A Project Report On

Fundtastic: Make Fundraising Fun And Easy

Industrial Internship Project report submitted in partial fulfillment of the Requirements for the award of the degree in

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

VANGA NAVYA	**218X1A05I6**
BONIGALA YOGESWARARAO	**218X1A05D9**
VEPURI SMILY	**218X1A05G7**
YARRAM SAGAR	**218X1A05H3**

Under the Esteemed Guidance of

Mr. CH. SAMSONU M.Tech.

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

ACCREDITED BY NAAC WITH 'A' GRADE

(APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA)

NH-5, CHOWDAVARAM, GUNTUR - 522019.

2021 - 2025

KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

ACCREDITED BY NAAC WITH 'A' GRADE

(APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA)

NH-5, CHOWDAVARAM, GUNTUR - 522019.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Industrial Internship Project work entitled **FUNDTASTIC: MAKE FUNDRAISING FUN AND EASY** being submitted by

VANGA NAVYA	**218X1A05I6**
BONIGALA YOGESWARARAO	**218X1A05D9**
VEPURI SMILY	**218X1A05G7**
YARRAM SAGAR	**218X1A05H3**

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering in the Kallam Haranadhareddy Institute of Technology is a record of bonafide work carried out by them.

Internal Guide
Mr. CH. SAMSONU
Associate Professor

Head of the Department
Dr. V. RAJIV JETSON
Professor & HOD

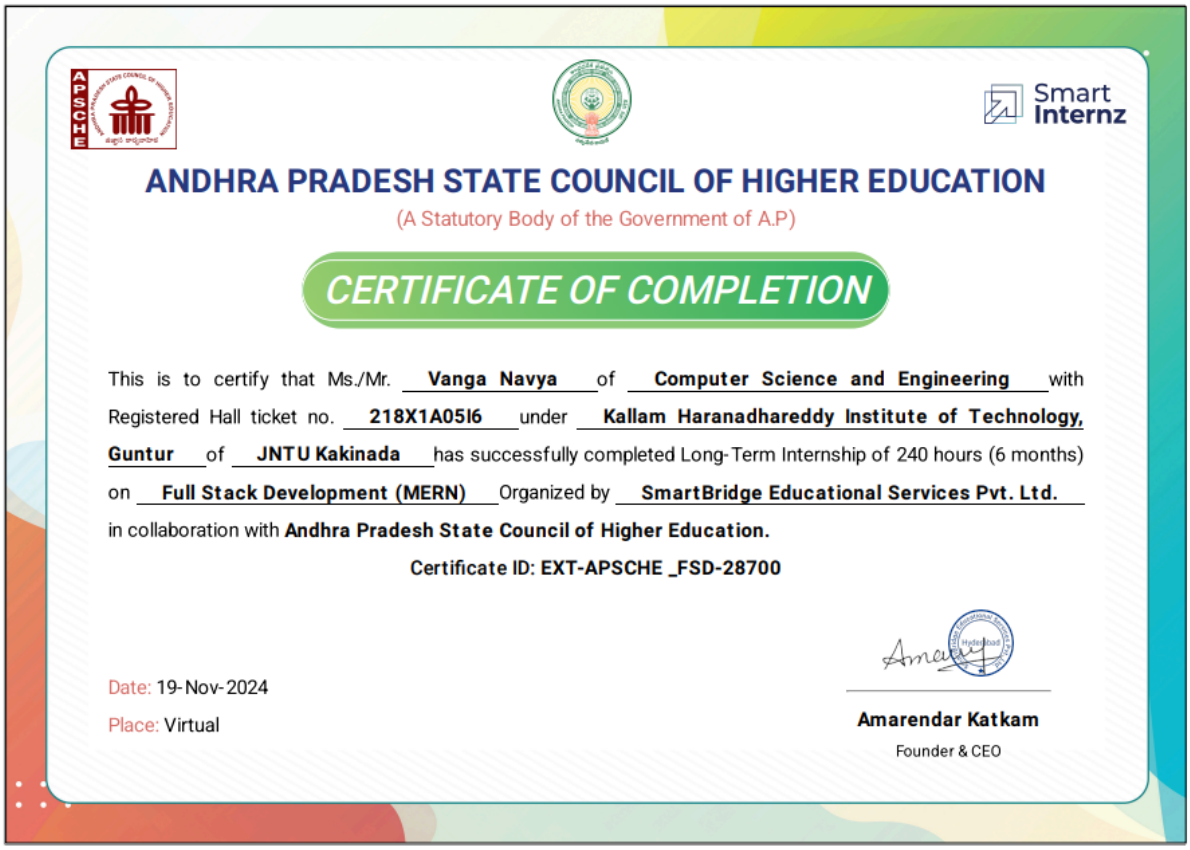
External Examiner

CERTIFICATE FROM INTERN ORGANIZATION

This is to certify that **VANGA NAVYA, BONIGALA YOGESWARARAO, VEPURI SMILY** and **YARRAM SAGAR** Reg.No. **218X1A05I6, 218X1A05D9, 218X1A05G7** and **218X1A05H3** of Kallam Haranadhareddy Institute of Technology underwent industrial internship in SMART INTERNZ from 15-07-2024 to 02-11-2024. The overall performance of the intern during his/her internship is found to be Satisfactory.

VANGA NAVYA	**218X1A05I6**
BONIGALA YOGESWARARAO	**218X1A05D9**
VEPURI SMILY	**218X1A05G7**
YARRAM SAGAR	**218X1A05H3**

Authorized Signatory with Date and Seal



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Vanga Navya** of **Computer Science and Engineering** with Registered Hall ticket no. **218X1A05I6** under **Kallam Haranadhareddy Institute of Technology, Guntur** of **JNTU Kakinada** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

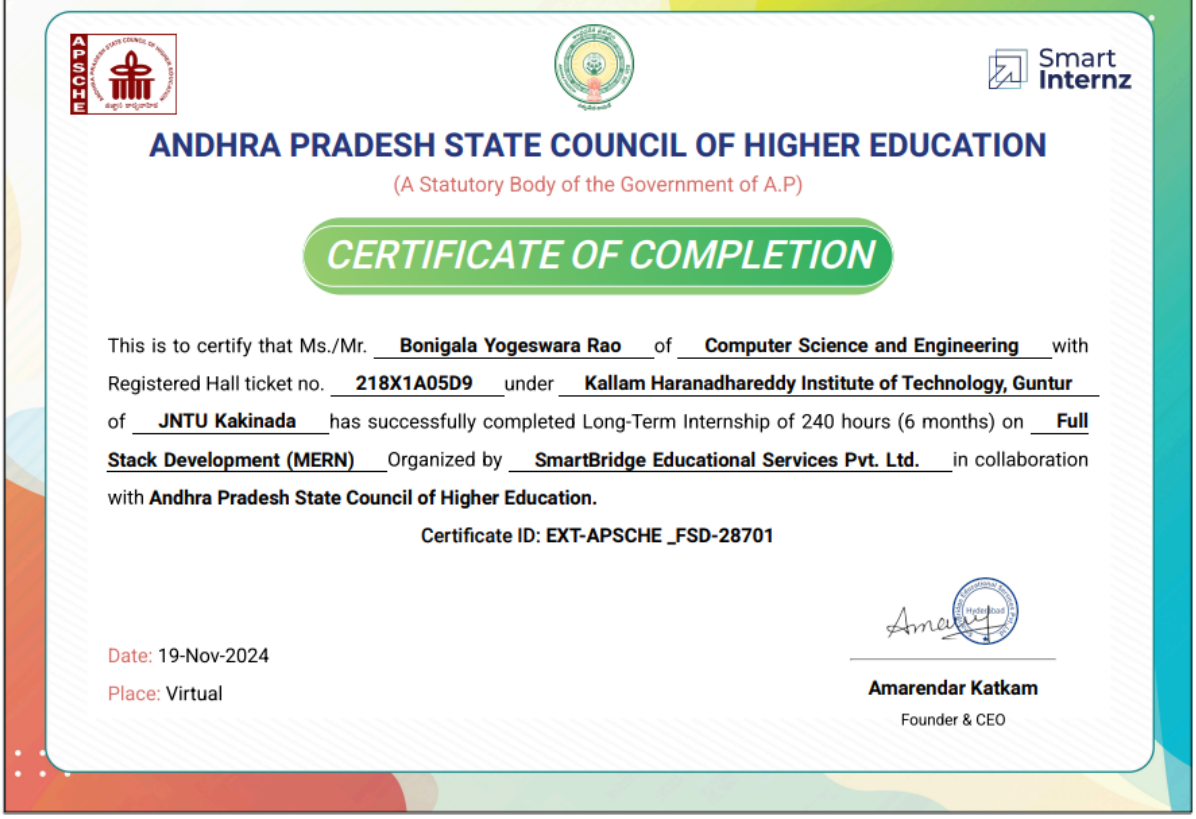
Certificate ID: EXT-APSCHE _FSD-28700

Date: 19-Nov-2024

Place: Virtual

Amarendar Katkam

Founder & CEO



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Bonigala Yogeswara Rao** of **Computer Science and Engineering** with Registered Hall ticket no. **218X1A05D9** under **Kallam Haranadhareddy Institute of Technology, Guntur** of **JNTU Kakinada** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

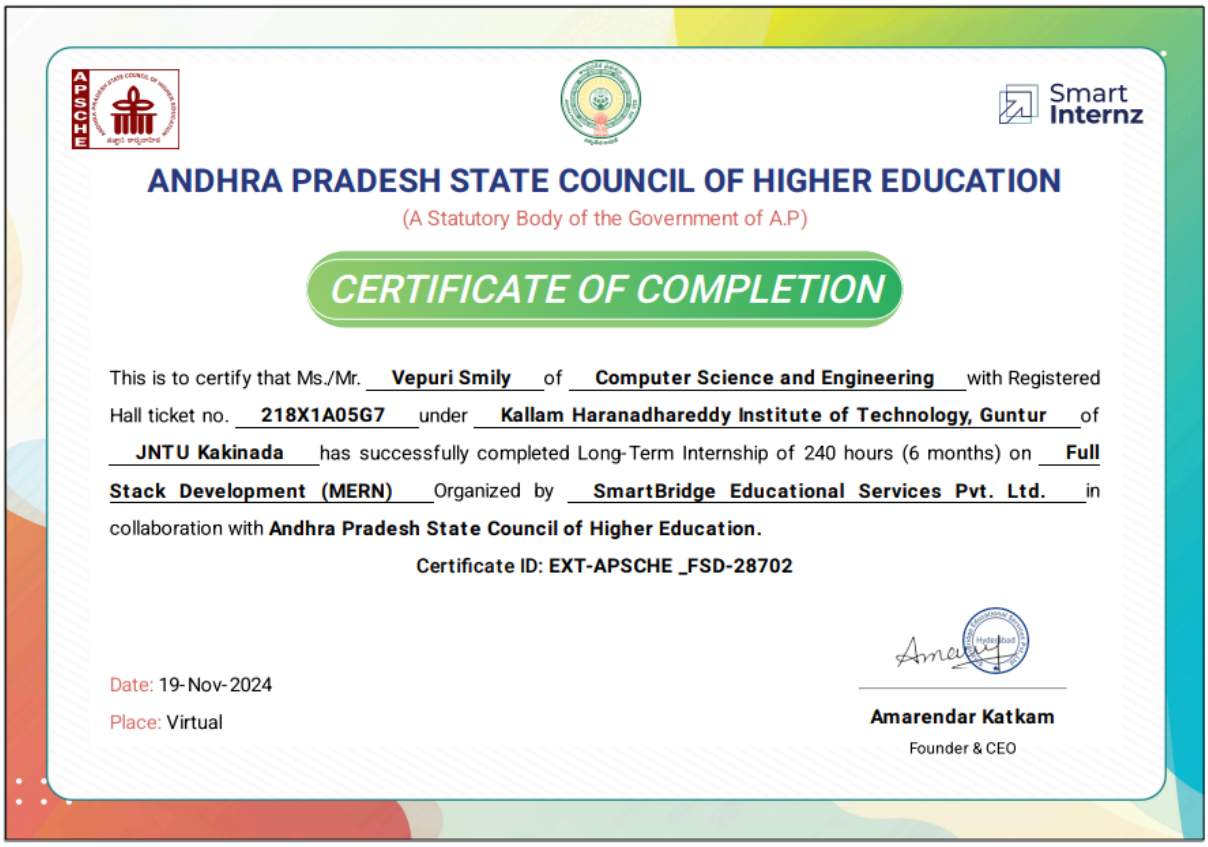
Certificate ID: EXT-APSCHE _FSD-28701

Date: 19-Nov-2024

Place: Virtual

Amarendar Katkam

Founder & CEO



ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Vepuri Smily** of **Computer Science and Engineering** with Registered Hall ticket no. **218X1A05G7** under **Kallam Haranadhareddy Institute of Technology, Guntur** of **JNTU Kakinada** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

Certificate ID: EXT-APSCHE _FSD-28702

Date: 19-Nov-2024

Place: Virtual

Amarendar Katkam

Founder & CEO

ANDHRA PRADESH STATE COUNCIL OF HIGHER EDUCATION

(A Statutory Body of the Government of A.P)

CERTIFICATE OF COMPLETION

This is to certify that Ms./Mr. **Yarram Sagar** of **Computer Science and Engineering** with Registered Hall ticket no. **218X1A05H3** under **Kallam Haranadhareddy Institute of Technology, Guntur** of **JNTU Kakinada** has successfully completed Long-Term Internship of 240 hours (6 months) on **Full Stack Development (MERN)** Organized by **SmartBridge Educational Services Pvt. Ltd.** in collaboration with **Andhra Pradesh State Council of Higher Education.**

Certificate ID: EXT-APSCHE _FSD-28703

Date: 19-Nov-2024

Place: Virtual

Amarendar Katkam

Founder & CEO

DECLARATION

We **VANGA NAVYA (218X1A05I6), BONIGALA YOGESWARARAO (218X1A05D9), VEPURI SMILY (218X1A05G7), YARRAM SAGAR (218X1A05H3)** hereby declare that the project report titled **FUNTASTIC: MAKE FUNDRAISING FUN AND EASY** under the guidance of **Mr. CH. SAMSONU** is submitted in partial fulfillment of the requirements for the award of the degree of bachelor of technology in Computer Science and Engineering.

This is a regard of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project have not been submitted to any other university for the award of any degree.

VANGA NAVYA	**218X1A05I6**
BONIGALA YOGESWARARO	**218X1A05D9**
VEPURI SMILY	**218X1A05G7**
YARRAM SAGAR	**218X1A05H3**

ACKNOWLEDGMENT

We profoundly grateful to express our deep sense of gratitude and respect towards **Sri. KALLAM MOHAN REDDY**, Chairman, KHIT, Guntur for his precious support in the college.

We are thankful to **Dr. M. UMA SANKAR REDDY**, Director, KHIT, GUNTUR for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B. SIVA BASIVI REDDY**, Principal, KHIT, GUNTUR for his support during and until the completion of the project.

We are greatly indebted to **Dr. V. RAJIV JETSON**, M.Tech, Ph.D Professor & Head of the department, COMPUTER SCIENCE AND ENGINEERING KHIT, GUNTUR for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college during Industry Internship.

We are also thankful to our Project Coordinators **Mr. G. MANTHRU NAIK** who helped us in each step of our Project

We extend our deep sense of gratitude to our Internal Guide **Mr. CH. SAMSONU**, M.Tech Assistant Professor and other Faculty Members & Support staff for their valuable suggestions, guidance and constructive ideas in each and every step, which was indeed of great help towards the successfulcompletion of our project.

VANGA NAVYA	**218X1A05I6**
BONIGALA YOGESWARARO	**218X1A05D9**
VEPURI SMILY	**218X1A05G7**
YARRAM SAGAR	**218X1A05H3**

ABSTRACT

A fundraiser application is a web-based platform designed to streamline the process of organizing and participating in fundraising activities. These apps provide users with the tools to create and customize fundraising campaigns, engage with donors and supporters, securely process donations, organize fundraising events, and access valuable data and analytics for campaign performance. Users can also explore nonprofit and cause profiles, access resources for successful fundraising, and leverage social sharing to expand their reach. With a strong emphasis on privacy and security, these applications ensure the safety of users personal and financial information. Fundraiser apps are accessible on mobile devices, offering convenience for campaign management and engagement with donors on the go, ultimately fostering a supportive environment for individuals and organizations seeking to make a positive impact and garner support for their causes.

Keywords: Online Fundraising, MERN Stack, Fundraiser Experience, Campaign create Management, Digital Fundraising, E-commerce Platform, User Centered Design, Funds Raising.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER -1
INTRODUCTION

INTRODUCTION

1.1 Problem Statement

In today's fast-paced economic landscape, many aspiring entrepreneurs and project leaders struggle to secure funding for their innovative ideas. Traditional fundraising methods often lack transparency and efficiency, creating barriers that hinder access to potential investors.

1.2 Scope

A fundraiser application is a web-based platform designed to streamline the process of organizing and participating in fundraising activities. These apps provide users with the tools to create and customize fundraising campaigns, engage with donors and supporters, securely process donations, organize fundraising events, and access valuable data and analytics for campaign performance.

Users can also explore nonprofit and cause profiles, access resources for successful fundraising, and leverage social sharing to expand their reach. With a strong emphasis on privacy and security, these applications ensure the safety of users personal and financial information. Fundraiser apps are accessible on mobile devices, offering convenience for campaign management and engagement with donors on the go, ultimately fostering a supportive environment for individuals and organizations seeking to make a positive impact and garner support for their causes.

Key Issues Include

- **Responsive Design**: The platform's interface is responsive, ensuring optimal usability across various devices, including desktops, tablets, and smartphones. This design approach guarantees that users have a consistent experience regardless of the device they use.
- **User-Friendly Interface**: The front-end is designed with simplicity and ease of navigation in mind. Users can quickly access key functionalities such as campaign creation, investment tracking, and profile management without encountering complex menus or interfaces.

- **Real-Time Updates**: Leveraging React's capabilities, the platform provides real-time updates to users. This feature is crucial for fundraisers to monitor their campaign progress and for investors to track their investments' status without needing to refresh or reload pages.
- **Interactive Components**: The use of interactive components such as forms, progress bars, and dashboards enhances user engagement. These elements provide immediate feedback and visual cues to guide users through various processes on the platform.
- **Secure Authentication**: The front-end integrates secure authentication mechanisms using JWT (JSON Web Tokens) to ensure that user sessions are protected. This security measure helps maintain user privacy and data integrity throughout their interactions with the platform.

1.3 Aim and Objectives

The primary goal of the Fundraising and Investment Platform is to create an innovative and user-friendly online environment that bridges the gap between entrepreneurs seeking funding and investors looking for viable investment opportunities. This platform aims to enhance transparency, streamline the fundraising process, and foster a community of collaboration and support.

To achieve this overarching goal, the project will focus on the following specific objectives:

1. **Develop a User-Centric Platform**
    - Create an intuitive interface for both fundraisers and investors to ensure a seamless user experience.
    - Implement responsive design principles to optimize access across various devices (desktop, tablet, mobile).
2. **Facilitate Efficient Fundraising**
    - Enable users to easily register, create, and manage fundraising campaigns with essential details such as project descriptions, goals, and progress updates.

3. **Enhance Investor Engagement**
    - Provide tools for investors to register, browse, and evaluate fundraising campaigns based on their interests and investment criteria.
    - Allow investors to approach fundraisers directly for inquiries and discussions regarding potential investments.

1.4 Purpose of the Project

The purpose of the project is to develop a comprehensive Fundraising and Investment Platform using the MERN stack (MongoDB, Express.js, React.js, and Node.js) to bridge the gap between individuals seeking funds for their projects and potential investors. This platform aims to provide a user-friendly interface that facilitates efficient communication and transaction management between fundraisers and investors, thereby simplifying the process of raising and managing funds while enhancing investor engagement.

By addressing key challenges such as limited visibility for fundraisers, complex investment processes, and lack of transparency in fund allocation, the platform seeks to empower entrepreneurs by offering them a dedicated space to showcase their projects and engage directly with investors. It also aims to simplify the investment process for investors by allowing them to easily discover, evaluate, and engage with promising campaigns.

Key Features

The key features are mentioned as

- Registration and Login
- Raise Fund Request
- View Raised Requests
- Profile Management
- Logout

CHAPTER-2
LITERATURE SURVEY

LITERATURE SURVEY

2.1 Literature Survey

Literature on crowdfunding platforms, particularly those leveraging blockchain technology, reveals significant insights into the advantages and challenges of such systems.

1. **Decentralization and Security**

   Blockchain technology, particularly Ethereum, allows for the creation of decentralized crowdfunding platforms, enhancing security and transparency in The transactions. Smart contracts automate processes, reducing fraud risk and ensuring that funds are released only when predefined conditions are met [1][5].

2. **User Engagement and Accessibility**

   Platforms developed using the MERN stack facilitate user interactions, enabling fundraisers to connect with potential backers effectively. Such platforms enhance user experience by allowing easy navigation and management of fundraising campaigns [2][3].

3. **Development Insights**

   The development of crowdfunding platforms requires careful consideration of user needs, technical infrastructure, and compliance with regulations to ensure a successful launch and sustained operation [4].

Overall, integrating blockchain technology into crowdfunding enhances transparency, trust, and user engagement while presenting unique challenges that must be addressed during development.

Sources

1. researchgate.net - Secure and Transparent Crowdfunding using Blockchain
2. github.com - Balaji-Ganesh/CrowdHelp-Blockchain-based-crowdfunding-platform
3. researchgate.net - Developing a Social Platform using MERN Stack
4. vlinkinfo.com - Crowdfunding Platform Development Guide: How to Build ...
5. ijsrem.com - Literature Review of Crowdfunding Blockchain
6. sciencedirect.com - A systematic literature review of blockchain-based ...

2.2 Existing System

The current landscape of fundraising and investment platforms is characterized by a variety of solutions, each with its own strengths and weaknesses. While these systems aim to connect fundraisers with investors, they often face significant limitations that hinder their effectiveness. Below are key aspects of existing systems:

1. Traditional Fundraising Platforms

- **Examples**: GoFundMe, Kickstarter, Indiegogo
- **Features**:
    - Allow users to create fundraising campaigns.
    - Enable sharing campaigns through social media.
    - Provide basic analytics for campaign performance.
- **Limitations**:
    - High fees and commissions that reduce the overall funds raised.
    - Limited communication options between fundraisers and backers.
    - Lack of transparency regarding fund allocation and usage.

2. Investment Platforms

- **Examples**: SeedInvest, Crowdcube, EquityNet
- **Features**:
    - Facilitate investments in startups and small businesses.
    - Offer due diligence resources and evaluation metrics for donor.
    - Allow for equity crowdfunding and investments in exchange for shares.
- **Limitations**:
    - Complex processes for both fundraisers and investors, often requiring extensive documentation.
    - Lack of personalized support or guidance for new users.
    - Limited access to niche markets or unique projects, focusing primarily on high-potential startups.

3. Peer-to-Peer Lending Platforms

- **Examples**: Lending Club, Prosper

- **Features**:
    - Connect borrowers with individual lenders.
    - Provide loan options with various terms and interest rates.
- **Limitations**:
    - Primarily focused on loans rather than equity investments.
    - Limited interaction between borrowers and lenders, making it difficult to establish trust.

4. Social Media and Community Platforms

- **Examples**: Facebook Groups, Reddit
- **Features**:
    - Facilitate informal fundraising efforts through community engagement.
    - Allow users to share projects and funding needs within their networks.
- **Limitations**:
    - Lack of structured processes for managing funds and campaigns.
    - Limited tools for tracking progress and reporting on fund usage.

2.3 Disadvantages of Existing System

While existing fundraising and investment platforms offer various features to connect entrepreneurs with investors, they also present significant disadvantages that limit their effectiveness. Below are some key drawbacks:

1. High Fees and Commissions

Many platforms charge substantial fees and commissions on funds raised, which can significantly reduce the amount of capital that reaches the fundraisers. This is particularly detrimental for small businesses or individual entrepreneurs who rely on every dollar raised.

2. Lack of Transparency

Existing platforms often do not provide clear visibility into how funds are used after they are raised. This lack of transparency can lead to distrust among investors, who may hesitate to support campaigns without knowing how their contributions will be allocated.

3. Inefficient Communication

Most platforms do not facilitate direct communication between fundraisers and investors. This lack of interaction can hinder relationship building, prevent timely feedback, and reduce the overall effectiveness of fundraising efforts.

4. Complex User Experience

Many current platforms have convoluted registration and campaign setup processes, making it challenging for users, particularly those unfamiliar with technology, to navigate. A complicated user interface can deter potential fundraisers and investors from engaging with the platform.

5. Limited Support for Niche Projects

Existing systems often focus on mainstream projects and may not provide adequate support or visibility for niche or unconventional ideas. This exclusion can stifle innovation and prevent unique projects from obtaining the funding they need.

6. Insufficient Due Diligence Tools

Although some investment platforms offer evaluation metrics, many lack comprehensive due diligence tools that help investors assess the viability and potential of campaigns thoroughly. This can lead to uninformed investment decisions.

The proliferation of multiple platforms can overwhelm users, as they may find it challenging to determine which platform best suits their needs. This fragmentation can lead to confusion and reduced engagement across the ecosystem.

7. Ineffective Reporting and Analytics

Many platforms fail to provide robust analytics and reporting tools for tracking campaign performance, investment returns, and user engagement. Without these insights, fundraisers and investors may miss opportunities for improvement and optimization.

CHAPTER -3
PROPOSED SYSTEM

PROPOSED SYSTEM

3.1 Proposed System

The Fundraising and Investment Platform aims to create a simple, secure, and transparent way for fundraisers and investors to connect. It will address the problems in existing systems and offer better features for both parties. Below is an outline of the key parts of the system.

1. System Overview

The platform will serve two main types of users: Fundraisers and Investors. Each user type will have specific features tailored to their needs. The platform will be built using the **MERN** stack (MongoDB, Express.js, React.js, and Node.js) to ensure that the system is fast, reliable, and secure.

2. User Interface (UI) and User Experience (UX)

The platform will have an easy-to-use interface, allowing users to navigate easily. Dashboards will provide real-time information on campaign and investment performance.

3. Scalability and Future Enhancements

The platform is designed to grow, with plans to add features like:

- Crowdfunding: Allowing multiple investors to contribute smaller amounts.
- Investment Recommendations: Helping investors find campaigns that match their interests.
- Detailed Reporting: Giving fundraisers more insights into campaign performance and investor engagement.

3.2 Methodology

The development of the Fundraising and Investment Platform follows a structured approach to ensure smooth project execution and delivery. The methodology includes several key phases, each designed to guide the project from initial concept to final implementation.

1. Requirement Gathering

The first step was to gather and define the functional and non-functional requirements of the system. This involved:

- Identifying User Roles: The platform will have two main user roles: Fundraisers and Investors.
- Defining Features: Understanding the key functionalities each user type needs, such as creating campaigns, requesting or offering funds, viewing investments, and secure registration.
- Technology Selection: Choosing the MERN stack (MongoDB, Express.js, React.js, and Node.js) for a scalable, full-stack JavaScript solution.

2. System Design

Based on the requirements, the system architecture was designed. This phase includes:

- Database Design: Creating schemas for Users, Campaigns, Investments, and Activity Logs using MongoDB and Mongoose for efficient data storage and retrieval.
- API Design: Developing RESTful APIs for handling user registration, login, campaign creation, investment management, and fund requests. This ensures smooth interaction between the frontend (React) and backend (Node.js with Express).
- User Interface Design: Designing a user-friendly interface using React, focused on simplicity and ease of navigation for both fundraisers and investors.

3. Frontend and Backend Development

Development of the platform was divided into two parts:

- **Frontend Development:** Using React.js, the frontend provides a clean, responsive interface. Features like campaign creation, user authentication, and viewing investments are implemented. Interactive components such as forms, dashboards, and progress bars were developed to enhance user experience.
- **Backend Development:** The backend was built using Node.js and Express.js to handle the business logic and API endpoints. MongoDB was used as the database to store user, campaign, and investment data. Security features like password encryption (using bcrypt) and authentication (using JWT) were also implemented.

4. Integration and Testing

After developing the frontend and backend, the two were integrated to ensure they work seamlessly together. The system underwent several testing phases:

- Unit Testing: Individual components and functions were tested for functionality.
- Integration Testing: Testing the interaction between different modules like user campaign management, and investment processing.
- Security Testing: Ensuring that sensitive information like passwords and financial details are securely encrypted and protected against vulnerabilities.
- User Testing: Testing the system with real users to gather feedback and identify any user experience issues.

5. Deployment

Once the platform was tested and refined, it was deployed on a cloud-based server, ensuring it's accessible from anywhere. Technologies such as Heroku or AWS were used for hosting, and GitHub was used for version control throughout the development process.

6. Maintenance and Updates

After deployment, the system will be monitored for performance and any bugs or issues that arise. Updates will be made based on user feedback and performance metrics, ensuring the platform continues to run smoothly and scales as more users join.

3.3 Advantages

The Fundraising and Investment Platform offers several key advantages over existing systems, improving the user experience for both fundraisers and investors. Below are the main benefits of the proposed system:

1. User-Friendly Interface

The platform has an intuitive and responsive user interface designed with React.js. Both fundraisers and investors can easily navigate through features like creating campaigns, investing

in projects, and viewing investment status without any technical difficulties.

2. Real-Time Transparency

The platform provides real-time updates on campaign progress, investments, and fund requests. This helps build trust between fundraisers and investors, as they can track where the funds are going and how much has been raised at any given time.

3. Secure Authentication and Data Encryption

User authentication is handled securely using JWT (JSON Web Tokens), ensuring only authorized users can access their accounts. Sensitive data, such as passwords and bank details, are encrypted with bcrypt, providing a high level of security for both fundraisers and investors.

4. Efficient Fund Management

Fundraisers can manage multiple campaigns in one place, while investors can track their investment contributions efficiently. This makes the entire process streamlined and organized for both parties.

5. Scalability and Flexibility

Built on the MERN stack, the platform is highly scalable and can easily accommodate more users, campaigns, and investments as the platform grows. New features can also be added in the future without affecting the current performance of the system.

6. Comprehensive Role-Based System

The system clearly differentiates between two user roles—Fundraiser and Investor—allowing for a customized experience for each. Each user role has access to tailored features, making it easier for them to perform actions relevant to their role.

7. Improved Trust and Transparency

By allowing fundraisers to view which investors have approached them and showing the status of investments, the platform fosters trust and transparency between both parties. Investors can also see campaign details before committing funds.

8. Secure Payment Processing

All financial transactions are processed through secure payment gateways, ensuring that investors' contributions are protected. The system reduces the risk of fraud and ensures that funds reach their intended recipients.

9. Faster Time to Market

Using the MERN stack accelerates the development process, meaning new features and enhancements can be introduced quickly, making the platform adaptable to market needs and user feedback.

10. Data Logging and Tracking

The system logs user actions through the Activity Schema, providing a clear audit trail of activities like investments, campaign creation, and fund requests. This ensures accountability and can help resolve any disputes.

3.4 Feasibility Study

A feasibility study evaluates the practicality and potential success of the Fundraising and Investment Platform by analyzing its technical, operational, and economic aspects. This study helps determine whether the project can be executed successfully within the proposed timeline and budget.

1. Technical Feasibility

- **Technology Stack:** The project is based on the MERN (MongoDB, Express.js, React.js, Node.js) stack, which is well-suited for developing scalable web applications. This stack enables rapid development and supports real-time interactions.
- **Integration Capabilities:** The platform can integrate with various payment gateways for secure financial transactions, enhancing user trust and convenience. Additionally, it can incorporate APIs for data analytics and reporting tools.

- **Security Measures:** Implementing security protocols such as data encryption, secure authentication (JWT), and regular security audits will ensure that user information and financial transactions are protected against breaches.

2. Operational Feasibility

- **User Roles:** The system will cater to two main user roles: fundraisers and investors. Each user will have distinct functionalities, ensuring a tailored experience that meets their specific needs.
- **Resource Availability:** The development team possesses the necessary skills in MERN technologies and project management, which will facilitate the successful execution of the project.
- **Scalability:** The platform is designed to accommodate an increasing number of users and campaigns, allowing for future expansion without compromising performance.

3. Economic Feasibility

- **Cost Analysis**
    - **Development Costs:** Estimating expenses related to software development, hosting, and maintenance.

CHAPTER-4
SYSTEM REQUIREMENTS SPECIFICATION

SYSTEM REQUIREMENTS SPECIFICATION

4.1 Functional Requirements

The functional requirements define the specific operations, behaviours, and features that the **Fundraising and donation Platform** must perform. These requirements are grouped by user roles: **Fundraisers** (Users) and **donors**.

1. User (Fundraiser) Functionalities

Fundraisers are individuals or organizations looking to raise funds for their campaigns. The platform must provide the following functionalities for them:

- **Register**
  - The user can create an account by providing their name, email, and password. An email verification process may be required for security.
- **Login**
  - Users can log in using their email and password. Password recovery options should be available.
  - **Raise Fund Request:** The user can create a new campaign by providing details such as title, description, goal amount, and an image. This campaign can then be viewed and approved by investors.
- **View Raised Requests / Approached Investors**
  - The user can view a list of their campaigns, including details of investors who have shown interest and their contributions.
- **View Profile**
  - Users can view and edit their personal profile details, including name, email, and bank account information for receiving funds.
- **Logout**
  - Users can securely log out of the platform.

2. Donor Functionalities

Donors are individuals or organizations willing to invest in fundraising campaigns. The platform must provide the following functionalities for them:

- **Register**
    - Investors can create an account by providing their personal details and contact information.
- **Login**:
    - Investors can securely log in to the platform with their credentials.
- **Approach for Investments**
    - Investors can browse through campaigns and show interest by pledging an investment. The platform allows them to specify the amount they want to invest in a campaign.
- **View Investments**
    - Investors can view a list of their investments, including details of the campaigns they have contributed to and the current status of each investment (pending or confirmed).
- **Logout**
    - Investors can log out securely from the platform.

3. Admin Functionalities

Administrators manage the platform and ensure smooth operations. The platform should provide the following functionalities for the admin:

- **Approve/Reject Campaigns**
    - Admins can review new campaign submissions and either approve or reject them based on compliance with platform guidelines.
- **Manage Users**
    - Admins can view, edit, or deactivate user accounts, including both fundraisers and investors.
- **View Platform Activities**
    - Admins can view logs of user activities, such as campaigns created, investments made, and profile updates for monitoring purposes.

4. General Functionalities

These functionalities apply to all users, regardless of their role.

- **Email Notifications**
    - The platform should send email notifications for key events like new campaign approvals, investment confirmations, and fund withdrawals.
- **Search and Filter Campaigns**
    - Users (both fundraisers and investors) can search and filter campaigns by categories such as funding goal, current status, or campaign type.
- **Dashboard**
    - Users should have access to a personalized dashboard summarizing their activities (for fundraisers, it shows their campaigns; for investors, it shows their investments).
- **Secure Authentication**
    - All users must authenticate securely, with encrypted passwords stored using methods like bcrypt and access controlled through JWT tokens.

4.1.1 User Authentication

User (Fundraiser) Functionalities

Fundraisers are individuals or organizations looking to raise funds for their campaigns. The platform must provide the following functionalities for them:

- **Register**
    - The user can create an account by providing their name, email, and password. An email verification process may be required for security.
- **Login**
    - Users can log in using their email and password. Password recovery options should be available.
    - **Raise Fund Request**: The user can create a new campaign by providing details such as title, description, goal amount, and an image. This campaign can then be viewed and approved by investors.
- **View Raised Requests / Approached Donors**
    - The user can view a list of their campaigns, including details of investors who have shown interest and their contributions.

- **View Profile**
    - Users can view and edit their personal profile details, including name, email, and bank account information for receiving funds.
- **Logout**
    - Users can securely log out of the platform.

4.1.2 Data Management

The platform should provide the following data management capabilities:

- **Campaign Management**
    - Users can create, edit, or delete campaigns as needed.
- **Investment Management**
    - Investors can view and track their contributions and see the status of their investments (whether confirmed or pending).
- **Profile Management**
    - Users can update their profiles, including personal information and banking details, as needed.

Payment and Financial Transactions

The platform will handle payments and financial transactions securely:

- **Fund Transfer**
    - Once a campaign is completed, the platform should allow the transfer of funds to the fundraiser's bank account.
- **Transaction Status**
    - Users can view the status of each transaction (whether it’s pending, completed, or failed).

4.2 Non Functional Requirements

Non-functional requirements describe the quality attributes, performance metrics, and operational aspects of the **Fundraising and Investment Platform**. These ensure that the platform is secure, scalable, and user-friendly, while also providing a good user experience.

The platform should handle a large number of users, campaigns, and transactions simultaneously without significant performance degradation. It should be able to scale vertically (by upgrading hardware) and horizontally.

1. Performance Requirements

- **Scalability**
  - The platform should handle a large number of users, campaigns, and transactions simultaneously without significant performance degradation.
- **Response Time**
  - Key actions, such as logging in, browsing campaigns, or making investments, should complete within **2 seconds** on average.
- **Load Handling**
  - The system must support at least **1,000 simultaneous users** without affecting the platform's performance.

2. Security Requirements

- **Data Encryption**
  - All sensitive data, such as user credentials and financial information, should be encrypted in transit using SSL/TLS protocols and at rest using appropriate encryption standards (e.g., AES-256).
- **Authentication**
  - The platform must implement secure authentication mechanisms, such as **JWT tokens** for session management, and **bcrypt** for password hashing.
- **Authorization**
  - Role-based access control (RBAC) must be enforced, ensuring users can only access resources and actions allowed by their role (e.g., user, investor, admin).
- **Data Privacy Compliance**
  - The system should comply with relevant data protection regulations like **GDPR** and **CCPA** to safeguard user information.

3. Usability Requirements

- **User Interface (UI)**
  - The platform should have an intuitive, easy-to-navigate interface that is accessible to users with varying levels of technical expertise.
- **Accessibility**
  - The platform should follow **WCAG 2.1** (Web Content Accessibility Guidelines) to ensure it is usable by people with disabilities, including support for screen readers, keyboard navigation, and appropriate contrast ratios.
- **Help and Documentation**
  - Users should have access to documentation and help sections to assist them in understanding and using the platform's features.

4. Reliability and Availability

- **Uptime**
  - The platform should have an uptime of at least **99.9%**, ensuring that it is available for users most of the time without interruptions.
- **Backup and Recovery**
  - Regular backups of all data should be performed to ensure that data can be recovered in case of a system failure. A disaster recovery plan must be in place to restore services quickly.

5. Maintainability

- **Modular Codebase**
  - The code should follow **modular design principles**, making it easy to update or fix components without affecting other parts of the system.
- **Documentation**
  - The platform's code and architecture should be well-documented, enabling easy maintenance and future enhancements.

- **Bug Fixing and Updates**
    - The platform should be structured to allow easy deployment of bug fixes and updates, minimizing downtime and disruption for users.

6. Portability

- **Cross-Browser Compatibility**

  The platform should work across all major browsers, including **Google Chrome, Firefox, Safari,** and **Edge**.

7. Compliance Requirements

- **Legal Compliance**
    - The platform must comply with financial transaction regulations and anti-money laundering (AML) laws to ensure secure and lawful handling of investments.
- **Ethical Standards**
    - The system should follow ethical guidelines for fundraising and investing, ensuring transparency and preventing fraudulent activities.

8. System Integration

- **Third-Party Services**
    - The platform may need to integrate with third-party payment gateways and banks for processing transactions. These services must be securely integrated following standard APIs.
- **API Support**
    - The platform should provide APIs to allow other services to interact with it, such as fetching campaign data or processing payments.

The non-functional requirements ensure that the **Fundraising and Investment Platform** delivers a high-quality user experience while being secure, reliable, and scalable. These requirements are critical for maintaining the platform's integrity and performance over time.

4.2.1 Software Requirements

Operating system	:	Windows 7 or 7+
Scripts	:	JavaScript.
Front End	:	React js, Html, Css
Technology	:	Node Js
Database Connectivity	:	JDBC
Database	:	MongoDbs

4.2.2 Hardware Requirements

Ram	:	8 GB
Hard disc or SS	:	More than 500 GB
Processor	:	Intel 3rd generation or high with 8GB Ram
Software’s	:	Java 8 or high version, Eclipse.

CHAPTER- 5
SOFTWARE DESIGN

SOFTWARE DESIGN

5.1 UML Diagrams

5.1.1 System Architecture

The system architecture of a **fundraising platform** typically refers to the design, structure, and components that enable the platform to support and manage online donations, fundraising campaigns, donor interactions, and related services. The architecture must ensure high availability, security, scalability, and a seamless user experience for both fundraisers (campaign organizers) and donors.

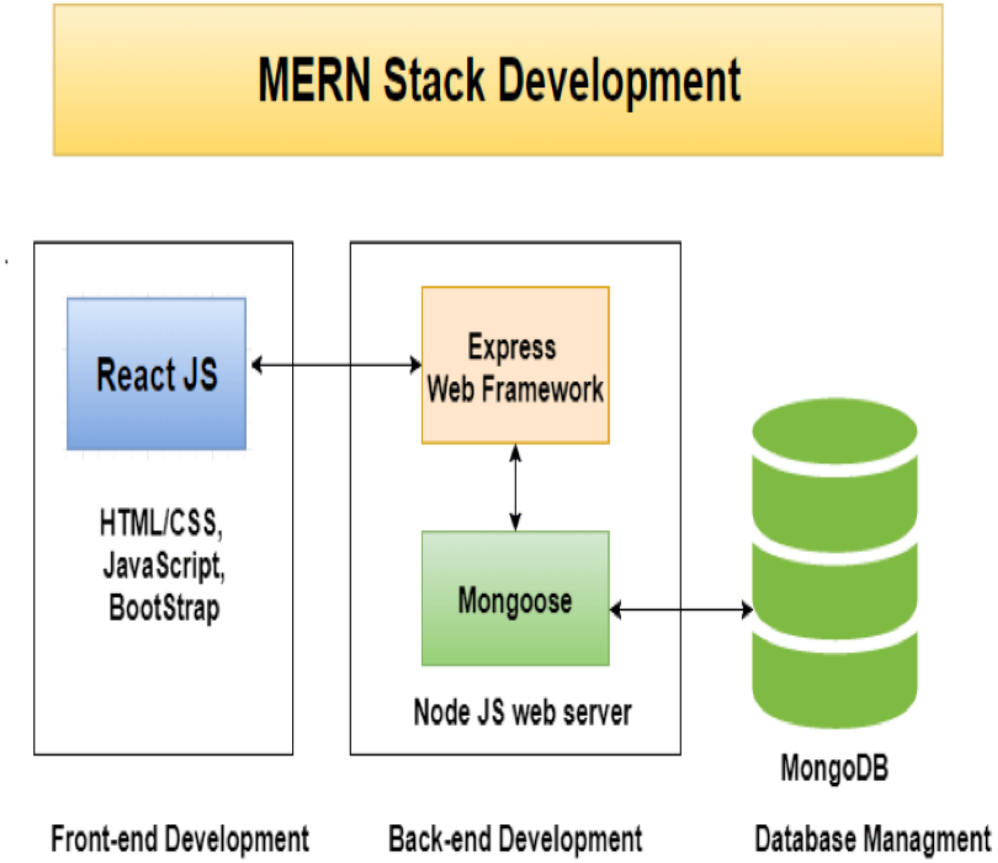


Fig: 5.1.1 System architecture between frontend and backend

5.1.2 ER Diagram

An emergency room (ER) diagram in the context of fundraising can represent the rapid response and resource allocation necessary for urgent fundraising needs, such as disaster relief or unexpected medical expenses. This diagram would typically illustrate key processes, actors, and flows involved in efficiently mobilizing resources and funds.

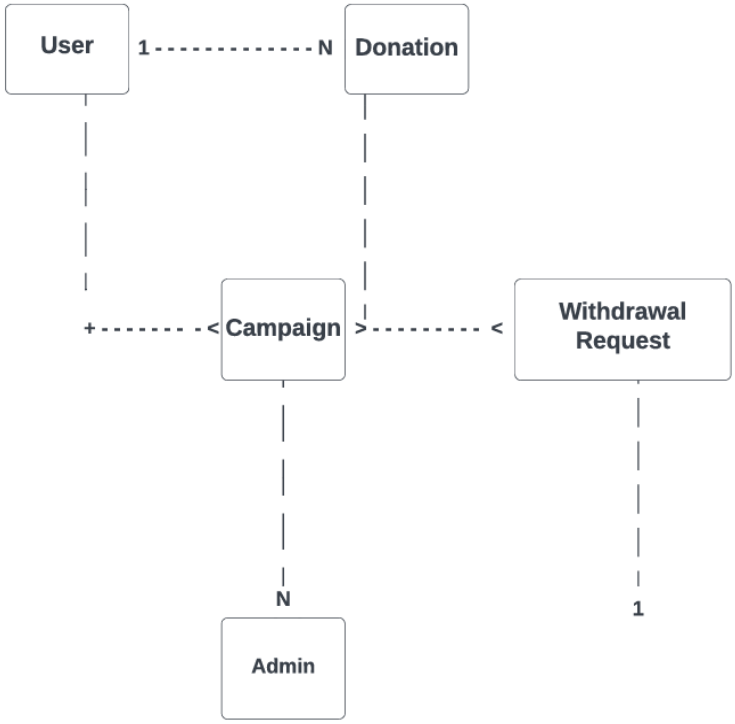


Fig: 5.1.2 ER diagram for Admin and User

5.1.3 Use Case Diagram

The Use Case Diagram for fundraising provides a clear visual representation of how different users interact with the fundraising system. It helps stakeholders understand the essential functionalities and the roles involved in the fundraising process. By identifying these interactions, organizations can better design their systems, improve user experience, and ensure that all necessary functions are effectively addressed. This foundational understanding is crucial for developing a successful fundraising strategy.

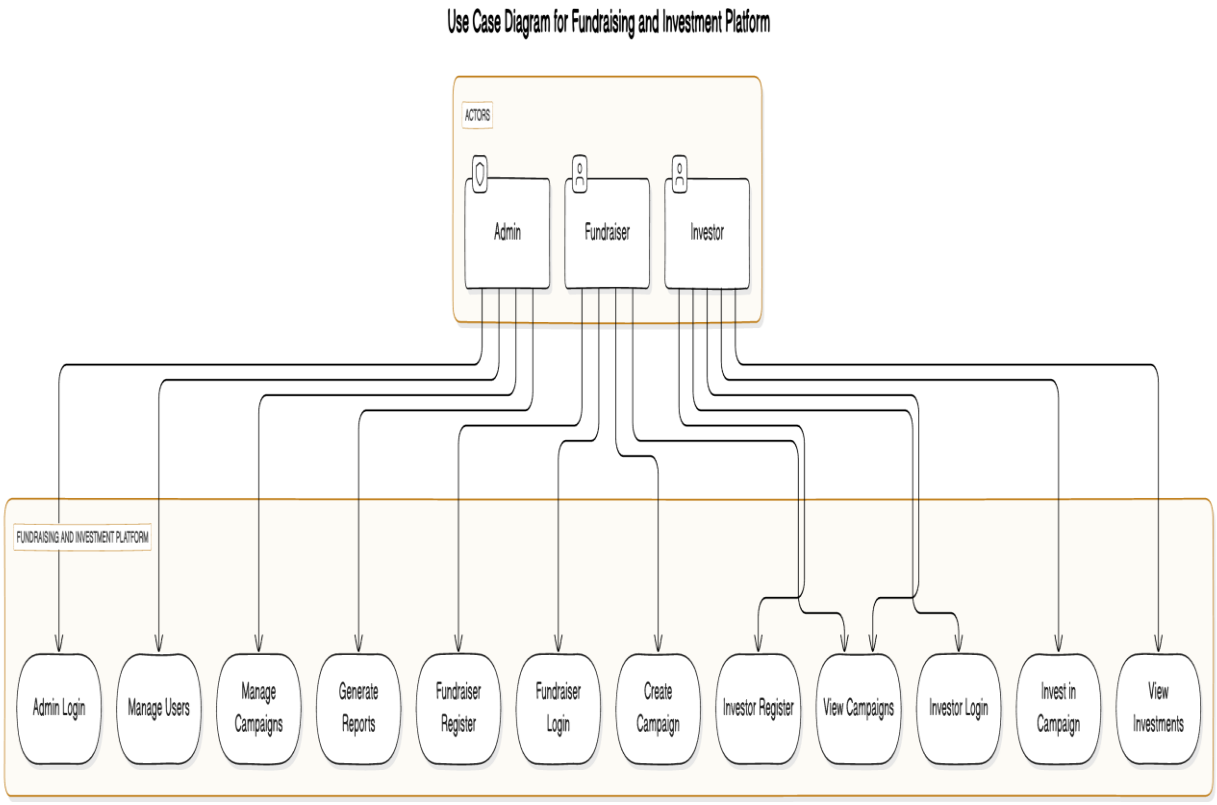


Fig: 5.1.3 Use case diagram for fundraising and investment platform

5.2 Sequence Diagram

A Sequence Diagram for fundraising illustrates the step-by-step interactions between various actors and the system during a specific fundraising process, such as making a donation. It shows the chronological order of messages exchanged between participants, providing clarity on how the process unfolds.

Key Components

1. **Actors**
    - **Donor**: The individual or organization making the donation.
    - **Fundraising System**: The platform or application managing the donation process.
    - **Payment Processor**: The external service handling payment transactions.
    - **Fundraising Coordinator**: Oversees the campaign and tracks donations.
2. **Process Steps**
    - **Initiate Donation**: The donor starts the donation process.
    - **Enter Donation Details**: The donor inputs the amount and payment method.
    - **Submit Donation**: The donor submits the donation request to the fundraising system.

The fundraising process involves a series of coordinated interactions among key participants, each playing a crucial role in successfully securing contributions. It begins with the **donor**, who expresses interest in contributing to a specific campaign, initiating the engagement. The **fundraiser** responds by providing detailed information about the campaign's purpose, goals, and how the funds will be utilized, ensuring the donor feels informed and valued. Once the donor decides to contribute, they fill out a donation form, which can be either digital or physical. This form is then submitted by the fundraiser to the **organization**, which is responsible for managing the funds.

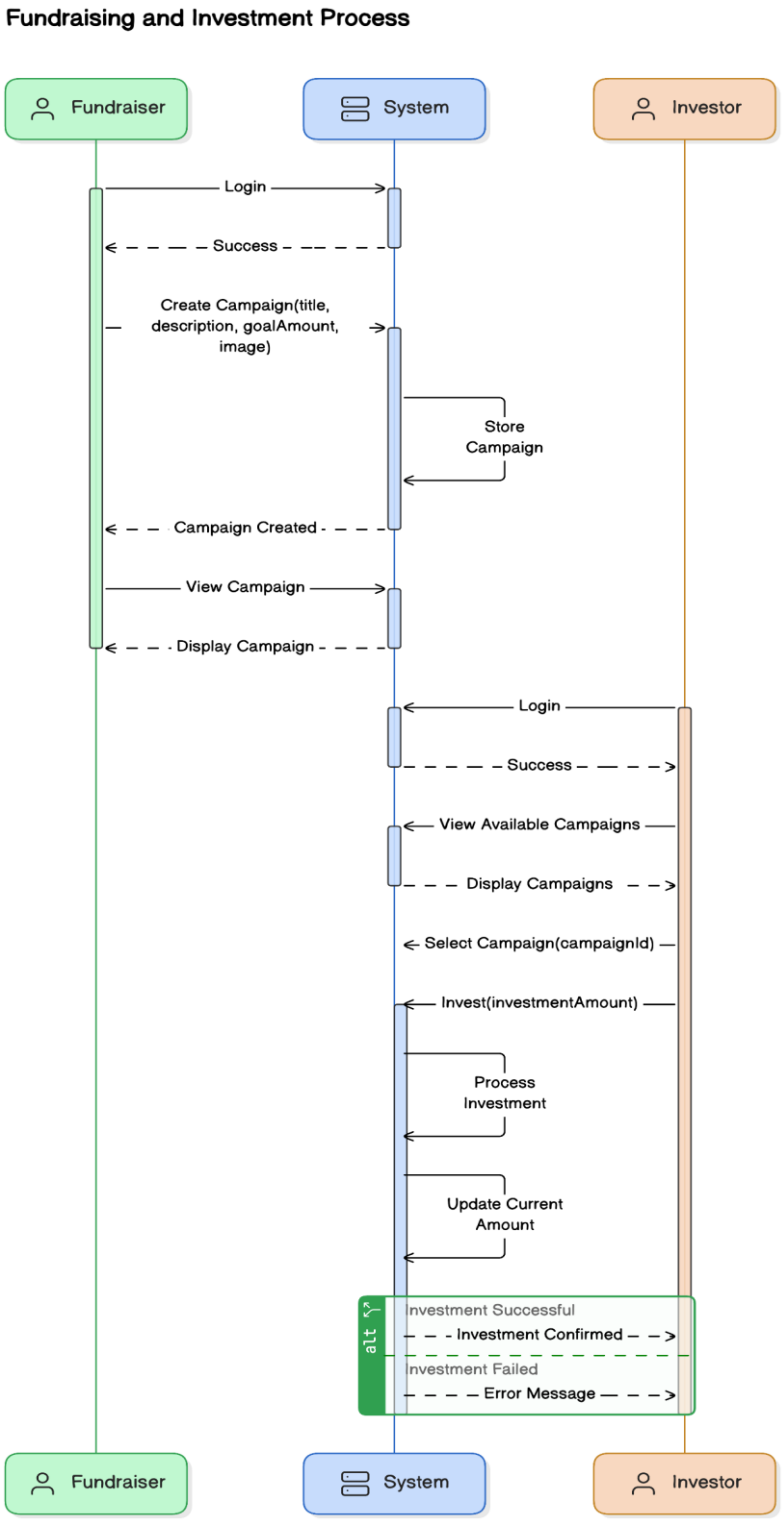


Fig: 5.2 Sequence diagram for fundraising and investment process

5.3 Class Diagram

The Class Diagram for fundraising provides a clear overview of the system's structure, defining how different entities interact and the roles they play in the fundraising process. By visually mapping out classes, attributes, and methods, stakeholders can better understand the system's functionality and organization.

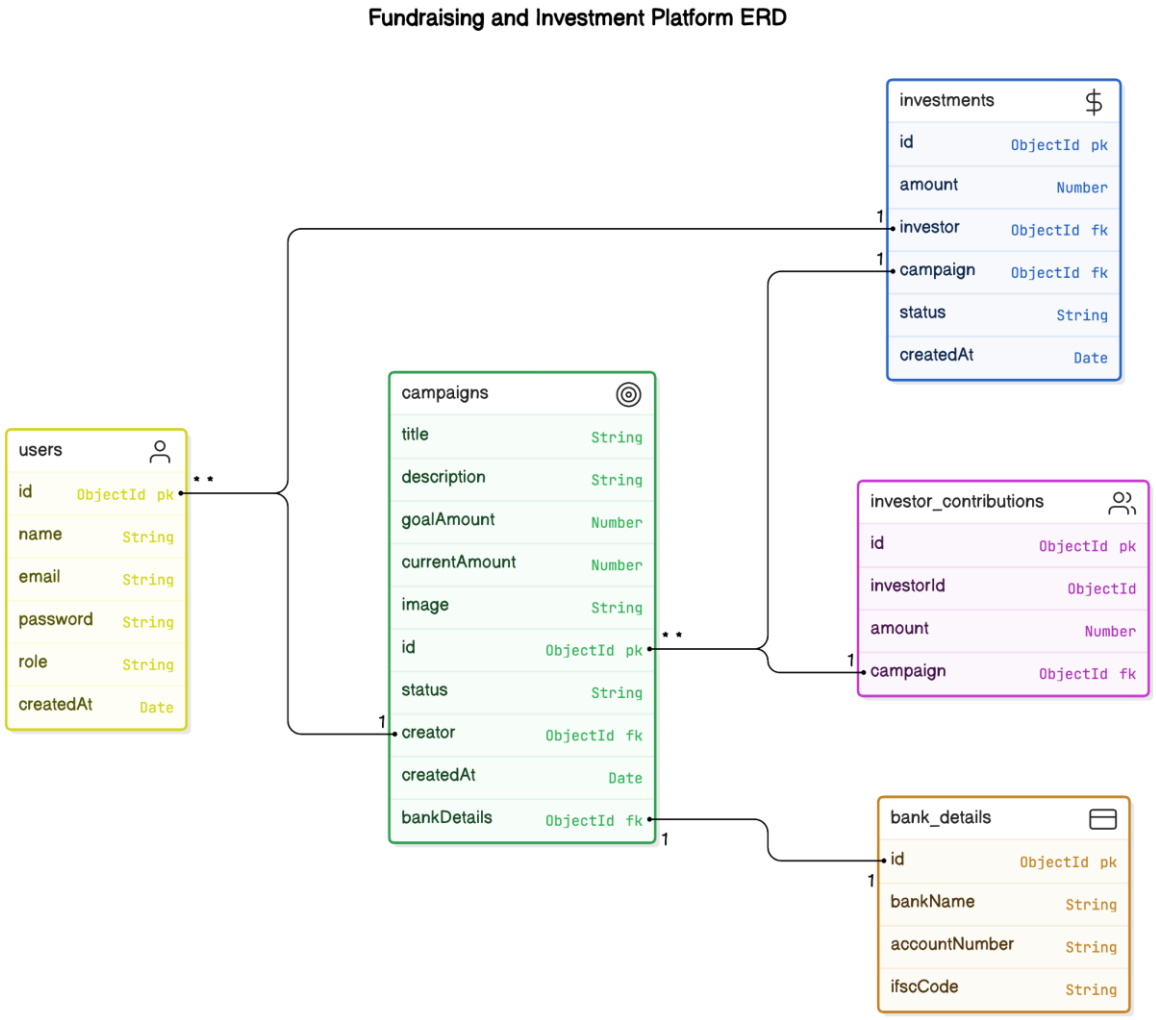


Fig: 5.3 Class diagram for fundraing and investment platform ERD

5.4 Activity Diagram

The **fundraising process** involves outlining the different actions and decisions taken by the fundraising system. The process generally includes activities like creating a campaign, donating to a campaign, processing payments, and reviewing campaigns. Below is a detailed **activity diagram** description for the core activities in a typical fundraising system.

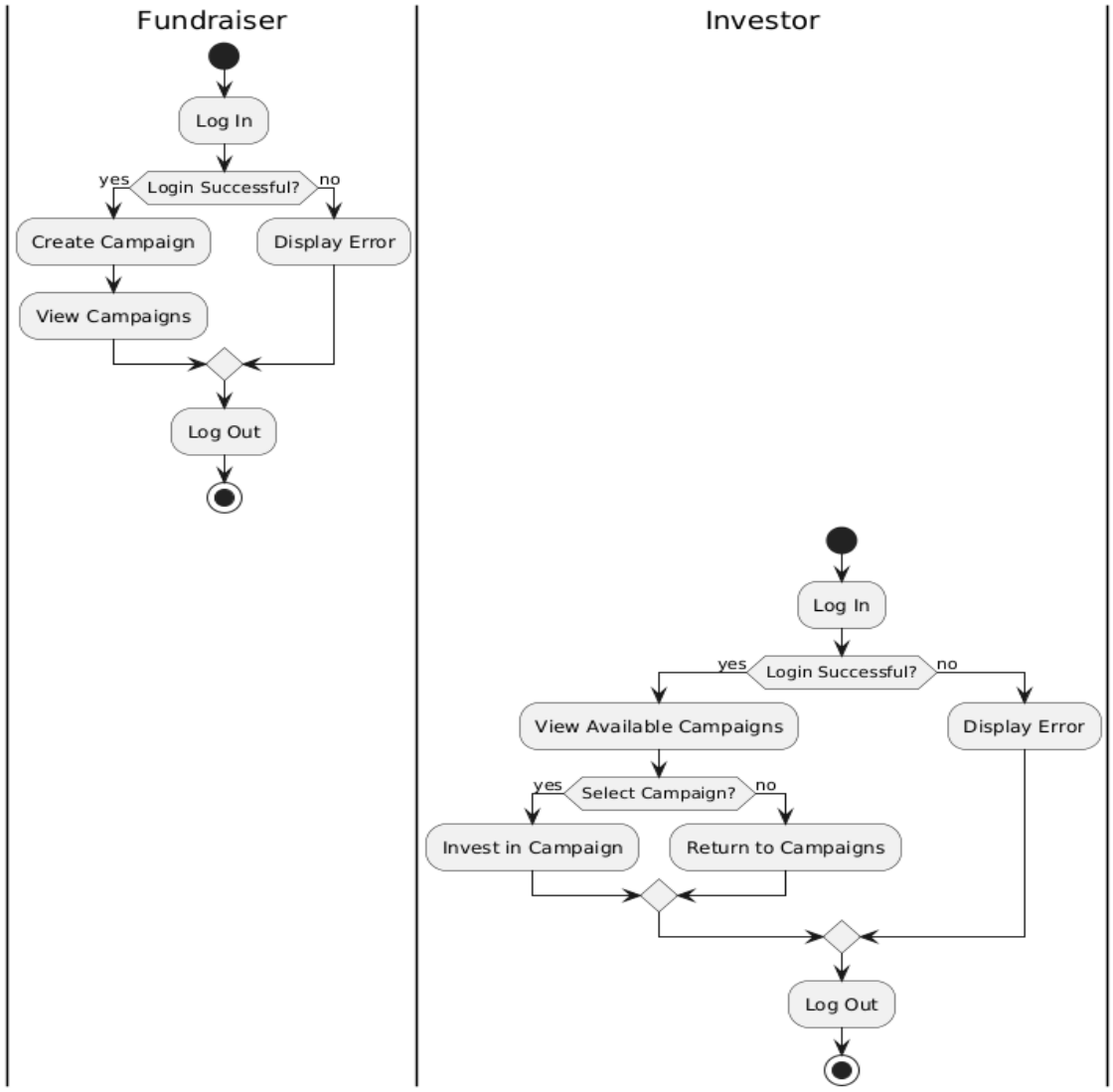


Fig: 5.4 Activity Diagram for fundraiser and investor

CHAPTER-6
IMPLEMENTATION

IMPLEMENTATION

6.1 User Authentication Module

Authentication and authorization are critical components of the Fundraising and Investment Platform, ensuring that only registered users can access their data and perform specific actions securely.

Authentication

Authentication is the process of verifying a user's identity, typically using credentials like email and password. Once authenticated, a user is issued a JWT (JSON Web Token), which allows the Server to recognize the user in future requests. How Authentication Works

- **Registration (/api/auth/register)**

    1. **User Sign-up:** When a user signs up, they send their name, email, and password to the backend via a POST request.
    2. **Password Hashing:** The backend uses a library like bcrypt to hash the password before saving it in the database for security purposes.
    3. **User Creation:** After successful validation and hashing, the user's data (name, email, and hashed password) is stored in MongoDB.
    4. **JWT Token Generation:** Once registered, the backend creates a JWT for the user and sends it as part of the response. This token can be used for subsequent authentication (login).

```
javascript
const token = jwt.sign({ userId: user._id }, process.env.JWT_SECRET, {
  expiresIn: '1h',
});
```

- **Login (/api/auth/login)**

    1. **User Login**: When a user logs in, they provide their email and password.
    2. **Password Verification**: The backend checks if the user exists and compares the

provided password with the hashed password stored in the database using bcrypt.compare().

3. **JWT Token Issuance**: If the login credentials are valid, the backend generates a JWT token and returns it to the user.

   Example response after login:

   json

   ```
   {
    "message": "Login successful",

   "token": "JWT_TOKEN"
   }
   ```

6.2 Authorization

Authorization ensures that a user can only access resources (e.g., campaigns, profiles) they are permitted to access. This is enforced using the JWT token that was issued during authentication.
How Authorization Works

- **JWT Token Storage**
  - After login, the JWT token is typically stored on the client-side (frontend) in localStorage or sessionStorage (or more securely, in a cookie).
  - The token is included in the Authorization header of every HTTP request that requires authorization (e.g., accessing user's campaigns).

    Example request with JWT:

    json

    ```
    {
     "Authorization": "Bearer JWT_TOKEN"
    }
    ```

- **Access Control**
  - **Fetching User's Campaigns**: When a user requests to fetch their campaigns (GET /api/campaigns), middleware verifies their token, extracts their user ID, and queries the database for campaigns associated with that user ID.

    javascript

```
// Get all campaigns for the authenticated user
router.get('/campaigns', authenticateToken, async (req, res) => {
  const campaigns = await Campaign.find({userId: req.user});
  res.json(campaigns);
});
```

- **Updating or Deleting a Campaign**: For routes like updating (PUT /api/campaigns/:id) or deleting (DELETE /api/campaigns/:id), middleware ensures that the campaign belongs to the authenticated user by checking the campaign's userId field against req.user (the authenticated user's ID). Example of protecting update logic:

```
javascript
router.put('/campaigns/:id', authenticateToken, async (req, res) => {
  const campaign = await Campaign.findById(req.params.id);

  if (!campaign || campaign.userId.toString() !== req.user) {
    return res.status(403).json({ message: 'Access denied' });
  }
// Proceed with updating the campaign
  campaign.title = req.body.title;
  await campaign.save();
  res.json(campaign);
});
```

6.3 Admin Module

1. Admin Dashboard

- **Key Metrics**: Visual displays of total funds raised, number of active campaigns, new users, and recent donations.
- **Alerts & Notifications**: Highlight urgent tasks or issues that require attention (e.g., low-performing campaigns).

2. User Management

- User Profiles: Access detailed profiles for each user, including their activity history and donation records.
- Role Management: Assign roles and permissions (admin, campaign manager, moderator) to control access levels.
- User Support: View and respond to user inquiries or issues reported through the platform.

3. Campaign Management

- Campaign Overview: List of all campaigns with essential details (title, goal, amount raised, status).
- Edit/Delete Campaigns: Modify campaign details or remove campaigns as necessary.
- Approval Process: Implement an approval system for new campaigns before they are published.

4. Donation Management

- Transaction History: Monitor all donations made, including donor information and associated campaigns.
- Refunds: Process refunds for donations when applicable and maintain a log of transactions.
- Donor Engagement: Track interactions with donors and analyze patterns in giving.

5. Reporting and Analytics

- Financial Reports: Generate detailed reports on total funds raised over specific periods and by individual campaigns.
- User Insights: Analyze user demographics, donation trends, and campaign performance metrics.
- Custom Reports: Allow admins to create tailored reports based on specific criteria (e.g., date range, campaign type).

6. Communication Tools

- Email Campaigns: Create and send newsletters or targeted emails to users and donors.
- Event Announcements: Inform users about upcoming fundraising events or milestones reached by campaigns.

7. Content Management

- Static Pages: Manage content for informational pages (FAQs, About Us, Terms of Service).
- Blog or News Section: Post updates, success stories, or fundraising tips to engage users.

8. Security and Compliance

- Audit Trails: Keep logs of admin actions for accountability and transparency.
- Data Protection: Ensure compliance with data protection regulations (e.g., GDPR, CCPA) and secure handling of user information.

6.4 Roles Module

1. Role Definitions

- Admin: Full access to all features, including user management, campaign oversight, and reporting.
- Campaign Manager: Access to create, edit, and manage fundraising campaigns, view donation metrics, and communicate with donors.
- Donor: Ability to view campaigns, make donations, and track their donation history.
- Volunteer: Access to support specific campaigns, view volunteer opportunities, and report on engagement activities.
- Viewer: Limited access to view campaigns and donation progress without the ability to make changes or donations.

2. Roles Management

- Create/Edit Roles: Admins can create new roles or modify existing ones to meet the platform's needs.
- Role Assignments: Admins can assign or change roles for users based on their responsibilities.
- Role Deletion: Ability to remove roles that are no longer needed or applicable.

3. Permissions

- Permission Levels: Define what each role can and cannot do within the platform (e.g., view reports, edit campaigns, manage users).
- Custom Permissions: Allow for customization of permissions at a granular level for specific roles.

4. Role-Based Access Control (RBAC)

- Access Control Lists: Implement lists that define permissions for each role, ensuring that users only see what they need to.
- Dynamic Access: Adjust access levels based on user activity or changes in role status.

5. User Interface

- Role Assignment Interface: A user-friendly interface for admins to easily assign roles to users.
- Role Overview Dashboard: Visual representation of all roles, users assigned to each role, and their activity levels.

6. Notifications and Alerts

- Role Changes: Notify users when their role is assigned or modified.
- Access Denied Alerts: Inform users when they attempt to access features outside their permissions.

7. Auditing and Reporting

- Audit Logs: Keep track of role changes, including who made changes and when.
- Reporting: Generate reports on user activity by role, helping to analyze engagement and effectiveness.

8. Security Measures

- Authentication and Authorization: Implement secure authentication methods to verify user identities before role changes are made.
- Data Protection: Ensure that role definitions and permissions comply with data protection regulations.

9. Integration with Other Modules

- Link to User Management: Ensure seamless integration with the user management system for easy role assignments.
- Campaign Management Integration: Allow campaign managers to manage their campaigns based on their assigned roles.

10. Documentation and Support

- User Guide: Provide documentation on how to assign roles and the implications of each role.
- Help Section: Include a support area for users to learn about role functionalities and report issues.

Implementation Considerations

- **Technology Stack**
  Choose a tech stack suitable for your needs (e.g., React, Angular for frontend; Node.js, Django for backend).
- **User Experience (UX)**
  Ensure the interface is user-friendly, with clear navigation and responsive design for both desktop and mobile.
- **Testing and Feedback**
  Conduct user testing with admins to gather feedback and refine the module before full deployment.

6.5 Sample Code

```
class QuickLRU {
   constructor(options = {}) {
      if (!(options.maxSize && options.maxSize > 0)) {
         throw new TypeError(`maxSize` must be a number greater than 0');
      }

      if (typeof options.maxAge === 'number' && options.maxAge === 0) {
         throw new TypeError(`maxAge` must be a number greater than 0');
      }

      this.maxSize = options.maxSize;
      this.maxAge = options.maxAge || Infinity;
      this.onEviction = options.onEviction;
      this.cache = new Map();
      this.oldCache = new Map();
      this._size = 0;
   }

   _emitEvictions(cache) {
      if (typeof this.onEviction !== 'function') {
         return;
      }

      for (const [key, item] of cache) {
         this.onEviction(key, item.value);
      }
   }

   _deleteIfExpired(key, item) {
      if (typeof item.expiry === 'number' && item.expiry <= Date.now()) {
         if (typeof this.onEviction === 'function') {
            this.onEviction(key, item.value);
         }

         return this.delete(key);
      }

      return false;
   }

   _getOrDeleteIfExpired(key, item) {
      const deleted = this._deleteIfExpired(key, item);
      if (deleted === false) {
         return item.value;
      }
   }
```

```
getItemValue(key, item) {
   return item.expiry ? this._getOrDeleteIfExpired(key, item) : item.value;
   }

   _peek(key, cache) {
      const item = cache.get(key);

      return this._getItemValue(key, item);
   }

   _set(key, value) {
      this.cache.set(key, value);
      this._size++;

      if (this._size >= this.maxSize) {
         this._size = 0;
         this._emitEvictions(this.oldCache);
         this.oldCache = this.cache;
         this.cache = new Map();
      }
   }

   _moveToRecent(key, item) {
      this.oldCache.delete(key);
      this._set(key, item);
   }

   * _entriesAscending() {
      for (const item of this.oldCache) {
         const [key, value] = item;
         if (!this.cache.has(key)) {
            const deleted = this._deleteIfExpired(key, value);
            if (deleted === false) {
               yield item;
            }
         }
      }

      for (const item of this.cache) {
         const [key, value] = item;
         const deleted = this._deleteIfExpired(key, value);
         if (deleted === false) {
            yield item;
         }
      }
   }

   get(key) {
```

```
if (this.cache.has(key)) {


 const item = this.cache.get(key);
 return this._getItemValue(key, item);
      }

      if (this.oldCache.has(key)) {
         const item = this.oldCache.get(key);
         if (this._deleteIfExpired(key, item) === false) {
            this._moveToRecent(key, item);
            return item.value;
         }
      }
   }

   set(key, value, {maxAge = this.maxAge === Infinity ? undefined : Date.now() + this.maxAge}
= {}) {
      if (this.cache.has(key)) {
         this.cache.set(key, {
            value,
            maxAge
         });
      } else {
         this._set(key, {value, expiry: maxAge});
      }
   }

   has(key) {
      if (this.cache.has(key)) {
         return !this._deleteIfExpired(key, this.cache.get(key));
      }
      if (this.oldCache.has(key)) {
         return !this._deleteIfExpired(key, this.oldCache.get(key));
      }
      return false;
}
   }
```

CHAPTER-7
TESTING

TESTING

7.1 Types of Testing

1. Unit Testing

- Purpose: Test individual components or functions in isolation.
- Focus: Verify that each unit of code behaves as expected (e.g., donation calculations, user authentication).

2. Integration Testing

- Purpose: Ensure that different modules or services work together correctly.
- Focus: Test interactions between components, such as the donation process and payment gateway integration.

3. Functional Testing

- Purpose: Validate that the application meets specified requirements.
- Focus: Test features like user registration, campaign creation, and donation processing.

4. End-to-End Testing

- Purpose: Simulate real user scenarios to validate the entire workflow.
- Focus: Test complete user journeys, such as a donor signing up, selecting a campaign, and completing a donation.

5. Performance Testing

- Purpose: Assess the application's responsiveness and stability under load.
- Focus: Measure how the platform performs during peak usage (e.g., high traffic during a fundraising event).

6. Security Testing

- Purpose: Identify vulnerabilities and ensure data protection.
- Focus: Test for common security issues like SQL injection, cross-site scripting (XSS), and data encryption.

7.2 Key Components of Testing

1. User Registration and Authentication

- Functionality: Test the sign-up, login, and password recovery processes.
- Security: Verify secure password storage, account lockout after failed attempts, and two-factor authentication.

2. Campaign Creation and Management

- Creation Process: Ensure users can create, edit, and delete campaigns without issues.
- Validation: Test for proper input validation (e.g., required fields, character limits).
- Approval Workflow: If applicable, verify that the admin approval process works as intended.

3. Donation Processing

- Payment Integration: Test integration with payment gateways (e.g., Stripe, PayPal) for smooth transactions.
- Transaction Flow: Ensure donations can be made successfully, with confirmations sent to users.
- Refunds and Cancellations: Verify that users can request refunds and that the process is handled correctly.

4. User Roles and Permissions

- Role Functionality: Test different user roles (admin, campaign manager, donor) to ensure each has appropriate access.
- Access Control: Verify that unauthorized users cannot access restricted features.

5. Reporting and Analytics

- Data Accuracy: Ensure that fundraising totals, user engagement metrics, and campaign statistics are accurately calculated and displayed.
- Custom Reports: Test the ability to generate custom reports based on various parameters.

7.3 Black Box Testing

Black box testing focuses on verifying the functionality of the application without considering its internal code structure. It validates whether the expected output matches the provided input.

Test Cases

- **Test Case 1: User Registration**
  - **Input**:

    json

    ```
    {
      "email": "testuser@email.com",
      "password": "testpassword"
    }
    ```

  - **Expected Output**:

    json

    ```
    {
      "message": "User registered successfully"
    }
    ```

- **Test Case 2: Create Campaign**
  - **Input**:

    json

    ```
    {
      "title": "New Campaign",
      "description": "This is a test campaign",
      "goalAmount": 5000
    }
    ```

  - **Expected Output**

    json

    ```
    {
      "message": "Campaign created successfully"
    }
    ```

7.4 White Box Testing

White box testing examines the internal logic, structure, and code paths of the application. It ensures that all parts of the code are functioning as expected.

Test Cases

- **Test Case 1: Update User Profile**
  - **Input**:

    ```
    json
    {
      "email": "updateduser@email.com",
      "password": "newpassword"
    }
    ```

  - **Expected Output**

    ```
    json
    {
      "message": "User profile updated successfully"
    }
    ```

- **Test Case 2: Edit Campaign Details**
  - **Input**:

    ```
    json
    {
      "title": "Updated Campaign Title",
      "description": "Updated description for the campaign"
    }
    ```

  - **Expected Output**

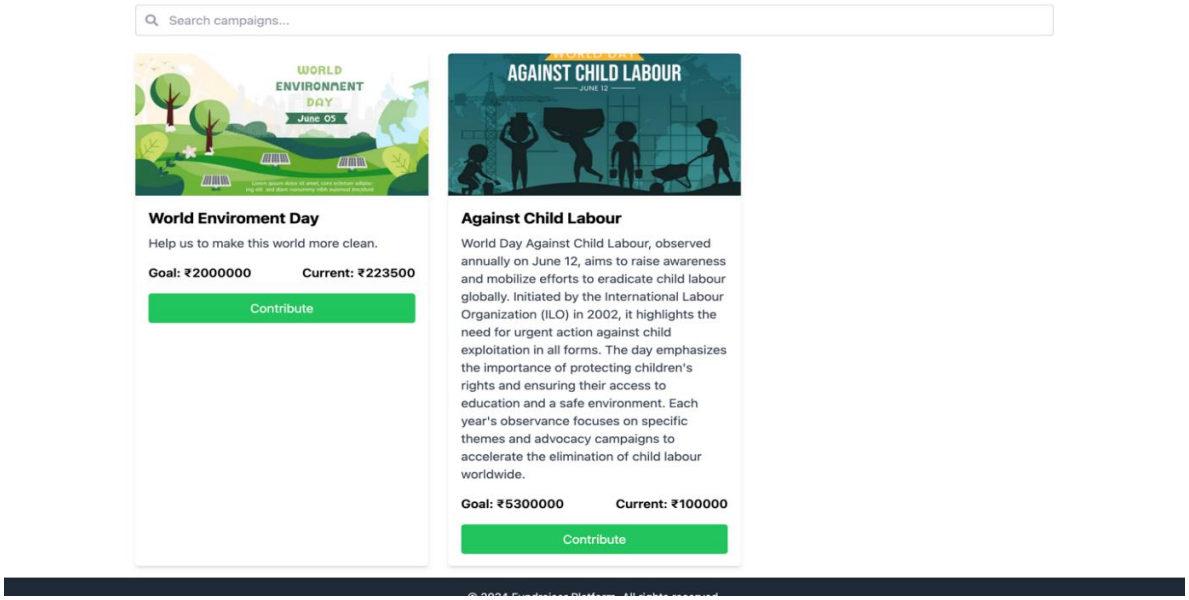
    ```
    json
    {
      "message": "Campaign details updated successfully"
    }
    ```

CHAPTER-8
OUTPUT SCREENS

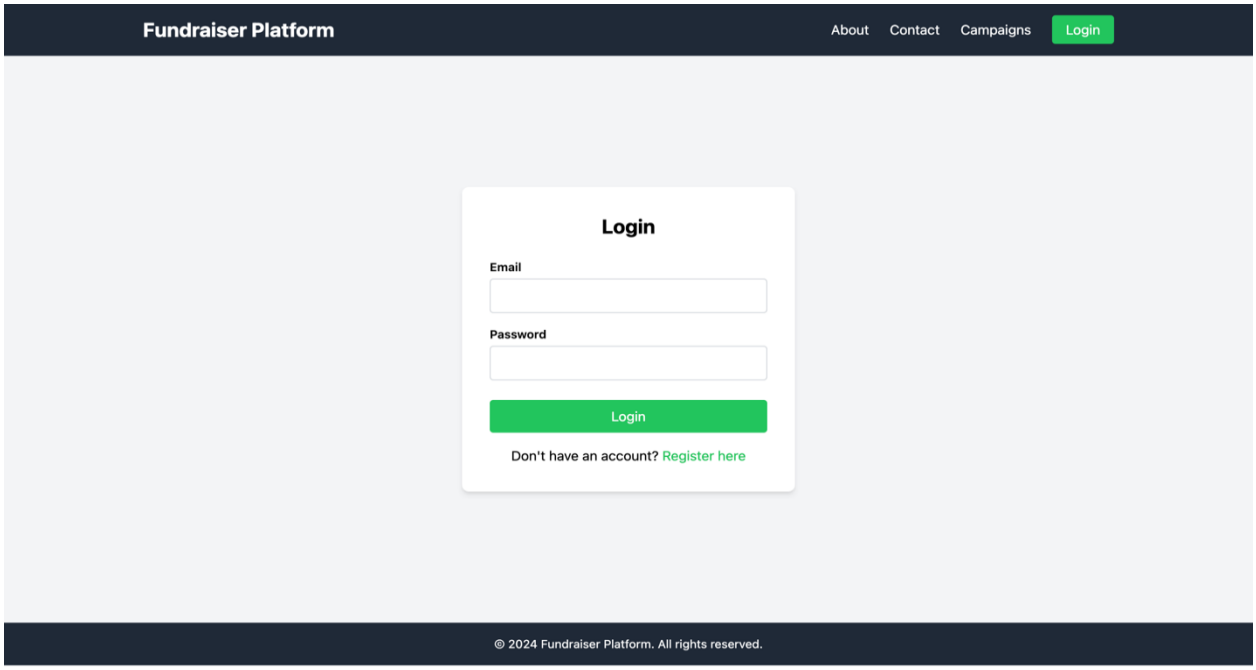
OUTPUT SCREENS

8.1 Home Page

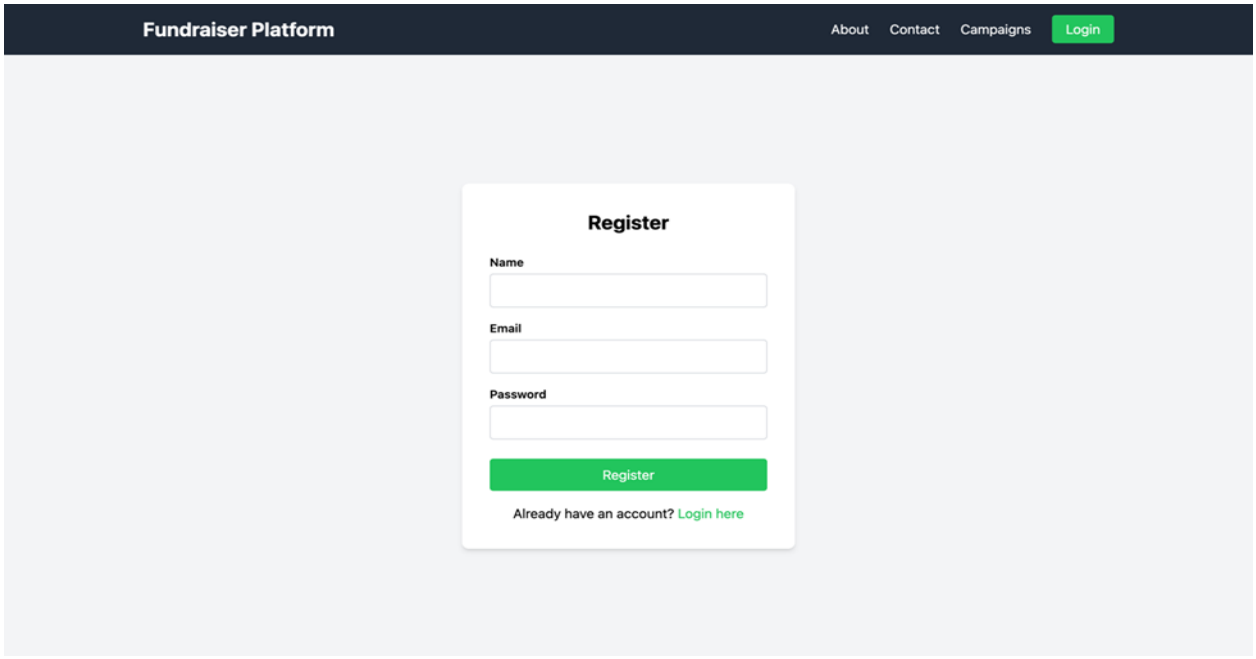
8.2 Campaigns Page



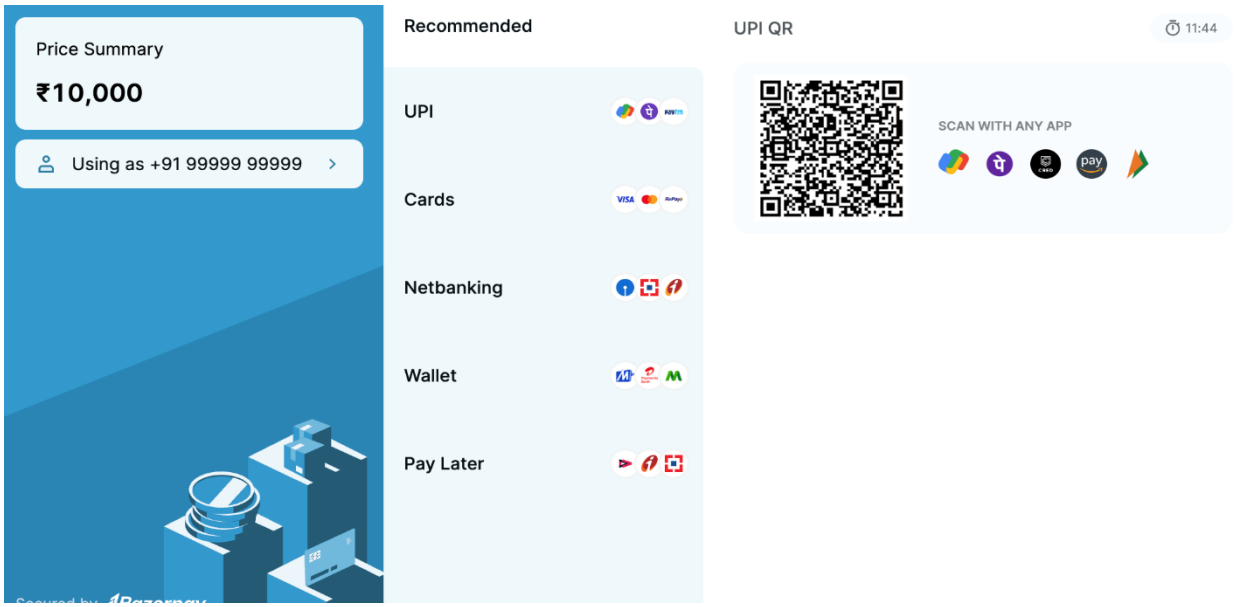
8.3 Login page



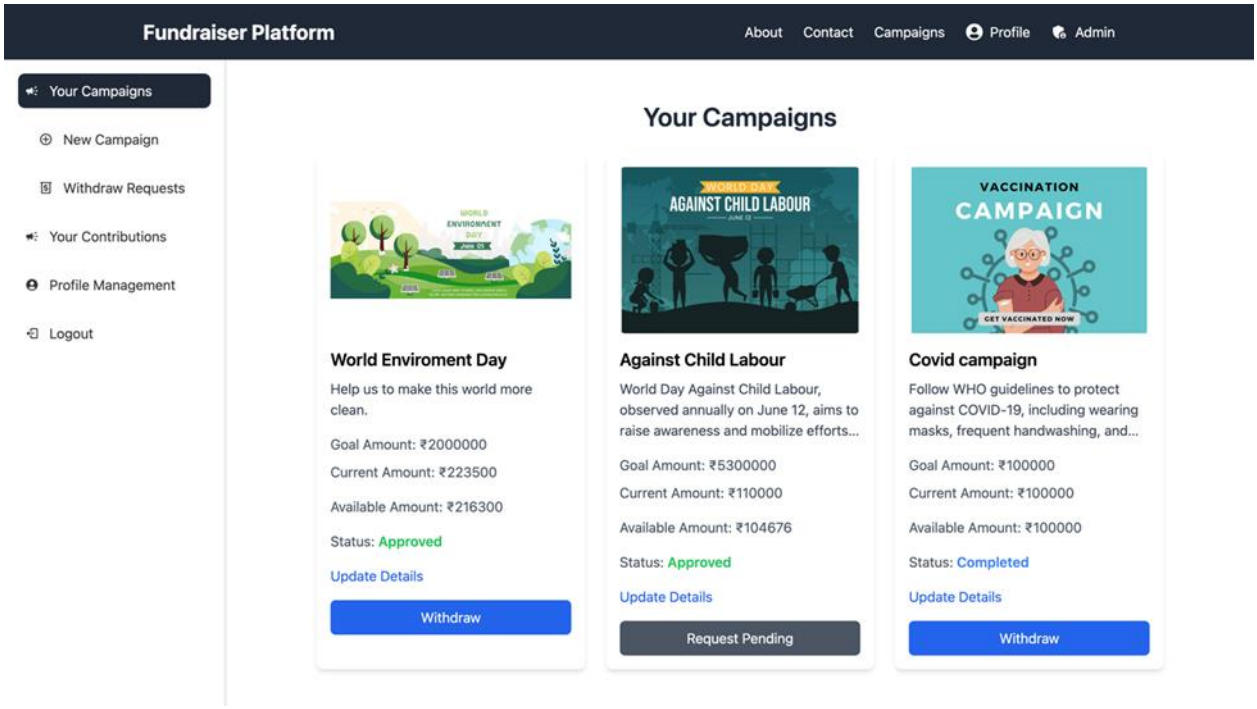
8.4 Register Page



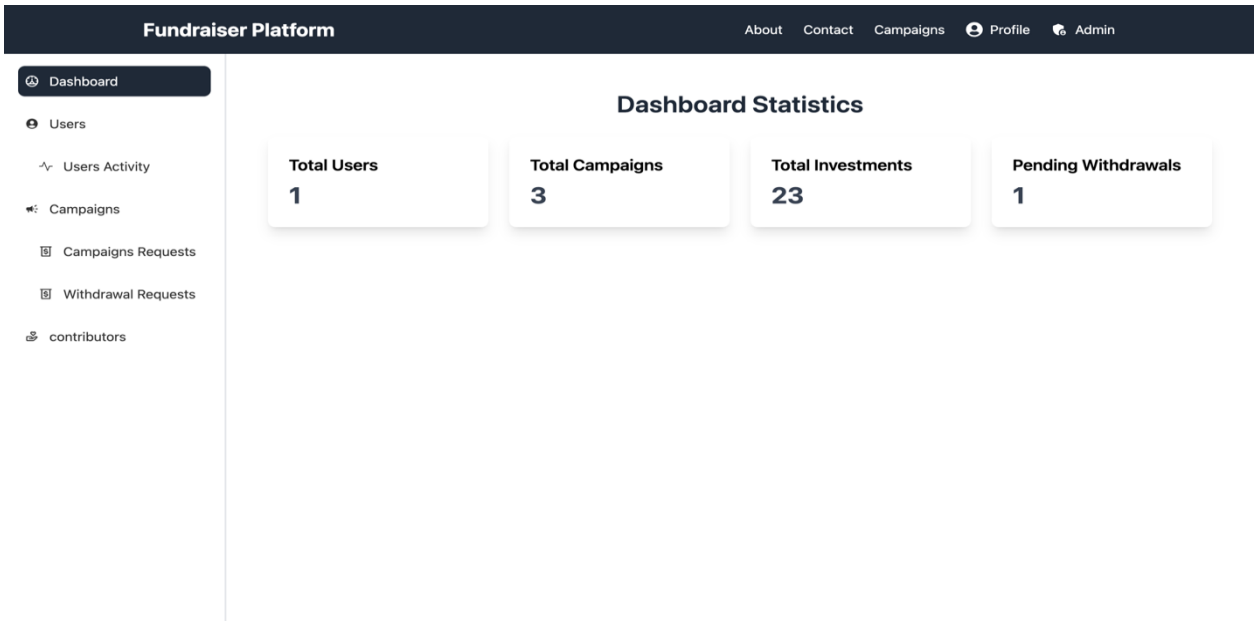
8.5 Razor pay



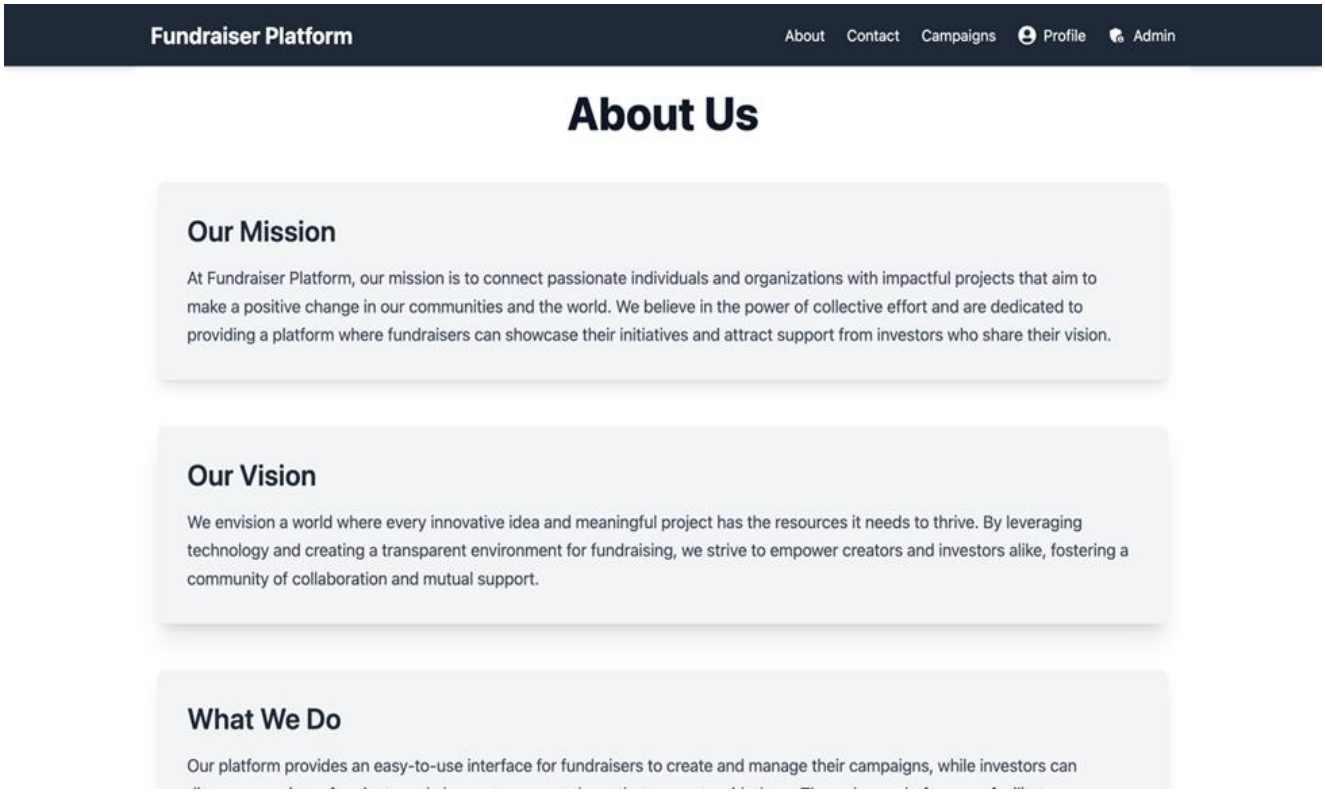
8.6 Your Campaigns Page



8.7 Admin Panel



8.8 About Us



CHAPTER-9
FUTURE ENHANCEMENTS

FUTURE ENHANCEMENTS

The Fundraising and Investment Platform is designed to be scalable and adaptable, allowing for the integration of new features and improvements over time. Here are some potential future enhancements that could further enhance the platform's functionality and user experience:

1. Crowdfunding Features

- Micro-Investments: Introduce the ability for multiple investors to contribute smaller amounts to a campaign, enabling fundraisers to reach their goals through collective micro-investments.
- Reward-Based Funding: Implement a system where fundraisers can offer non-monetary rewards to investors based on their contributions, similar to popular crowdfunding platforms.

2. Advanced Analytics and Reporting

- Investor Insights: Provide detailed analytics for investors to track their investment performance, returns, and overall portfolio health.
- Campaign Performance Metrics: Offer fundraisers comprehensive analytics on campaign Performance, including engagement rates and funding trends.

3. Enhanced User Engagement

- Community Features: Develop forums or discussion boards where fundraisers and investors can interact, share insights, and build a community.
- Social Sharing Tools: Enable users to easily share campaigns on social media platforms to increase visibility and attract more investors.

4. Improved Security Measures

- Two-Factor Authentication (2FA): Implement 2FA for added security during login processes, protecting user accounts from unauthorized access.
- Fraud Detection Algorithms: Develop algorithms to detect and prevent fraudulent activities on the platform, ensuring a safe environment for all users.

CHAPTER-10
CONCLUSION

CONCLUSION

- In conclusion, our fundraising campaign has not only highlighted the importance of our mission but has also fostered a sense of community and shared purpose among our supporters. As we reflect on the journey we've undertaken, it's clear that our efforts have transcended mere financial goals; they have cultivated a vibrant network of advocates committed to effecting meaningful change.
- The funds raised will directly impact our programs, allowing us to expand our reach and enhance the services we provide to those in need. Every contribution, whether large or small, has played a vital role in driving our mission forward, and we are deeply grateful for the generosity displayed by our supporters.

- Moreover, this campaign has demonstrated the power of collective action. Each event, social media share, and personal outreach contributed to a growing awareness of our cause, engaging new supporters who resonate with our vision. The stories shared by beneficiaries have inspired many to take action, reinforcing the belief that together, we can make a significant difference. The enthusiasm and commitment from our volunteers, donors, and community members have been invaluable; they remind us that our work is not just about raising funds, but about building relationships and fostering a culture of giving.

- Thank you for believing in our mission and for your unwavering support. Together, we are not just making a difference; we are building a legacy of hope and resilience that will resonate for years to come. Your involvement is crucial, and we look forward to continuing this journey with you as we strive to create a brighter future for those we serve.

CHAPTER-11
REFERENCES

REFERENCES

1. Statistical Data

- Utilizing statistics can highlight the need for support. For example, referencing studies or reports from reputable organizations can illustrate the scope of a problem, such as poverty rates or health disparities.
- Example: “According to the World Health Organization, over 1 billion people lack access to clean water, underscoring the urgent need for our clean water initiatives.”

2. Success Stories

- Sharing case studies or testimonials from individuals who have benefited from your programs can be compelling. These stories personalize your mission and show tangible results.
- Example: “Meet Sarah, a participant in our job training program, who secured a full-time job within three months of completing the course.”

3. Research Findings

- Citing peer-reviewed research can bolster your organization’s credibility, especially if it supports the effectiveness of your programs.
- Example: “A study published in the Journal of Public Health found that our approach to community health outreach reduced emergency room visits by 30%.”

4. Endorsements and Partnerships

- Highlighting endorsements from influential figures or partnerships with respected organizations can enhance your credibility.
- Example: “Supported by the United Nations, our initiative has reached over 10,000 families this year.”

5. Financial Accountability Reports

- Providing links to independent financial audits or transparency reports demonstrates your organization’s commitment to responsible stewardship of donor funds.
- Example: “Our annual financial report, prepared by an independent auditor, is available on our website for full transparency.”