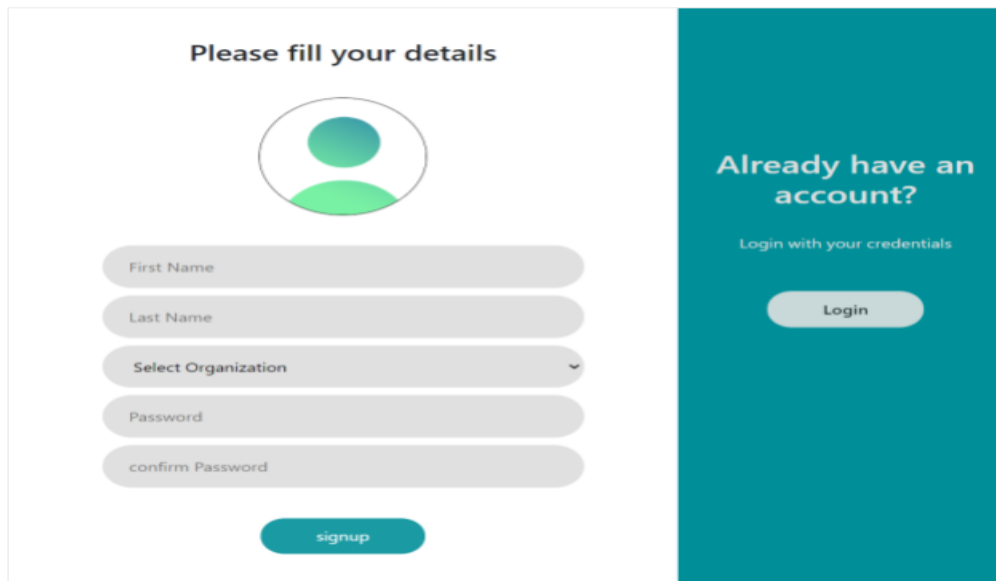
The image shows a user registration page. On the left, there is a white form titled "Please fill your details". At the top of the form is a circular profile picture placeholder with a teal and light green gradient. Below it are five input fields: "First Name", "Last Name", "Select Organization" (a dropdown menu), "Password", and "confirm Password". A teal "signup" button is at the bottom of the form. On the right, there is a teal sidebar with the text "Already have an account?" and "Login with your credentials" above a white "Login" button.

Figure 7: User Registration Page

All the details entered by the user in the form are sent to the backend using axios as form data and unique user ID is created to the user, the role is set as “USER” and status as 0 for all new users who registered into the platform. Then the user details are inserted into the “user_details” and “users” table using database connection. The response is that the user object is stored as JSON string with the following parameters used:

```
user = {  
  userid:",  
  firstname:" " ,  
  lastname: "“" ,  
  organizationid:" " ,  
  organizationname:" " ,  
  password:" " ,
```

```
role:",  
status:" "  
};
```

User Login includes a screen where the user must provide details like the ID generated once the user has been registered and the provided password during the registration phase of the user. It is important that the user must be approved by the admin after manually doing a thorough background check and only then can the user log in to the platform.

Figure 8 shows the screen which the user uses to log in to the platform.

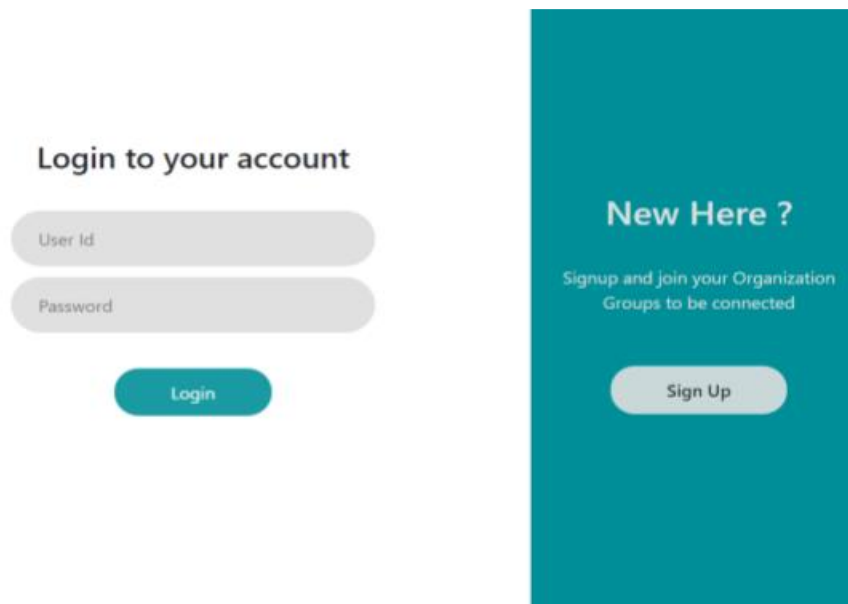


Figure 8: User Login Page

Once the user has been registered it is the admin turn to either approve or decline the user from the platform. The admin for this platform has a unique ID and password to log into the platform where the admin is taken to a separate dashboard which is different from the regular users of the platform. Figure 9 shows the screen for admin to either approve or decline the users.



Figure 9: Admin Dashboard Page

Once the admin approves the user, the status of the user is updated to 1 using “UPDATE” command on the “user_details” table. If the admin rejects the user, then the “user_id” is deleted using “DELETE” command on the “users” and “user_details” table.

5.3. USER DASHBOARD AND EMERGENCY/NORMAL POSTS

Upon successful completion of registration process and login, the user is redirected to the user dashboard page and their details are stored in browsers local cache. The user dashboard includes a navigation bar on the top of it which contains a logo of the application, link to groups, a home button, and a logout button, clicking on the logout button allows the user to log out of the application which will also clear the data in the browsers local cache.

The user dashboard was designed using React components to create a dynamic and responsive user interface (UI). Routing was done between different dashboard sections such as user dashboard, groups/organization page and the events page. Express.js was employed to develop RESTful APIs to fetch user-specific data like profile information like the user details, the data on the groups page which they belong to respectively, the

messages, the dashboard for that user and the events page for that specific user. On the database side queries were optimized for quick retrieval of user-related information.

The dashboard panel has three main panels, the left panel lists the user details such as User Image, user-id, organization, and user details provided during sign-up. The central panel contains an accordion model form to create posts with message, post-type, image in jpg/jpeg/png format, location fields followed by list of posts each containing image of the author of the post(user), First name and Lastname and time lapsed from post creation along with link to the location tagged in the post and image. The interface for posts(emergency/normal) is created using react components for input forms and buttons. Each post allows logged in users to like/unlike the post, view comments for the post, view comments for the post and comment on the post. Posts were stored in database with related attributes, indexing was used for retrieving efficiently based on type and timestamp.

The right panel consists of a list of users in the top section followed by a chat box in the bottom section. Upon clicking on user details in the user list, the chat box is populated with the appropriate chat data with message and time lapsed from message creation. There is a provision for messaging the user at the bottom of the chat box. Figure 10 shows the dashboard view for the user registered into the platform.

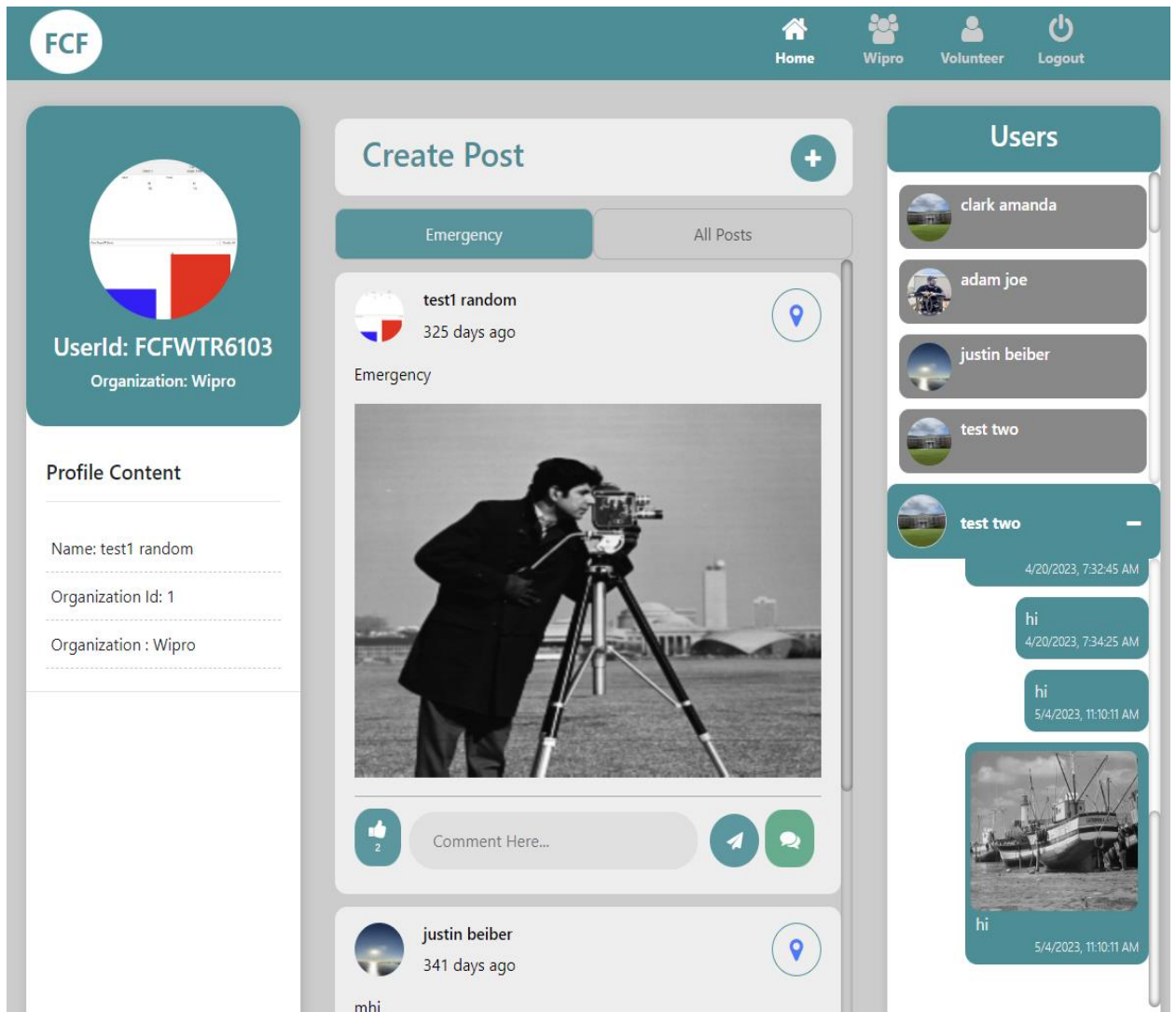


Figure 10: User dashboard Home Page

In the central panel of the dashboard as mentioned previously, there is an accordion model for creating posts. If the user wants to have immediate attention to the post, then they can change the post type to emergency and Post. So, the post will be displayed in the emergency panel as the latest post. The users can even do normal posts. After posting there will be a list of posts from different users according to time, and the latest emergency post will be displayed first. And to see all the posts on the panel, click on all posts tab, user can see list of all the posts including emergency posts.

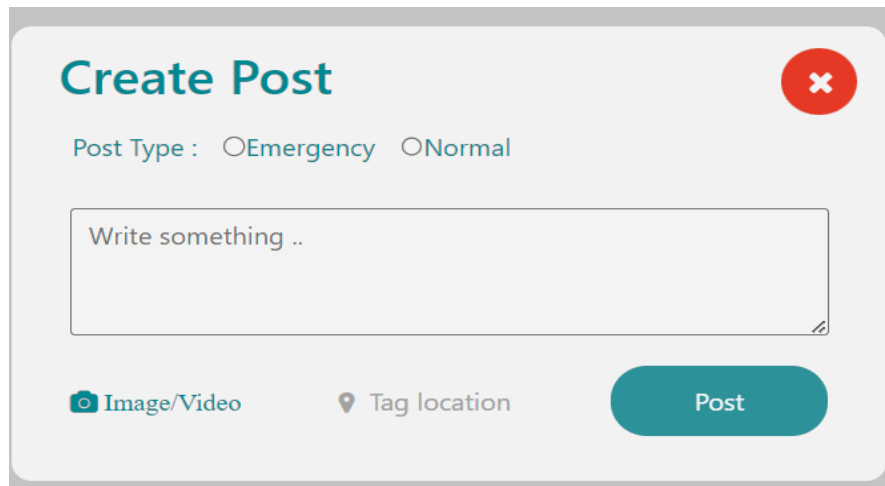


Figure 11: Post Creation window

To create a post the user must give the following input parameters. All the details entered by the user in the form are sent to the backend using axios as formdata. Using the “INSERT” command all the details are stored in the “post” table. The response is the user object as JSON string with the following parameters:

```
user = {  
  posttext : "",  
  posttype : "",  
  lat: "",  
  long: "",  
  userid: "",  
};
```

Comments for a post can also be created by the user by clicking the message icon under the post. The user must just include the comment they want to give towards the post and then post the comment. Once the comment is sent the details of the comment are stored in the “comments” table. The response is stored as a JSON string with the following parameters:

```
user = {  
    comment:""  
    userid:""  
    postid:""  
    timestamp:""  
};
```

5.4. GROUPS/ORGANIZATION PAGE

In the navigation bar of the dashboard, there is a link to groups. Clicking on the link redirects users to the groups panel. The group/organizations panel is the same as the dashboard panel as mentioned previously. In the groups panel, only the users who are in the same organization can see the data posted by other users. The left panel lists the user-details such as User-image, user-id, organization.

The central panel contains an accordion model form to create group-posts with message, post type, image in jpg/jpeg/png format, location fields followed by list of group-posts each containing image of the sender of the post(user), Firstname, Lastname and time lapsed from post creation along with the link to the location tagged in the post and image. Each post allows logged in users to like/unlike the post, view comments for the post and comment on the post.

Using the “INSERT” command all the group post details are stored in the post table using organization id. The response is the JSON string with the following parameters:

```
user = {  
    posttext : "",  
    posttype : "",
```

```
lat: """,
long: "",
userid: "",
orgid:""
};
```

The right panel consists of a list of users in the top section followed by a group chat box in the bottom section. The group chat box is populated with the appropriate chat data with name of the user, message and time lapsed from message creation. All the members in the group can chat in this Group chat box. Figure 10 shows the group page of the platform.

All the details entered by the user are sent to the backend using axios as formdata. Using the “INSERT” command all the chat details are stored in the “group_chat” table. The response is JSON string with the following parameters:

```
user = {
    group_id:"",
    userid:"",
    message:"",
    timestamp:""
};
```

The user interface includes sections for group posts, group chat, and member-specific content. React components are used to render these sections dynamically. To display content according to the user and the organization they belong to, conditional rendering is used to display content based on user roles and group memberships which are restricted to only specific groups or users.

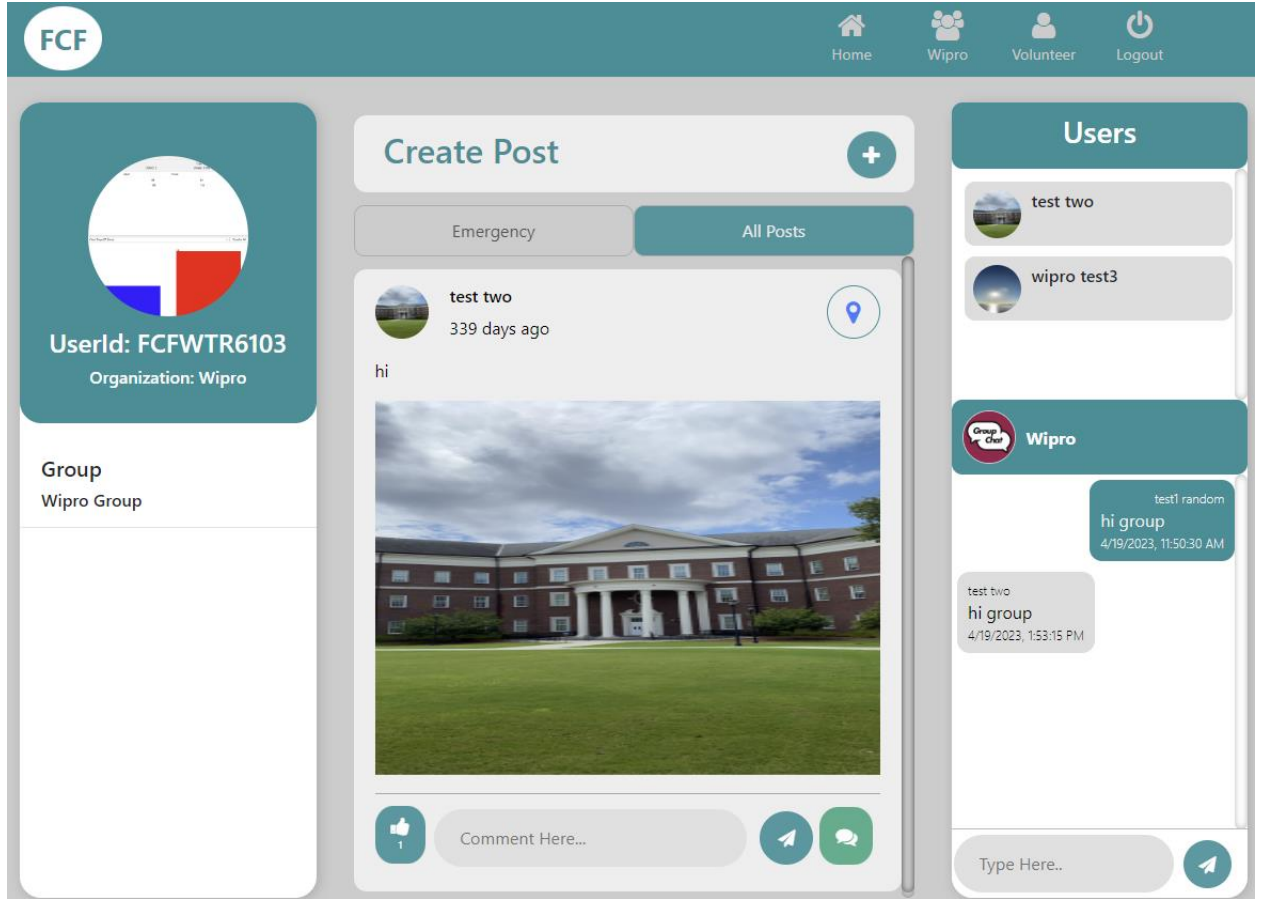


Figure 12: Group/Organization Page

5.5. VOLUNTEERING OPPORTUNITIES/ EVENTS PAGE

According to the requirements gathered from the users regarding how the opportunities should be listed on the platform to make it easy and cover all the essential elements required to know about the event or volunteering opportunity, the frontend components are kept like that of the user dashboard, but it includes only two panels. The left panel contains the same elements as mentioned previously for user dashboard that is details of the user. The right panel of the page includes the opportunities which are either displayed as normal text, as an image or a video. The user has the options to hit on the button

named as “Interested” if the user is interested to attend any event posted on the platform or if they want to participate in any of the volunteering opportunities.

Simultaneously the user has the option to create an opportunity for any volunteers or create any team. The details required to create an event or volunteering opportunity are:

- A brief text description of what the event or opportunity is about,
- An image or video related to the event or describing the event or the volunteering opportunity.
- Date of the event or volunteering opportunity
- Time of the event or volunteering opportunity
- Location of the place where the event or volunteering takes place.

The volunteering/events page in Feed Cape Fear, developed using React for frontend and Express.js for backend, facilitates user interaction with upcoming events and volunteer opportunities. With an intuitive UI design and API endpoints, users can view event summation, registrations for events and an application for the volunteering positions. The backend system uses MySQL to store data, Web Tokens for authentication, and SQL queries for effective data access. Through having an integrated system, users will experience ease of use, organization of events, a smooth registration process that contributes to establishing unity in the community and the active involvement in Feed Cape Fear's projects.

Figure 13 explains the above-mentioned details required to create an event or volunteering opportunity.

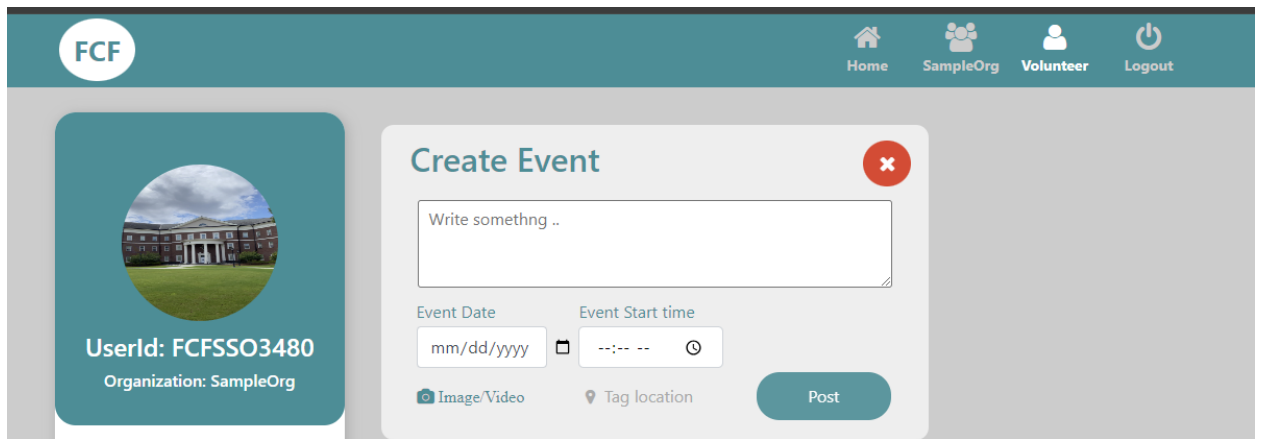


Figure 13: Volunteering page showing the create event option.

Figure 14 shows us the events or volunteering opportunities page of the Feed Cape Fear Platform.

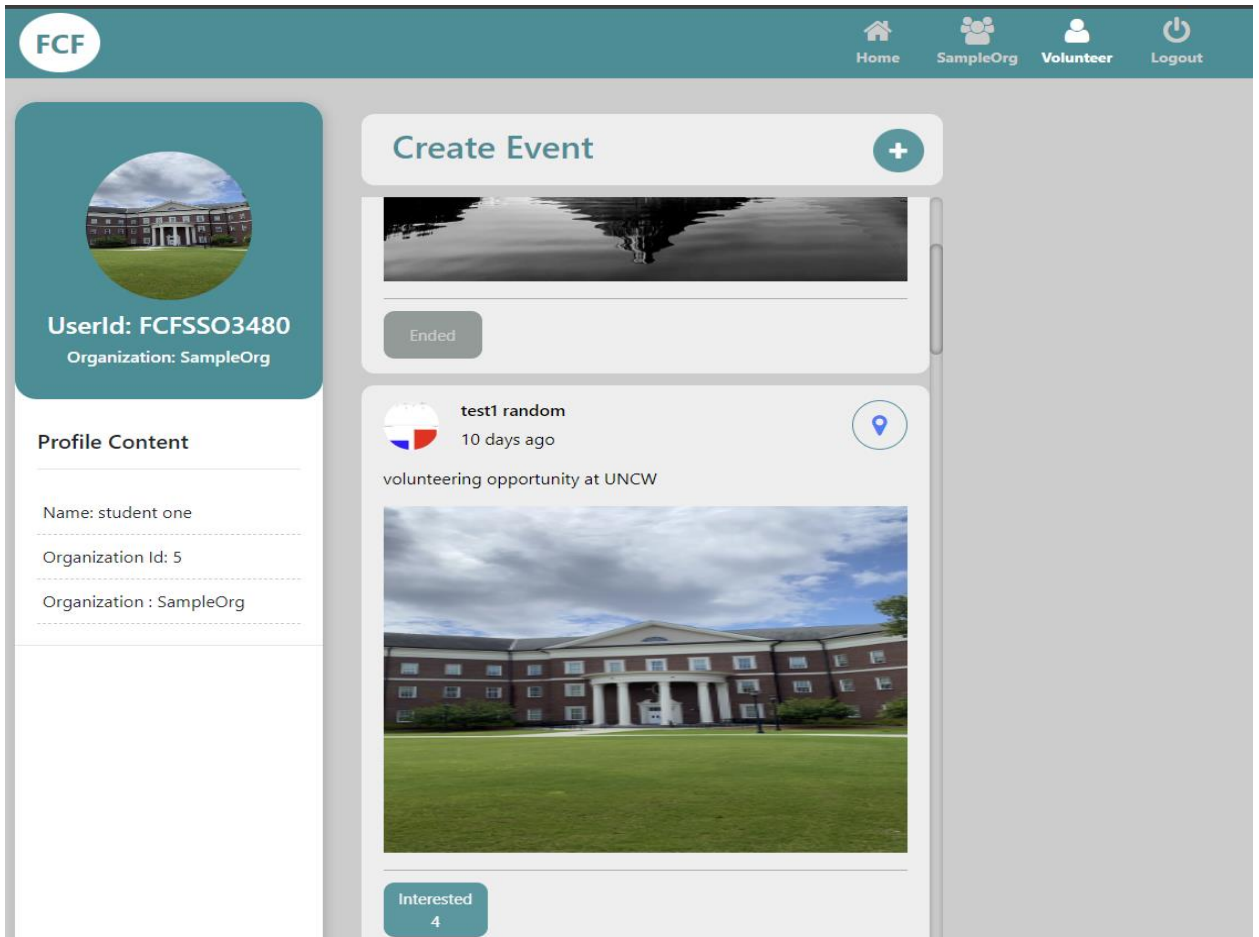


Figure 14: Volunteering/ event page

Figure 15 shows us the list of users who are interested in attending a particular event posted by a test user. The figure also shows the date and the time at which a particular user has clicked on the interested button.

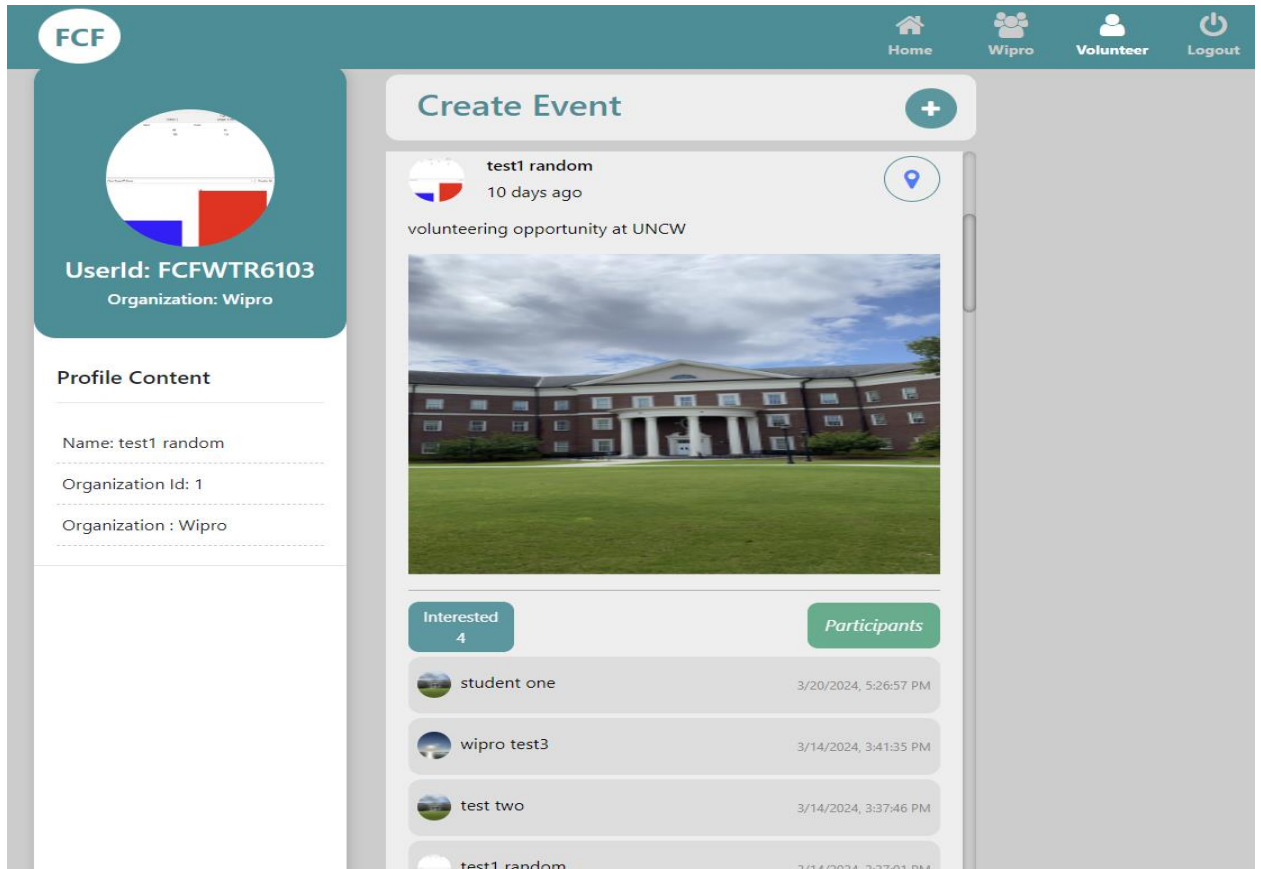


Figure 15: Displaying the List of participants interested in the event or opportunity.

5.6. LOGOUT

In the navigation bar the last option for the user is to log out of the platform. Clicking on this button allows the user to log out of the application which also clears the data in the browser's local cache. A request is sent to the backend and that is powered by Express.js for handling routes and authentication middleware. Being able to perform this request, backend invalidates a user's session token and clears out any session data that is stored. The front end using React will bring a new UI which will show the logged-out state and eventually the user will be redirected to login page. This procedure, among others, helps to maintain the proper handling of sessions, and therefore, the security of the application is guaranteed.

CHAPTER 6: TESTING

Documenting the testing process for the Feed Cape Fear project involves several key steps to ensure thorough coverage of testing activities. As a single member developing this project, I had to make sure I take simple and yet effective steps to test Feed Cape Fear platform and ensure everything is running according to user requirements without any hassle. Each stage of testing was documented to ensure the quality and functionality of the application. The outline of the testing processes for Feed Cape Fear is briefly described in the subsections below:

6.1. MANUAL TESTING

This type of testing involves testing individual components manually when developed completely, such as React components for UI and Express routes for backend logic. This ensures that each unit of code performs as expected and handles edge cases (Scenarios or inputs that are extreme ends of what is considered as normal) successfully.

Table 8 below shows the different scenarios in which manual testing was done for user registration and login to Feed Cape Fear:

Test Case	Expected	Result	Action
User not able to register successfully with the system if they are not providing all the	User should not be able to register with incomplete information in the given registration form.	As Expected,	

required details in the registration form.			
The password should be at least six letters in length.	User should be shown a warning message if the password entered is less than six words.	Not as Expected,	Corrected the code to show up an error message when user doesn't meet criteria for password.
User should have the freedom to upload any professional picture of their choice with any kind of resolution.	User image uploaded is compressed accordingly irrespective of the original size and added to Feed Cape Fear.	As Expected,	
User should not be able to login if they provide invalid credentials.	An error message is popped saying the user has entered invalid credentials.	As Expected,	
User should not include any special characters or	An error message should pop up informing the user to	Not as Expected	Corrected the appropriate filtering code to

numbers in their first and last name.	not include special characters or numbers in their first and last name.		make sure no special characters or numbers were included in the user first or last name.
User is given an autogenerated ID which is unique for every user of Feed Cape Fear upon registering to the platform.	ID with a format of “FCF” + “First alphabet of organization” + “First alphabet of first name” + “first alphabet of last name”.	As expected,	
User can successfully login with their appropriate credentials	Successful login will lead user into user home dashboard.	As Expected,	
User cannot login to the platform until admin approval	User will not be able to login immediately after registration or	As Expected,	

	before admin can approve the user.		
--	------------------------------------	--	--

Table 8: Manual testing for user registration and login

Table 9 below shows the different scenarios in which manual testing was done on the user dashboard on the Feed Cape Fear platform:

Test Case	Expected	Result	Action
Dashboard widgets are displayed accordingly to users	User dashboard displays all the information accurately to users.	As Expected,	
User should be able to create a post.	User can create a post and mention it as emergency or normal and once done the priority is assigned to the post based on which category of post.	As expected,	
User should be able to navigate through	User can click on any username in the chat	As Expected,	

chats with all the users on the platform.	window and chat with them.		
Users should be able to communicate with any user of the platform.	User can click on any user showing up above the chat window to communicate with them.	Not as expected	Corrected the code such that all the users are listed in the chat window and made available to communicate with.
User should be able to like any post or comment on it.	User can like or comment on post which is visible under the post.	As Expected,	
User should be able to navigate to the groups/organizations page and Events page from the home screen.	User is redirected to groups/organizations page upon clicking the groups icon and they are navigated to events page on clicking the events button.	As Expected,	

Table 9: Manual testing for user dashboard

Table 10 below shows the different scenarios in which manual testing was done on groups/organizations page.

Test Case	Expected	Result	Action
User dashboard in group page should be accurately displayed as per user.	User who belongs to a particular organization, their group page only includes their own organization users and content.	As expected,	
Users should be able to find all the other members of the same organization in the group page and send messages on the group/organization chat.	Users can find all the other members of their organization and send messages on their group chat.	Not as expected,	Corrected the code to show the list of all users from the organization to communicate with.
User should be able to post on their group/organization page either as an	Users can create a post on group/organizations page and choose either to post it as an	As Expected,	

emergency or normal post.	emergency post or as a normal one.		
Users should be able to like and comment on any group/organizations page.	Users can like or comment on any post on group/organizations chat and they can see in the bottom of the post.	As Expected,	

Table 10: Manual Testing on Groups/Organizations page

Table 11 below shows the different scenarios in which manual testing was done on the events/volunteering page.

Test Case	Expected	Result	Action
User dashboard in volunteering/events page should be displayed accurately.	User can find appropriate events or volunteering opportunities to participate.	As Expected,	
User should be able to create event/opportunity will all the required details filled in.	User can create an event with the event date/time and the description of the event.	As Expected,	

<p>User should be able to mark their interest on any event open for participation.</p>	<p>User can click on the “Interested” button and confirm their participation for the event or opportunity.</p>	<p>Not as Expected,</p>	<p>Due to continuous version change in libraries on visual studio code, this feature works when a version it supports works on. This will be seen working on deploying onto a server.</p>
<p>Users who created an event or opportunity should be able to view the list of participants who are interested in the event.</p>	<p>Users can click on the “list of participants” button to view who are interested in the event.</p>	<p>Not as Expected,</p>	<p>Corrected the code to show the list of users who clicked on Interested button for a particular event, also the due to the version issues from the previous mentioned test case when an event shows as “ended”</p>

			<p>the participants list will not be shown.</p> <p>This will be working fine once deployed on a server.</p>
--	--	--	---

Table 11: Manual testing for volunteering/events page

Table 12 below shows us the different scenarios in which manual testing was done regarding the admin side controls on the platform:

Test Case	Expected	Result	Action
Admin should be able to login with the proper administrator credentials.	Admin can login using a Unique ID generated for admin and password.	As Expected,	
When an admin login to the application, they are directed to their appropriate screen.	Admin after login can view a special designed dashboard where they can find a list of users who are waiting for	As Expected,	

	approval into the system.		
Admin should be able to approve or decline the requests from the users.	Admin can click on the approve if they find the user valid or they can click on decline button to reject the user request.	Not as Expected,	Few users who registered were not showing up due to issue in query, resolved issue to display all the users who have registered to the platform and waiting for approval from admin.
Admin should be able to log out of the application	Admin can click on the log out button and is directed to login screen	As Expected,	

Table 12: Manual testing on admin controls

Table 13 below shows us the different scenarios in which manual testing was done regarding the real-time communication on the platform:

Test Case	Expected	Result	Action
-----------	----------	--------	--------

<p>User should be able to click on any user of the platform and the chat window should pop up.</p>	<p>User can click on any user which is displayed on the right side of the screen to start a chat conversation.</p>	<p>Not as Expected,</p>	<p>Every user on the platform was not visible on the chat window, corrected the code and the query to display all users available for communication.</p>
<p>User should be able to send a text, or image or video to the other user on the platform through the chat window.</p>	<p>User can share a message through text, image, or video in the chat window.</p>	<p>Not as Expected,</p>	<p>User was not able to send a video and was taking more time to load on the chat window, corrected the code, videos can be sent in chat window easily and efficiently.</p>
<p>User when clicks on any user should be able to see their previous conversation in the chat window.</p>	<p>User can see the previous conversation with any user when they have clicked on any of the user from the</p>	<p>As Expected,</p>	

	right panel of the dashboard.		
--	-------------------------------	--	--

Table 13: Manual Testing on real-time communication

6.2. INTEGRATION TESTING

Feed Cape Fear involves many interactions between different modules, or systems. The primary goal of performing this kind of testing on Feed Cape Fear is to ensure that individual components work together as expected when they are integrated. The main goal is also to detect any issues related to data flow, communication, or functionality across the system. In this section we apply these testing scenarios into Feed Cape Fear and find out if everything is working as expected.

Table 14 below shows us the different scenarios of how integration testing was conducted on Feed Cape Fear:

Test Case	Scenario	Expectation	Result	Action
User Registration and Authentication Integration	Verify that the user who has registered successfully can log in with their	User who has been successfully registered into the system, now	As Expected,	

	unique ID and password.	can get into log in screen and get into application with required details.		
Admin Approval Integration	Ensure that a registered user cannot log in until approved by admin	When user registers and immediately tried to login without approval from admin, the access is denied, showing a message saying, “waiting for approval”.	As Expected,	
Error handling Integration	Testing how the system can handle errors such as invalid login credentials	Verify the system displays appropriate error messages and handles exceptions.	Not as Expected,	Included error handling messages to certain features reminding the user to enter

	or incomplete information.			information appropriately.
--	----------------------------	--	--	----------------------------

Table 14: Integration testing on Feed Cape Fear

6.3. USER FEEDBACK

As a part of the testing process, Feed Cape Fear was reintroduced to the stakeholders fully developed according to their requirements mentioned in the initial stages of preliminary research for Feed Cape Fear. Users who had the opportunity to test Feed Cape Fear had a positive response. They had an intuitive and easy to navigate user interface, it was convenient for them to access various features and functionalities. They appreciate the organized layout, which allowed them to quickly find relevant information and engage with the platform efficiently.

The other things highlighted by the users are the group/organization page, which provided a centralized space for collaboration and communication within specific groups. Users mentioned that this feature enhanced the sense of community and made it convenient to connect with like-minded individuals or organizations. User Dashboard attracted their attention with a simple and effective design. The integration of real-time communication tools also added value, facilitating seamless interactions and collaboration among users.

CHAPTER 7: DEPLOYMENT

The deployment process for Feed Cape Fear involved transitioning the application from the local development environment to a remote server. This chapter provides detailed information of steps taken, challenges faced, and solutions implemented during the deployment phase. The server on which the application is to be hosted was provided by the university.

7.1. INITIAL SETUP AND PREPARATION

This phase involved several key steps to ensure a seamless transition from the local development environment to the production server. This phase was critical as it lays the foundation for the entire deployment process and sets the stage for a successful launch of the application.

- **Development Environment Configuration:** For the development environment I used the visual studio code from my own system and connected to the server using VPN and made sure I was able to access the files on the server from visual

studio code by providing the path to the files on host server. This way I made sure I could edit the code anytime for any improvements or error correction from my own system and could still reflect on the host server.

- Database configuration: Configured MYSQL database locally on the server to store and manage application data. It also ensures data integrity and efficient retrieval.
- Node.js configuration: Installed Node.js locally on the server to make sure we could install the necessary backend and frontend tools and made sure there are no errors on existing code.

7.2. FILE TRANSFER AND SETUP

Once the server was ready for setup, there were two essential tasks:

- File transfer: uploading all the application files which include the codebase, configuration files and assets to the server. This was done when I had connected my own system with the server through virtual private network (VPN).
- Environment Configuration: Configuring environment variables, database connection settings, and server ports to ensure proper functioning of the application.

7.3. DEPENDENCY INSTALLATION AND DATABASE MIGRATION

Running the “npm install” to install all required npm packages and dependencies listed in the package.json file. Later the typescript code is compiled and optimized the application for production.

Database Schema was migrated into the server and made sure the database is synchronized. Later initial data or seed data was imported into the database to populate essential table and records.

7.4. CHALLENGES FACED AND SOLUTIONS

During the deployment process, several challenges were encountered:

- Had to face some compatibility issues between different versions of Node.js, npm packages and MYSQL on the server and local environment posed challenges.
- Finetuning the MYSQL configurations such as connection pooling, indexes for optimal performance was a bit challenging.

The main solution which was followed throughout was to maintain consistency between local and server codebases, so that the deployment process was smooth enough.

CHAPTER 8: FUTURE WORKS

In this chapter, we delve into the potential enhancements and expansions planned for Feed Cape Fear. This section explores the roadmap for further development and improvements to the platform, aiming to address emerging needs, incorporate user feedback, and stay aligned with evolving technological trends. More work in the future implies a lot that starts with improving current features so that users can enjoy the best experience to date and additionally exploring new ideas that can completely change the platform and have a greater impact on community engagement and food distribution in the Cape Fear area.

This platform is designed to facilitate seamless transition and efficient development for future enhancements and features. Some key points which make it easy for future enhancement include a Readme file which provides an overall overview of the project structure, setup instructions. The database has a dedicated field for admin identification, with the ID always set to one. This allows easy differentiation of admin-related role and the credentials for admin are easily accessed through the same ID, which enables developers to log in as admin and perform any necessary changes for the future work. Making sure to have the necessary development environment setup, including Node.js, npm packages, and the code editor (i.e., Visual Studio Code). Following the instructions in the readme file to install all the dependencies, configure the environment variables and to set up the local development server. These steps mentioned can help the developers seamlessly transition into the project, contribute effectively to future development works. The following enhancements are drawn from user feedback in terms of improving this platform in terms of staying updated to the always evolving technological trends.

- **Enhanced Security Measures:** Feed Cape Fear can become more secure in terms of just relying on the unique ID for the user by introducing the two-factor authentication for user login. This can significantly improve the security of user accounts on Feed Cape Fear platform by safeguarding sensitive information and ensuring a user secure environment.
- **Profile Update:** Feed Cape Fear can be improved to include the profile update option to the users post-registration. This adds a layer of customization and flexibility, which not only allows users to keep their information accurate but also stay up to date with the platform.
- **Email Notifications:** Users can be notified on whether their registration is accepted or declined by the admin through email notifications. This can foster a seamless onboarding process.
- **Using Polls on post:** Including polls as an option in post types enables users to gather quick feedback and insights from the community efficiently, which enhances the engagement and facilitates decision making process.
- **Voice Chat Option:** Voice chat option can be an additional mode of communication, where the goal is to promote more expressive and real-time interactions within the groups and communities.
- **Priority Timer for Emergency Posts:** Assigning a timer for emergency posts so that they can adjust their visibility based on the clearance of the emergency can ensure that critical information remains prominent and relevant, enhancing the platform's utility during urgent situations.
- **Add to Calendar feature:** Integrating this feature with the volunteering page on Feed Cape Fear platform can enable users to add their registered on interested

events to calendar, which can improve their organization and give timely reminders for any upcoming events or volunteering opportunities.

- **User roles system with varying privileges:** Administrators manage and approve users with full event management rights. Standard users participate in events but cannot create or manage them. Observers can view events but have no interaction abilities. This setup ensures security, with only verified users creating events, and fosters genuine engagement while preventing unauthorized actions.
- **Organization Management Feature:** This feature in the platform enables administrators to maintain an updated database of organizations. Users can easily submit to new organizations using the platform. This is important to ensure that the platform has all the needed details like name, description, and contact information. Consequently, administrators have the power to do the editing of the organizational profiles, hence the updates of the information or correction as can be seen necessary. Besides the system does away with associations that are purposeless and inactive to maintain the platform's records as clean and accurate as possible. This module is responsible for smooth operation of organizational data management.
- **Deleting/Ban Users from the platform:** The platform comes with a feature that allows admins to manage and control the accounts created. Administrators can remove or delete users who misuse the platform or violate the terms of community. This can keep a secure and secure environment for all users by responding to such behavior or actions at a fast pace. Administrators will view user activities and then take the measures that are necessary to ensure the system remains secure and provide positive user experience.

CHAPTER 9: LESSONS LEARNT

This chapter encapsulates the insights and experiences gathered throughout the development of Feed Cape Fear. The process of creating the project has been an experience that has been made up of trials, successes, and knowledge that has played not only a role in the project itself but also in the characters of the individuals involved in its making. Each turn and step, from conceptualization to implementation, has brought realizations that go beyond technicalities to cover human empathy, agility, security awareness, and collaborative ability. In this process of self-discovery, we examine the underlying values which influenced every step, the most profound encounters that molded our perception, as well as the lessons which light up our way into the future. It's a story of growing and toughening-up with each passing day and of everlasting learning—a tribute to the fact that there are always projects that make a difference.

- **User Centric Design:** Prioritizing the experience of the user and incorporating their feedback iteratively was crucial for creating Feed Cape Fear platform which meets the expectations of users effectively.
- **Security First Approach:** Recognizing the importance of robust security measures early in the development process was essential to safeguard the environment for the users to interact.
- **Effective Communication:** Clear Communication with stakeholders and users was the key element for shaping Feed Cape Fear.
- **Continuous Learning:** Embracing a growth mindset and being open to learning from any challenges or mistakes, and successes has been instrumental in enhancing skills, knowledge, and problem-solving abilities.

A couple of important examples included how we developed the ability to communicate using images and videos in real-time and how we were able to efficiently handle events based on the date and time submitted by users. In the beginning I had not been acquainted with these concepts but, through time I mastered Sharp's use as a quick method of media compression and improved speed during video and image heavy interactions. Moreover, running events using specified dates required using algorithmic approaches that were timely and that ended events at the specified time. Learning and applying the algorithmic approaches greatly enhanced the usability of the application and the users experience.

- **Adaptability and Flexibility:** Being adaptive and flexible in responding to few changing requirements upfront and user needs enabled the project to stay relevant and responsive.
- **Documentation and best practices:** Comprehensive documentation and adherence to best practices in coding, project management, and software development methodologies contribute to maintainability, scalability, and knowledge sharing.

REFERENCES

1. America, F. (n.d.). *MealConnect*. <https://mealconnect.org/>
2. Andersen, E. (2021, February 17). *How should companies communicate CSR on social media?* *Forbes*.
<https://www.forbes.com/sites/forbescommunicationscouncil/2021/02/17/how-should-companies-communicate-csr-on-social-media/?sh=1367867c2c2d>
3. Centers for Disease Control (CDC) and Prevention. *Principles of Community Engagement (1st ed.)*. Atlanta (GA): CDC/ATSDR Committee on Community Engagement; 1997.
4. *Crisis cleanup*. (n.d.-b). <https://crisiscleanup.org/about>
5. GeeksforGeeks. (2024, January 17). *SDLC models Software development models*. <https://www.geeksforgeeks.org/sdlc-models-types-phases-use/>
6. *GivePulse | Giving Platform supporting your community and your causes*. (n.d.).
<https://learn.givepulse.com/>
7. Greene, L. (2024, January 4). *Careit: Turning Food Surplus into Community Solutions*. *Food Tank*. [Careit: Turning Food Surplus into Community Solutions – Food Tank](#)
8. Hope, C. O. (2024, January 31). *Food for People: Insights to Combat Food Insecurity*. Convoy of Hope. <https://convoyofhope.org/articles/food-for-people/>
9. Karczewski, D. (2022, January 17). *Best React.js Alternatives for Front-End development in 2022*. <https://www.ideamotive.co/blog/best-react-js-alternatives-for-front-end>
10. Kirpichny, A. (2024, February 7). *5 best Open Source Databases in 2024*. *Codemotion Magazine*. <https://www.codemotion.com/magazine/devops/5-best-open-source-databases-in-2024/>
11. Klein, M. (2021, September 22). *Flask vs. Django: Which Framework Should You Choose?* Codecademy Blog. <https://www.codecademy.com/resources/blog/flask-vs-django/>
12. Masum, A., Aziz, H. H. A., & Ahmad, M. H. (2020). *Corporate social responsibility and its effect on community development: An overview*. [IOSR Journal of Business and Management, 22\(1\), 35-40](#)
13. McCloskey, D. J., McDonald, M. A., Cook, J., Heurtin-Roberts, S., Updegrove, S., Sampson, D., Gutter, S., & Eder, M. (2011). *Community engagement: definitions and organizing concepts from literature*. In *Principles of community engagement (2nd ed., pp. 3-41)*. NIH Publication No. [11-7782](#)
14. Morello, P. (2023, July 3). *What is MealConnect? Learn about Feeding America's food rescue platform*. *Feeding America*. <https://www.feedingamerica.org/hunger-blog/what-mealconnect-learn-about-feeding-americas-food-rescue-platform>
15. Nextdoor. (n.d.). *Join Nextdoor, an app for neighborhoods where you can get local tips, buy and sell items, and more*. nextdoor.com. <https://nextdoor.com/#neighbors>

16. Pantic, V. (2022, September 8). *Why you need a community engagement platform.* *CitizenLab's Blog.* <https://www.citizenlab.co/blog/civic-engagement/why-you-need-a-community-engagement-platform/>
17. *VolunteerMatch* - where volunteering begins. (n.d.). <https://www.volunteermatch.org/>
18. *What is Community Engagement in a Digital World & How Can You Improve It?* (n.d.). <https://www.powerdms.com/policy-learning-center/what-is-community-engagement-digital-world#engagement-ideas>

APPENDIX A: USE OF ARTIFICIAL INTELLIGENCE TOOLS

During the research phase of “Feed Cape Fear”, various AI tools were employed to gather a few insights about existing solutions and exploring various alternative technologies. AI was utilized responsibly and ethically. No information was copy-pasted; instead, active reading and comprehension to insights from the provided content was done during the research process. The knowledge obtained was synthesized and articulated in my own words to contribute to the development of the project. AI was also used to breakdown complex diagrams from various websites to understand the flow of their platform which helped to build the flow of “Feed Cape Fear”. This project is committed to upholding academic integrity and ensuring that all contributions are original and attributed properly. Table briefs about how AI was used in this project.

Usage	Tool Used	How was the output edited
Analysis of few challenges in existing Solutions and Problems	ChatGPT	Output was actively read and put in my own words.
Analysis of flow of existing solutions.	Bing copilot	Output was edited in my own words and included in the paper.
Researching for alternative solutions for project	ChatGPT	Nothing was taken from the output produced. The solution produced is later researched for more information.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Dr. Avinash Geda for introducing me to the Feed Cape Fear project, which has been the one of the major focus points of my entire master's degree. Dr. Geda's support and directions turned out to be a great starting point for me to acquire the necessary skills to make this project happen.

I am immensely grateful to Dr. Kevin Matthews, for agreeing to be my committee head, and for his unfailing support and the highly valuable advice he offered while developing the project. His experience, backing, and support were paramount in evolving and creating the success of this project. I am also immensely grateful to Dr. Curry Guinn for his teachings and mentorship significantly contributed to the robustness and efficiency of the technical aspects of Feed Cape Fear.

I also want to acknowledge Dr. Jefferey Cummings, the Director of Computer Science and Information Systems, and Information Systems, for his continuous support throughout my master's program. Additionally, I appreciate the initial support extended by Dr. Vetter Ron during my academic pursuits.

I am also grateful to my family and friends for their unwavering support, understanding, and encouragement throughout this challenging yet rewarding journey.