

PANCHAYAT SPECIFIC CHARITY FUND COLLECTION THROUGH ENTERTAINMENT

Main Project Report

Submitted by

SAFA BEEGUM P A

FIT23MCA - 2097

In partial fulfillment of the requirements for award of the degree of

MASTER OF COMPUTER APPLICATIONS

Of

A P J Abdul Kalam Technological University



**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY(FISAT)®
ANGAMALY-683577, ERNAKULAM
APRIL 2025**

PANCHAYAT SPECIFIC CHARITY FUND COLLECTION THROUGH ENTERTAINMENT

Main Project Report

Submitted by

SAFA BEEGUM P A

Reg No: FIT23MCA-2097

*Submitted in partial fulfillment of the requirements for the award of the degree
of
Master of Computer Applications of
A P J Abdul Kalam Technological University*



**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®
ANGAMALY - 683577, ERNAKULAM (DIST.)
APRIL 2025**

DECLARATION

I Safa Beegum P A, hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (FISAT), Angamaly in partial fulfilment of the award of the degree of Master of Computer Applications is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date:

Place: Angamaly



Safa Beegum P A



FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)[®]

(An ISO 9001:2015 Certified, NAAC ('A+') Accredited Institution with NBA Accredited Programmes)

Approved by AICTE - Affiliated to APJ Abdul Kalam Technological University, Kerala)

Owned and Managed by Federal Bank Officers' Association Educational Society

HORMIS NAGAR, MOOKKANNOOR P.O., ANGAMALY - 683 577, ERNAKULAM, KERALA, INDIA.

Tel : (0) 0484 - 2725272

E-Mail : mail@fisat.ac.in

Website : www.fisat.ac.in

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr./Ms. Safa Beegum P A (FIT23MCA-2097) has successfully completed his/her Main Project with the title "Panchayat Specific Charity Fund Collection Through Entertainment" under the guidance of Ms. Senu Abi in the Department of Computer Applications, FISAT, during the period from 16th December 2024 to 4th April 2025.

Dr. Deepa Mary Mathews
Head of the Department



**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY
(FISAT)®
ANGAMALY, ERNAKULAM-683577**

DEPARTMENT OF COMPUTER APPLICATIONS

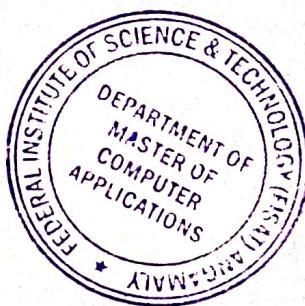


Focus on Excellence

CERTIFICATE

*This is to certify that the project report titled '**PANCHAYAT SPECIFIC CHARITY FUND COLLECTION THROUGH ENTERTAINMENT**' submitted by *Safa Beegum P A* (Reg No. FIT23MCA-2097) towards partial fulfilment of the requirements for the award of the degree of Master of Computer Applications is a record of bonafide work carried out by her during the year 2025.*


Ms. Senu Abi
Project Guide




Deepa Mary Mathews
Head of the Department

Submitted for the viva-voice held on at

Examiner:

ACKNOWLEDGEMENT

Gratitude is a feeling which is more eloquent than words, more silent than silence. To complete this project work I needed the directions, assistance and co-operation of various individuals, which is received in abundance with the grace of God.

I hereby express my deep sense of gratitude to **Dr. Jacob Thomas V**, Principal of FISAT and **Dr. Mini P R**, Vice Principal, FISAT, Angamaly for allowing me to utilize all the facilities of the college. I am equally grateful to **Dr. Deepa Mary Mathews**, HOD, Department of Computer Applications, FISAT, who had been a source of inspiration. During the period of my project work, I have received generous help from **Ms. Senu Abi**, my project guide, which I like to put on record here with deep gratitude and great pleasure. I would like to take this opportunity to sincerely thank **Ms. Anju L**, my Scrum Master, and express my deep gratitude with great pleasure.

Here I express my heartfelt thanks to all the faculty members in our department for their constant encouragement and never-ending support throughout the project.

Finally, I wish to express a whole hearted thanks to my parents, friends and well-wishers who extended their help in one way or other in preparation of my project. Besides all, I thank God for everything.

ABSTRACT

The Panchayat Specific Charity Fund Collection Through Entertainment platform is a web-based digital fundraising system designed to support local communities by integrating entertainment with social impact. Traditional donation methods often struggle with lack of transparency, donor disengagement, and inefficient fund allocation. This platform addresses these challenges by introducing an interactive gaming-based donation model, where users can play games, earn rewards, and contribute to verified charities in a structured and transparent manner. This approach not only encourages more participation but also fosters a sense of community-driven impact.

The platform is developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), ensuring high scalability, security, and efficient data management. It operates on a revenue-sharing model, where 70% of the collected funds directly support verified charities, 20% serves as incentives for top players, and 10% is allocated as a platform fee for operational costs. The integration of a wallet system, leaderboard rankings, and automated reward distribution encourages consistent engagement from users while ensuring financial transparency and accountability.

To enhance community engagement and streamline emergency responses, the platform includes a Community Announcement and Emergency Alert System. This feature enables email-based alerts for urgent situations such as floods, fires or accidents, ensuring that local residents and authorities receive timely notifications. The role-based access system ensures that admins, social workers, and users can effectively manage fund distribution, monitor financial transactions, oversee gaming activity, and maintain operational accountability.

Beyond fundraising, the platform serves as a hub for social responsibility, digital inclusion, and community-driven action. By leveraging technology for social good, it creates a sustainable, engaging, and transparent ecosystem that not only mobilizes community support but also ensures financial accountability, promotes participation, and strengthens local charity fundraising efforts in a structured, enjoyable, and impactful manner.

CONTENTS

1. INTRODUCTION

1.1 Introduction.....	1
1.2 Scope of the work.....	2

2. PROOF OF CONCEPT

2.1 Literature Review.....	4
2.2 Existing System.....	9
2.2.1 Limitation of Existing Systems.....	10
2.3 Proposed System.....	11
2.3.1 Objectives.....	12

3. SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis.....	14
3.1.1 Introduction.....	14
3.1.2 Methodology.....	15
3.1.3 Hardware and Software Requirements.....	16
3.2 System Design.....	17
3.2.1 Introduction.....	17
3.2.2 Module Description.....	17
3.2.3 System Architecture.....	20
3.2.4 Database Description.....	22
3.3 Issues faced and Remedies Taken.....	29
3.3.1 Issues.....	29
3.3.2 Remedies.....	30

4. RESULT

4.1 Result.....	31
-----------------	----

5. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion.....	35
5.2 Future Enhancements.....	36

6. APPENDIX

6.1 Source Code.....	37
6.2 Screenshots.....	52

7. REFERENCES

60

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Panchayat Specific Charity Fund Collection Through Entertainment project is an innovative initiative designed to integrate gaming with social responsibility, providing an engaging and transparent method for raising funds for charitable causes. This platform allows users to participate in various interactive games, where a portion of the collected funds is allocated to verified charities. By making donations a fun and rewarding experience, the project encourages more people to contribute to social welfare while ensuring accountability and transparency in fund distribution.

Fundraising for charity often faces challenges such as low participation, lack of transparency, and difficulty in sustaining contributions. Traditional donation methods do not always engage people actively, making it hard to maintain a steady flow of funds for charitable initiatives. This project addresses these issues by introducing an entertainment-driven approach to donations. By linking gaming with philanthropy, it ensures that users are both entertained and motivated to contribute. In addition to fundraising, the platform serves as a community announcement hub, providing residents with important updates on local events, government schemes, and public initiatives. Another key feature of the platform is its emergency alert system, which notifies users about critical situations such as floods, accidents, or other urgent matters, helping them take necessary precautions.

The platform operates on a structured revenue-sharing model, ensuring financial transparency. Seventy percent of the funds go directly to verified charities. Twenty percent is allocated as rewards for top players. The remaining ten percent is retained as a platform fee to cover operational and maintenance costs. Every transaction is recorded and monitored, allowing users to track their contributions and see the real-world impact of their participation. To further promote engagement, the platform incorporates a leaderboard system, where users can track their rankings based on their gameplay. By combining entertainment, charity, and community awareness, this project introduces a sustainable, transparent, and impactful way of engaging people in philanthropy while also serving as a valuable communication and alert system for the local community.

1.2 SCOPE OF THE WORK

The primary objective of the Panchayat Specific Charity Fund Collection Through Entertainment platform is to integrate gaming with social responsibility, creating a seamless and engaging way for users to contribute to local charitable causes while enjoying entertainment. This web-based platform is designed to ensure transparency, accountability, and social impact by effectively managing game-based contributions and reward distributions at the panchayat level.

Features and Functionalities

This system offers multiple advanced features to address the shortcomings of traditional charity models by leveraging entertainment for fundraising. The platform is structured around three user roles: Normal Users, Hosts/Admins, and Social Workers. Normal users participate in games through a pay-to-play model, earn rewards based on their leaderboard performance, and track fund allocation transparently. They can also report fraudulent activities to maintain a secure ecosystem. Hosts/Admins manage platform operations, oversee financial transparency, ensure compliance with regulations, and moderate reports to uphold platform integrity. Social Workers post verified charitable needs specific to panchayat communities, facilitate contributions, and collaborate with administrators to ensure effective fund allocation. This system fosters a transparent and engaging ecosystem where entertainment seamlessly integrates with philanthropy. By combining gaming with social impact, it encourages active user participation while ensuring that donations reach verified causes efficiently.

The platform's key functionalities include a Game Management Module to ensure fairness and engagement, a Wallet System for managing winnings that can be donated or withdrawn, and a Payment & Donation Module that guarantees secure transactions. The Leaderboard & Reward System motivates users by offering monetary incentives, while the Fund Allocation & Transparency feature ensures proper financial tracking and visibility. The Charity Management Module connects donors with social workers for real-world impact, and the Reporting & Inactivity Management feature helps monitor fraudulent activities and inactive users. Additionally, the Notifications & Communication System keeps users updated on transactions, leaderboard standings, and charity updates, while the Social Sharing & Engagement Module enhances platform visibility. The platform also includes a User Feedback & Review System for continuous improvement and a Content Management Module to ensure an intuitive user experience.

Expected Outcomes

This project aims to revolutionize charitable fundraising by integrating entertainment with social impact. By leveraging a pay-to-play gaming model, the platform will encourage greater user participation while ensuring transparent and efficient fund allocation. The inclusion of a leaderboard and rewards system will drive engagement, motivating users to contribute actively while earning incentives.

One of the key expected outcomes is enhanced financial transparency, as all transactions, donations, and fund allocations will be tracked and visible to users in real-time. The collaboration between social workers and administrators will ensure that funds reach verified charitable needs in panchayat communities, making a tangible difference in people's lives. Additionally, the automated wallet system will streamline transactions, allowing users to donate their winnings or withdraw them seamlessly.

Furthermore, the project fosters a sense of community and social responsibility, encouraging users to support causes while enjoying interactive gameplay. The reporting and fraud detection mechanisms will maintain platform integrity by preventing misuse and ensuring fair play. As the platform evolves, future enhancements like mobile app integration, AI-driven fraud detection, and multilingual support will further expand accessibility and impact. This initiative sets a new standard for digital fundraising, making charitable giving more engaging, accountable, and effective while bridging the gap between entertainment and social good.

CHAPTER 2

PROOF OF CONCEPT

2.1 LITERATURES REVIEW

Sommerville, in his book “Software Engineering (10th Edition)” [1], provides a comprehensive overview of software engineering principles, methodologies, and best practices. The book emphasizes the systematic development of software, covering key topics such as software process models, requirements engineering, system design, software testing, and project management.

The author discusses various software development paradigms, including waterfall, agile, and iterative models, highlighting their advantages and limitations. Additionally, the book explores software quality assurance, risk management, and the role of emerging technologies in modern software development. Special attention is given to software maintenance and evolution, recognizing that software systems must be continuously updated to meet changing user needs.

Overall, Sommerville’s work serves as a foundational reference for software engineering, offering insights into both theoretical concepts and practical applications. It remains widely used in academic and professional settings due to its structured approach to building reliable, maintainable, and efficient software systems.

Pressman, in his book “Software Engineering: A Practitioner’s Approach (9th Edition)” [2], provides an in-depth exploration of software engineering principles, methodologies, and best practices. The book presents a practical approach to software development, covering essential topics such as software process models, requirements engineering, design principles, coding techniques, testing strategies, and project management.

The author emphasizes modern software engineering practices, including agile development, model-driven engineering, and software architecture design. The book also discusses risk management, software quality assurance, and process improvement techniques to ensure the delivery of reliable and maintainable software. Additionally, Pressman explores emerging trends in software engineering, such as cloud computing, mobile applications, and artificial intelligence-driven software systems.

Overall, this book serves as a fundamental resource for both students and professionals, offering a structured approach to developing high-quality software systems. Its blend of theoretical concepts and real-world applications makes it a widely used reference in the field of software engineering.

Rajan and Thomas, in their paper “Gamification in Crowdfunding Platforms: Engaging Users for Social Good” [3], explore how gamification strategies can enhance user engagement in digital crowdfunding platforms aimed at social impact. The authors investigate key game design elements such as points, leaderboards, badges, and reward mechanisms, demonstrating how these features drive user motivation and participation.

The study highlights that gamification not only improves platform interactivity but also builds a sense of community and accountability among users. By incorporating psychological motivators like competition, achievement, and social recognition, the paper shows that users are more likely to contribute repeatedly and promote the platform within their networks. Furthermore, the authors analyse successful case studies and present metrics for measuring user engagement and platform effectiveness.

Overall, the paper contributes valuable insights into the intersection of game theory and crowdfunding, positioning gamification as a powerful tool for enhancing participation in socially driven digital initiatives.

Kaur and Bansal, in their paper “Game-Based Fundraising: Leveraging Gamification for Non-Profit Engagement” [4], examine how gamified systems can be strategically implemented to enhance donor engagement and fundraising outcomes for non-profit organizations. The authors analyze the integration of game mechanics—such as progress bars, rewards, storytelling, and competitive features—into digital fundraising platforms to stimulate user interaction and emotional involvement.

The study emphasizes that gamification transforms passive donors into active participants by making the donation experience more engaging and enjoyable. It also explores how real-time feedback and goal achievement triggers foster a deeper sense of involvement and loyalty among supporters. The paper further discusses the psychological and behavioural factors influencing donation patterns when game-based elements are incorporated.

Overall, the research underlines the effectiveness of gamified fundraising as a modern

approach to increasing donations, strengthening donor relationships, and sustaining long-term participation in social impact campaigns.

Dey and Ramesh, in their study “Gamified Platforms for Community Development: A Case Study Approach” [5], investigate how gamification can be effectively utilized to empower communities and foster social development. Through detailed case studies, the authors analyse several gamified platforms that aim to drive civic engagement, improve participation in local initiatives, and promote collective problem-solving.

The research identifies core game elements—such as leaderboards, rewards, progress tracking, and community challenges—as effective tools in motivating individuals to contribute toward community goals. The authors highlight that gamification not only enhances user interaction but also builds a sense of ownership and accountability within communities.

Moreover, the study emphasizes the importance of user-centered design in developing these platforms to ensure inclusivity and sustainability. The findings reveal that gamified systems can significantly contribute to long-term community engagement and social innovation when designed with cultural and contextual sensitivity.

In their paper “Digital Games for Social Good: Merging Play and Purpose” [6], Singh and Verma explore the potential of digital games to drive social change by blending entertainment with purposeful action. The authors discuss how game mechanics—such as challenges, point systems, and storytelling—can be effectively used to raise awareness, promote education, and encourage community involvement on social issues.

The study presents several case examples where digital games were successfully deployed to influence behaviour in areas such as environmental conservation, public health, and civic responsibility. It also emphasizes that engagement levels are significantly higher when users perceive their in-game actions as contributing to real-world impact.

Furthermore, the paper underscores the importance of aligning game objectives with meaningful causes, ensuring that gameplay remains both enjoyable and ethically grounded. It concludes that digital games offer a powerful medium for advocacy and education when designed with thoughtful integration of social goals.

Kumar and Patel [7] present a detailed exploration of the evolving landscape of secure payment gateways in web applications. The paper highlights the critical importance of safeguarding user financial data in an era of growing e-commerce. The authors classify payment gateway solutions into client-side encryption, hosted gateways, and API-based integrations, discussing their advantages and limitations. The analysis covers industry-standard protocols such as SSL/TLS, tokenization, and PCI DSS compliance, reinforcing the role of encryption and secure transmission in preventing fraud.

Furthermore, the paper delves into the implementation of advanced security techniques like biometric authentication and blockchain-based gateways. The study emphasizes the significance of user trust, user experience, and seamless integration for the successful deployment of secure payment systems. Their findings suggest that while technology is advancing, the human factor—such as user awareness and interface design—also plays a crucial role in the effectiveness of secure payment systems in real-world scenarios.

Singh and Verma [8] examine the rising trend of digital wallets in incentivizing user engagement through integrated reward systems. The paper explains the architecture of wallet-based systems and their role in gamified digital platforms. The authors analyze how rewards—either monetary or token-based—can be accumulated, stored, and redeemed through in-platform wallets. This system not only motivates user retention but also provides a seamless mechanism for transactions and donations, particularly in social impact-driven applications.

The study also discusses key concerns such as security vulnerabilities, transaction transparency, and the need for robust wallet management systems. The paper highlights the importance of user-friendly wallet interfaces and security protocols, such as two-factor authentication and real-time monitoring, to ensure trust. Singh and Verma argue that wallet systems, when integrated with gamification, can play a powerful role in driving both user motivation and platform sustainability.

The MongoDB Official Documentation [9] serves as an extensive guide for developers and system architects working with document-based NoSQL databases. It introduces MongoDB's flexible schema design, which allows for faster development cycles, especially in agile environments. The documentation explains CRUD operations, indexing, aggregation pipelines, and schema validation—providing foundational knowledge essential for efficient

data modelling.

Advanced topics such as replication, sharding, and performance optimization are also covered, ensuring developers can scale their applications while maintaining high availability and fault tolerance. MongoDB's support for horizontal scalability and its ability to handle large volumes of unstructured data make it particularly suitable for dynamic applications like gamified platforms and social services. The documentation also integrates examples and tutorials, making it a practical reference for real-time development and deployment.

The React.js Official Documentation [10] offers an in-depth understanding of one of the most widely used JavaScript libraries for building user interfaces. React's component-based architecture and virtual DOM are explained thoroughly, enabling developers to design fast, interactive web applications. The documentation walks through concepts like JSX, props, state, hooks, and lifecycle methods, which are core to managing dynamic data and user interactions in single-page applications.

Beyond basics, the documentation also includes advanced topics like routing, context API, performance optimization, and testing practices. React's declarative UI logic and modular structure make it ideal for developing scalable applications, including dashboards, games, and financial platforms. The hands-on examples and tutorials included in the guide allow developers to implement best practices and adapt React's ecosystem for both beginner and enterprise-level projects.

The Node.js Official Documentation [11] provides developers with a robust understanding of the event-driven, non-blocking I/O model that powers Node.js. Designed for building scalable server-side applications, Node.js supports rapid development of REST APIs and real-time web applications. The documentation explains core modules like HTTP, FS (File System), and Events, alongside package management via npm. It also covers asynchronous programming paradigms like callbacks, promises, and async/await, which are crucial for high-performance applications.

In addition to core modules, the documentation offers guidance on security practices, process management, debugging, and deploying applications. Node.js's ability to handle concurrent connections efficiently makes it an excellent choice for backend development of gamified platforms, chat applications, and real-time payment processing. The comprehensive examples and troubleshooting tips make this documentation essential for both novice and advanced

developers aiming to build robust web infrastructure.

The Express.js Guide [12] introduces developers to a minimal and flexible Node.js framework designed for building web applications and APIs. It explains how Express simplifies request handling, routing, middleware integration, and response rendering. The documentation emphasizes how developers can build RESTful APIs quickly while maintaining code modularity and clarity. Features such as route grouping, error handling, and templating support streamline the development of scalable server-side applications.

Additionally, the guide touches on authentication mechanisms, database connectivity, and CORS handling, which are vital for secure web development. Its compatibility with MongoDB and React makes Express a powerful middleware for full-stack JavaScript applications. The practical examples and sample projects provided in the documentation help developers design efficient backends, particularly for applications involving user management, payments, and real-time interactions.

2.2 EXISTING SYSTEM

The current charity fundraising landscape primarily relies on manual fundraising events, where donations are collected through in-person campaigns, charity auctions, or social gatherings. These events are often location-specific, requiring extensive planning, resources, and coordination. As a result, participation is restricted to individuals who can physically attend, limiting donor outreach and engagement. Additionally, fundraising success is often dependent on the scale and visibility of these events, making it difficult for smaller charitable causes to receive adequate support. Another major challenge in traditional charity models is the lack of continuous user engagement. Most donors contribute during specific events, and there is minimal interaction outside these occasions, leading to a decline in long-term donor retention. Unlike modern digital platforms that encourage ongoing participation, traditional fundraising methods fail to sustain user interest, resulting in inconsistent donation flows. Furthermore, due to the absence of gamification or incentives, there is no mechanism to motivate donors to contribute regularly or track their impact in an interactive way. The process also lacks automation, making it difficult to streamline donation collection, fund distribution, and donor engagement efficiently.

Transparency issues also pose significant concerns in traditional fundraising. Donors often lack direct insight into how their contributions are utilized, leading to scepticism and reduced trust in charitable organizations. Without a structured and verifiable system to track fund

allocation, there is a risk of mismanagement, delays, or inefficiencies in distributing donations to intended beneficiaries. The reliance on manual records also makes it challenging to provide real-time updates, leaving donors uncertain about the impact of their contributions. Moreover, traditional charity models suffer from accessibility limitations, as many fundraising initiatives do not cater to a global audience. Individuals who wish to donate but cannot attend in person may find it difficult to contribute due to limited payment options or geographic barriers. This restricts the overall reach and potential of charitable campaigns, reducing their effectiveness in addressing pressing social issues. Additionally, administrative costs and intermediary fees in traditional charity systems often reduce the actual amount reaching beneficiaries, further lowering donor confidence. Given these challenges, it is evident that a modern, technology-driven approach is necessary to enhance the efficiency, engagement, and transparency of fundraising initiatives. By integrating gamification, digital wallets, real-time tracking, and secure fund allocation mechanisms, an innovative entertainment-based charity model can address these issues, making charitable giving more accessible, interactive, and impactful.

2.2.1 LIMITATIONS OF EXISTING SYSTEMS

Traditional charity systems face several limitations that hinder their effectiveness, transparency, and scalability. One of the primary challenges is the reliance on manual fundraising events, which are often constrained by location, time, and resources. These events require significant planning and logistical efforts, limiting participation to those who can physically attend. Additionally, fundraising success depends largely on event popularity and visibility, making it challenging for smaller causes to secure consistent financial support. The absence of a structured engagement mechanism also leads to low donor retention, as most contributions are made on a one-time basis without any long-term involvement.

Another key limitation is the lack of transparency in fund allocation and impact assessment. Donors often have minimal insight into how their contributions are used, leading to trust issues and potential scepticism about fund management. Limited accessibility also poses a major challenge, as traditional fundraising efforts often fail to cater to a global audience. Geographic barriers, restricted payment options, and manual processing make it difficult for individuals worldwide to contribute seamlessly. Furthermore, the absence of gamification and incentives results in a lack of motivation for sustained participation, reducing overall engagement. Given these constraints, there is a need for a modern, technology-driven approach that enhances accessibility, engagement, and financial transparency in charity fundraising.

In addition to these limitations, traditional charity models often struggle to attract and retain younger, tech-savvy generations who expect digital convenience, real-time impact visibility, and social interaction in their philanthropic experiences. These donors are more inclined to engage with platforms that incorporate gamified elements, social sharing, and progress tracking features. Moreover, the absence of personalized donor journeys such as tailored recommendations based on giving history or cause preferences reduces the emotional connection between donors and the initiatives they support. Manual recordkeeping and delayed reporting also hinder timely responses to urgent needs, such as disaster relief or medical emergencies. Lastly, traditional systems often incur high administrative and intermediary costs, which dilute the actual funds reaching beneficiaries. A digital-first, gamified platform with built-in automation, personalized engagement, and transparent reporting can effectively address these issues, transforming charity fundraising into a more inclusive, scalable, and trust-driven ecosystem.

2.3 PROPOSED SYSTEM

The Panchayat Specific Charity Fund Collection Through Entertainment initiative revolutionizes fundraising by integrating gaming with social responsibility. Unlike traditional charity models, this system gamifies donations, encouraging engagement while ensuring transparency and accountability. By leveraging a MERN stack-based web application, the platform offers a seamless and interactive experience for users, hosts, and social workers. It enables gamified fundraising, allowing users to contribute through gameplay, making donations both engaging and enjoyable. A percentage of collected funds (70%) is directly allocated to verified charities, ensuring impactful contributions. To maintain transparent financial reporting, real-time financial tracking provides clear insights into fund collection and allocation.

A role-based collaboration system categorizes users into different roles. Normal users can play games, compete for rewards, access financial reports, and report fraudulent activities. Admins oversee platform operations, ensure transparency, and handle moderation, while social workers post verified charitable needs and connect with donors to address real-world issues. The reward system motivates users by providing the top scorer of each game with 20% of ₹10, which can be either withdrawn as cash or donated to charity. The secure and efficient payment system incorporates a wallet feature, allowing users to manage their winnings and donations seamlessly, track their transactions, and withdraw rewards when needed.

In addition to its innovative features, the platform prioritizes user engagement and retention through personalized experiences and gamified incentives. Each user's dashboard provides tailored updates on their contributions, game performance, and the real-world impact of their donations, enhancing emotional connection and trust. The inclusion of social sharing options allows users to promote causes they support, expanding the platform's reach organically and fostering a sense of community. Furthermore, the system's scalability makes it ideal for expansion into different regions and causes, enabling panchayats and grassroots organizations to raise funds efficiently. Built-in fraud detection mechanisms and admin oversight tools ensure the integrity of campaigns and fund allocation. By merging technology with social good, the Panchayat Specific Charity Fund Collection Through Entertainment platform not only bridges the gap between donors and beneficiaries but also redefines how communities can support each other through engaging and transparent digital ecosystems.

With global accessibility, the web-based platform eliminates geographical limitations, enabling users worldwide to participate and contribute. Additionally, automated reporting and monitoring ensure transparency by generating reports on game participation, fund allocation, and donation impact. This fosters accountability in charitable transactions. By implementing an intuitive web interface, a structured database, and real-time notifications, the Charity Through Entertainment platform creates a secure, transparent, and engaging ecosystem for fundraising and social impact.

2.3.1 OBJECTIVES

To Promote Social Responsibility

- Encourage users to engage in charitable activities by seamlessly integrating entertainment with impactful giving.
- Focus on Panchayat-specific charity fund collection, ensuring local communities benefit directly from contributions.
- Leverage gamification to make donating engaging and interactive.

To Ensure Financial Transparency

- Maintain clear, real-time records of fund collection, allocation, and usage.
- Provide users with access to detailed financial reports, ensuring transparency and building trust among contributors.
- Implement a structured database that allows tracking of transactions and fund distributions.

To Foster Community Collaboration

- Connect users, Panchayat authorities, admins, and social workers to address verified community needs effectively.
- Enable social workers to post verified charitable needs, ensuring funds are directed to critical local projects.
- Provide a communication channel where stakeholders can collaborate and oversee fund allocation.

To Inspire Healthy Competition and Engagement

- Motivate users to participate in fundraising activities by offering leaderboard-based rewards.
- Recognize the top contributors and players, ensuring their efforts are acknowledged and rewarded.

To Enable Efficient Fund Allocation

- Streamline the process of collecting and distributing funds to ensure they reach the right causes promptly.
- Implement a role-based access system where Panchayat authorities can validate and approve fund disbursement.
- Provide automated reports on fund usage, ensuring accountability and effective decision-making.

To Develop a Secure and User-Friendly System

- Create an intuitive web-based platform that makes participation accessible to all users.
- Implement secure payment processing and a wallet system to facilitate transactions.
- Use MongoDB for structured data storage, ensuring smooth and efficient platform performance.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 SYSTEM ANALYSIS

3.1.1 INTRODUCTION

System analysis is a crucial component in the development of the Panchayat Specific Charity Through Entertainment platform, as it involves evaluating existing charitable fundraising methods, identifying inefficiencies, and proposing a more engaging and transparent solution. Traditional fundraising approaches, especially at the local Panchayat level, often rely on manual donation drives and community events, which can be inefficient, lack transparency, and have limited reach.

The proposed system eliminates these inefficiencies by introducing an automated, gamified web-based platform utilizing the MERN stack (MongoDB, Express.js, React.js, and Node.js). This system ensures seamless donation tracking, real-time financial transparency, and structured fund allocation to verified charitable causes within the Panchayat. A key feature of the system is its gamified fundraising approach, where users contribute by playing games, making donations more engaging while fostering a sense of social responsibility. The real-time financial reporting system ensures donors can track where their contributions go, enhancing trust and accountability. Additionally, the system incorporates a role-based collaboration model, allowing Panchayat authorities, admins, and social workers to oversee fund collection and distribution efficiently.

By automating workflows and removing paper-based donation tracking, the system improves coordination between donors, Panchayat officials, and local communities. The integration of real-time notifications, secure financial processing, and a digital leaderboard for contributors ensures enhanced transparency, better community engagement, and a more efficient charitable ecosystem that directly benefits local initiatives.

Incorporating analytics and user engagement tracking, the platform also allows administrators to assess which games drive the most participation and optimize strategies accordingly. The modular design supports future expansions, such as seasonal campaigns, community voting on fund allocation, or integration with government welfare schemes, ensuring the system remains relevant and impactful over time.

3.1.2 METHODOLOGY

The development of the Panchayat Specific Charity Fund Collection Through Entertainment platform follows the Agile methodology, which is well-suited for this project due to its iterative approach, flexibility, and ability to incorporate continuous feedback. Agile allows for rapid adjustments based on user input, ensuring the platform evolves to meet both user engagement and financial transparency requirements throughout its development lifecycle.

The Software Development Life Cycle (SDLC) begins with the requirement analysis phase, where discussions with Panchayat authorities, social workers, and community members are conducted to identify key challenges in existing charity fundraising methods. This phase helps define the core system requirements to ensure transparency, efficiency, and engagement. During the design phase, architectural diagrams, entity-relationship (ER) models, and wireframes are created to define how various system components will interact, focusing on user-friendly navigation, real-time financial tracking, and secure fund allocation. The implementation phase involves:

- Frontend development using React.js to create an engaging and interactive user interface.
- Backend development using Node.js and Express.js, ensuring efficient server-side logic and API request handling.
- MongoDB as the database, securely storing user transactions, game records, and fund distribution details.

Rigorous testing is conducted to ensure the system functions as intended, addressing functionality, payment security, user authentication, and data reliability. Finally, the platform is deployed to a secure server, ensuring smooth integration into Panchayat-level charitable operations. Continuous maintenance and updates are provided based on real-time user feedback to enhance engagement, optimize performance, and improve transparency in fund management. In addition to these phases, regular sprint reviews and retrospectives are conducted to assess the progress and gather feedback from all stakeholders. This feedback loop helps the development team adapt quickly to evolving requirements and community needs. User testing sessions are also organized at the Panchayat level, allowing actual beneficiaries and donors to interact with the platform before full-scale deployment.

3.1.3 HARDWARE AND SOFTWARE REQUIREMENTS

The smooth operation of the Panchayat Specific Charity Fund Collection Through Entertainment platform depends on a combination of hardware and software components to ensure seamless functionality, secure transactions, and real-time interactions. On the hardware side, cloud-based or on-premise servers are required to host the application, ensuring reliability, scalability, and security. Client devices such as desktops, laptops, tablets, and smartphones are used by normal users, administrators, and social workers to access and manage the platform efficiently. A stable internet connection is essential for real-time game interactions, leaderboard updates, and secure payment processing, ensuring smooth user experience and uninterrupted financial transactions.

On the software side, the platform is built using the MERN stack, providing a robust, scalable, and high-performing application. The frontend is developed using React.js, ensuring an interactive and responsive user interface that enhances engagement and ease of navigation. The backend is powered by Node.js and Express.js, efficiently handling server-side logic, user authentication, API requests, and transaction processing. MongoDB, a NoSQL database, provides secure and scalable data storage, accommodating structured and semi-structured data related to user accounts, transactions, game scores, and donation records. To facilitate secure and efficient financial operations, the system integrates payment gateway APIs, allowing users to make payments, manage wallet transactions, and allocate funds to charities transparently. The leaderboard and rewards system are automated to update player rankings and distribute earnings to top players in real-time. Additionally, the platform incorporates an email and in-app notification system to keep users informed about game updates, rewards, and financial transactions.

In addition to these components, GitHub is used as the primary version control and collaboration tool during development. GitHub facilitates efficient source code management, enabling developers to track changes, manage branches, and collaborate on different modules without conflict. It also supports continuous integration workflows, code reviews, and issue tracking, which helps maintain code quality and accelerate the development cycle. Moreover, development tools such as Visual Studio Code for code editing and Postman for API testing and browser developer tools further support the build and debugging process. Integration of these tools ensures a professional, organized, and scalable approach to both development and long-term maintenance of the platform.

3.2 SYSTEM DESIGN

3.2.1 INTRODUCTION

The system design phase focuses on defining the architecture, data flow, and functionality of the Panchayat Specific Charity Fund Collection Through Entertainment platform, ensuring that all components work together efficiently to create a secure, scalable, and transparent system for community-driven fundraising. This phase establishes how data will be processed, stored, and accessed, ensuring seamless gameplay, fund collection, and real-time financial tracking. The system is designed to support different user roles, including normal users, administrators, and social workers, each with specific access controls and responsibilities. Key design considerations include secure payment processing, automated fund allocation to panchayat-approved charitable causes, real-time leaderboard tracking for user engagement, and transparency in financial transactions. The platform integrates real-time notifications to keep users informed about their contributions, rewards, and impact on community projects. By leveraging role-based access control, secure data management, and automated workflows, the system ensures that all transactions and activities align with the platform's goal of combining entertainment with meaningful social contributions at the panchayat level.

To further support scalability and maintainability, the system design also includes modular components that allow for the seamless integration of additional games, user features, and regional customizations. Visual modelling tools such as data flow diagrams and entity relationship models are employed during the design phase to represent data movement and system structure clearly. These models assist developers in maintaining consistent data handling and ensure that each component aligns with the overall architecture. Moreover, provisions for multilingual support and device responsiveness are included to cater to diverse user groups across different panchayats, enhancing accessibility and user satisfaction.

3.2.2 MODULE DESCRIPTION

The Panchayat Specific Charity Fund Collection Through Entertainment platform is structured into multiple interconnected modules, each playing a crucial role in ensuring smooth operation, transparency, and user engagement. This system is designed to facilitate seamless user authentication, efficient fund collection, secure transactions, and automated reward distribution while maintaining accountability in charitable contributions. By integrating gaming with social responsibility, the platform enables users to participate in entertainment-driven fundraising while ensuring that donations are directed to verified

charitable needs within the panchayat. The platform incorporates a secure payment system, a leaderboard-based reward mechanism, and an emergency alert feature to enhance user experience and community support. Additionally, it includes automated receipt generation, a comprehensive analytics system for tracking fund allocation, and an announcement module for updating users on community activities. Each module is designed to work collaboratively, ensuring that every transaction is recorded transparently and every contribution makes a tangible impact. This structured approach allows for an engaging yet responsible gaming environment, where players contribute to social causes while enjoying their favourite games. The modular design also allows for easy scalability, making it possible to introduce new games, add more panchayat-specific causes, and expand user features as community needs evolve. Real-time synchronization between modules ensures up-to-date data access, enabling users, administrators, and social workers to make informed decisions. Furthermore, integration with third-party APIs for authentication, payments, and geolocation services enhances functionality while maintaining high standards of security and performance. Through this cohesive and extensible architecture, the platform creates a sustainable model for community-driven fundraising, blending entertainment, technology, and social impact.

1. Registration and Login Module

The Registration and Login Module manages user authentication and access control. Users, including normal players, administrators, and social workers, can register and log in based on their assigned roles. The module ensures secure access through password encryption and optional multi-factor authentication. New users must verify their accounts via email before accessing the platform.

2. Charity Management Module

The Charity Management Module allows social workers to submit and manage requests for financial aid on behalf of their communities. Each charity post undergoes an approval process by administrators before being listed for donations. The module ensures transparency by tracking and displaying the total funds collected and allocated for each verified cause.

3. Content Management Module

The Content Management Module is responsible for organizing and managing user-generated posts. Administrators can review, approve, or reject charity requests, event updates, and other platform-related content. This ensures that only verified and relevant information is displayed on the platform.

4. Game Management Module

The Game Management Module handles all aspects of the gaming experience. It enables administrators to add and update games, manage game settings, and monitor leaderboard statistics. The module ensures fair play by tracking user participation and verifying high scores before rewarding players.

5. Payment Module

The Payment Module facilitates secure transactions on the platform. It integrates multiple payment gateways, allowing users to pay for game participation and make direct donations. It also supports wallet-based transactions, where players can store their rewards and use them for future payments or donations.

6. Game Donation Module

The Game Donation Module automates the process of allocating a percentage of game revenue to verified charity causes. A fixed percentage of each game fee is directed toward panchayat-approved charity posts, ensuring that gaming activities contribute to social good. The module provides real-time tracking of funds collected through games.

7. Transaction Module

The Transaction Module records and processes all financial transactions, including payments, donations, withdrawals, and fund allocations. It ensures financial transparency by generating transaction logs, which users and administrators can access at any time. This module also prevents fraudulent activities by monitoring unusual transaction patterns.

8. Leadership and Reward Module

The Leadership and Reward Module enhances user engagement by implementing a reward-based system. Players who perform well in games can earn points and monetary rewards, which can either be withdrawn or donated to charity.

9. Payment Receipt and Email Notifications Module

This module generates automatic receipts for all financial transactions, allowing users to download them for their records. It also sends email notifications to users after successful payments, withdrawals, and donations. Additionally, the module delivers urgent notifications during emergency fundraising campaigns.

10. Emergency Alert Module

The Emergency Alert Module is designed to send urgent notifications regarding critical situations, such as natural disasters or urgent community needs. Administrators can issue alerts requesting immediate financial aid, ensuring a rapid response from donors and players.

11. Announcement Module

The Announcement Module allows administrators to post updates on local body issues, community events, and important notices. It serves as an information hub where users can stay informed about upcoming charity drives, gaming tournaments, and policy changes.

12. Reports and Analytics Module

The Reports and Analytics Module provides detailed insights into platform activities, including donation trends, user participation rates, and game revenue statistics. Administrators can generate customized reports to assess the impact of fundraising campaigns and optimize future strategies.

13. Feedback and Review Module

The Feedback and Review Module allows users to submit their opinions and suggestions regarding the platform's functionality, gaming experience, and charity initiatives. It helps administrators identify areas for improvement and implement changes based on user feedback.

3.2.3 SYSTEM ARCHITECTURE

The Panchayat Specific Charity Fund Collection Through Entertainment platform is designed to facilitate transparent and efficient fundraising through interactive gaming, ensuring that collected funds are directed toward verified charitable needs within local communities. The use case diagram for this system visually represents how different users (actors) interact with the platform's key functionalities. The primary actors in the system include Normal Users, Admins, and Social Workers, each playing a distinct role in maintaining the platform's operations. Normal Users participate in games, contribute funds, and receive rewards based on performance. Admins oversee platform management, ensuring smooth transactions, approving content, and maintaining financial transparency. Social Workers verify and post charitable needs, ensuring that donations reach the intended beneficiaries. The use case diagram illustrates these interactions and the flow of actions within the system.

It highlights how a user's gaming participation triggers a structured process: from payment processing and donation allocation to leaderboard updates and reward distribution. The diagram also showcases key dependencies, such as how donations from game activities are recorded, verified, and disbursed based on system rules. This visual representation provides a clear overview of the platform's functionality, emphasizing automation, security, and transparency in fund collection and distribution. By integrating entertainment with social responsibility, the system enhances user engagement while promoting charitable contributions within the panchayat.

Additionally, the platform fosters a sense of community-driven impact by allowing users to view how their contributions are being utilized in real-time. It bridges the gap between donors and beneficiaries by providing transparency dashboards and activity logs. The use case diagram serves not only as a technical blueprint but also as a strategic communication tool, helping stakeholders understand the user journey and the system's commitment to ethical and impactful fundraising. This structured interaction model ensures accountability at every step, reinforcing trust and encouraging continuous participation in the platform's mission.

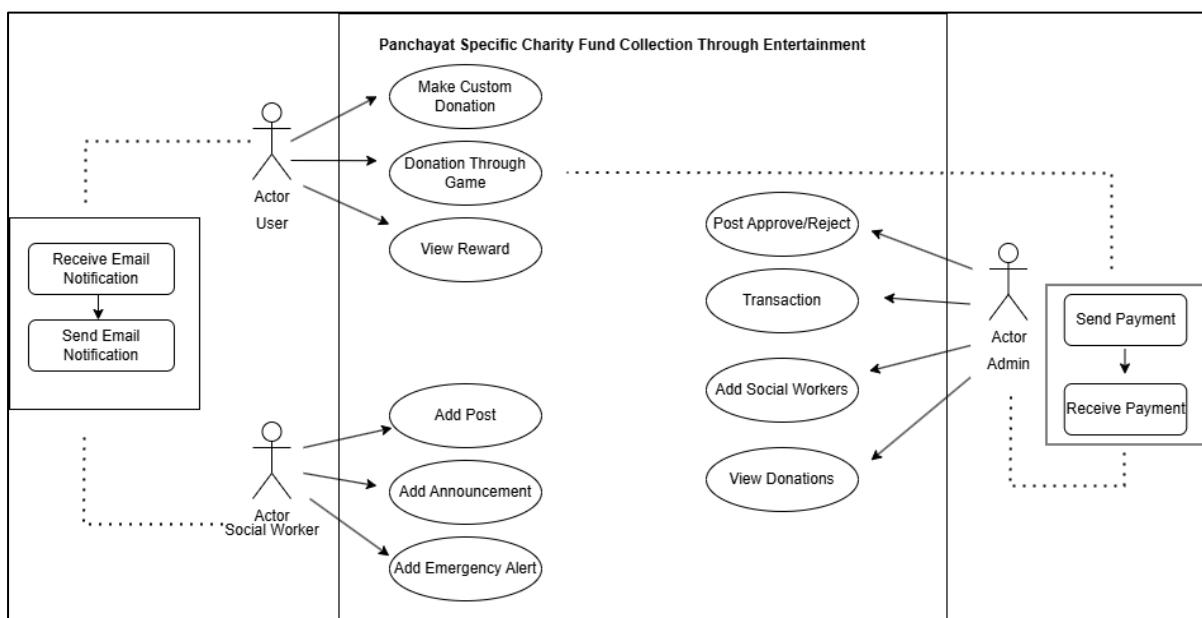


Fig 3.2.3: Use Case Diagram

3.2.4 DATABASE

The Panchayat-Specific Charity Fund Collection Through Entertainment platform utilizes a well-structured database to manage and process all relevant data concerning user activities, charity fund transactions, gaming rewards, and community interactions. The database plays a crucial role in supporting platform functionalities such as user registrations, donations, payment processing, game-based transactions, and community engagement. Efficient data organization ensures smooth interactions between various actors, including users, social workers, and administrators, while maintaining data integrity and security.

The database is designed to automate and digitize many processes, reducing manual effort, minimizing errors, and enhancing the overall efficiency of the platform. By leveraging interrelated tables, the system ensures structured data storage, quick retrieval, and real-time updates. Key tables include User Details, Donation Records, Payment Transactions, Game Scores, Wallet Balances, Community Announcements, Alerts, and Social Worker Posts. By maintaining a centralized and robust database, the platform ensures seamless financial transactions, secure fund allocations, and real-time engagement with users. The database serves as the backbone of the system, enabling smooth communication, transparent charity management, and scalable growth for future enhancements.

1. USER TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for user.
name	String	User name
username	String	User login name
address	String	User address
email	String	User email id
phone	String	User phone number
ward_no	Integer	User ward number
ward_name	String	User ward name
password	String	User account password

Table 3.4.1: User Table

This table stores essential information for each user of the system such as identification, contact details and other additional data. The design allows for scalable user management and supports communication, role-based access, and operational functionalities.

2. POSTS TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for post.
title	String	Post title
description	String	Post description
name	String	Name of the person
age	Integer	Age of the person
purpose	String	Purpose of the post
contact	String	Contact of the person
image	String	Image of relevant documents
requiredAmount	String	Required amount of donation
accountName	String	Bank account name of person
accountNo	String	Bank account number of person
ifsc	String	Bank IFSC code
bankName	String	Bank name of person
currentDonationsReceived	String	Donation amount received
status	String	Status of post
rejectionReason	String	Reason for post rejection
createdAt	Date	Time of post creation

Table 3.4.4: Posts Table

This table stores essential information for adding the charity post in the system such as the cause, the person needed the help, communication details, bank details etc. The design allows the admin to approve or reject the post. If approved, user can donate accordingly.

3. SOCIAL WORKERS TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for social worker.
name	String	Social worker name
email	String	Social worker email id
address	String	Social worker address
phone	String	Social worker phone number
password	String	Social worker account password

Table 3.4.2: Social Worker Table

This table stores essential information for each social worker of the system such as identification, contact details and other additional data. It is added by the admin.

4. ADMIN TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for admin.
email	String	Admin email id
password	String	Admin account password

Table 3.4.3: Admin Table

This table stores essential information for admin of the system such as email and password. The design allows for scalable admin management and supports role-based access.

5. GAME DONATIONS TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for game donation.
userId	String	User ID of user
amount	String	Default amount for game charity
method	String	Method of payment
status	String	Status of payment
createdAt	Date	Time of payment

Table 3.4.5: Game Donations Table

This table stores essential information for storing the details on game donations. The default amount taken is 10 and allows the payment to be added with the charity.

6. PAYMENTS TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for payment.
userId	String	User ID of user
postId	String	Post ID of particular post
amount	String	Amount for charity
method	String	Method of payment
status	String	Status of payment
createdAt	Date	Time of payment

Table 3.4.6: Payments Table

This table stores essential information of custom payments in the system such as the post to which it is donated, the user details, amount etc. The design allows the admin to track the payments.

7. REWARD TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for reward.
userId	String	User ID of user
upiId	String	UPI ID of the user
amount	String	Amount as reward
claimedAt	Date	Date of claiming reward

Table 3.4.7: Reward Table

This table stores essential information regarding the rewards which is obtained by the user for high scores in the games each day. The design allows the admin to track the rewards.

8. ANNOUNCEMENT TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for announcement.
topic	String	Announcement Topic
likes	String	Number of likes received
createdAt	Date	Date of announcement creation

Table 3.4.8: Announcement Table

This table stores essential information about adding the announcement by social workers. It could any public information for the citizens.

9. PLATFORM EARNINGS TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for platform earning.
amount	String	Amount received
receivedAt	Date	Date of amount received

Table 3.4.10: Platform Earnings Table

This table stores essential information about the platform earning that is obtained from the game donations. Admins keep the record of this.

10. EMERGENCY TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for alert.
title	String	Emergency alert title
description	String	Emergency alert description
location	String	Location of the alert
ward_no	Integer	Ward number of alert
alertType	String	Type of alert
reports	String	Number of people reported
createdAt	Date	Date of alert created

Table 3.4.9: Emergency Table

This table stores essential information about adding the emergency alert by social workers regarding any important issues like flood, fire or accident. It could be any public information for the citizens. The users with the ward number specified by the social worker in the post receives an email regarding the same.

11. TRANSACTIONS TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for transaction.
postId	String	Post ID of particular post
requiredAmount	String	Required amount of donation
accountName	String	Bank account name of person
accountNo	String	Bank account number of person
ifsc	String	Bank IFSC code
bankName	String	Bank name of person
status	String	Status of transaction
createdAt	Date	Date of transaction

Table 3.4.11: Transactions Table

This table stores essential information about the posts that completed the donation. This allows the admin to pay the collected amount to the respective person.

12. WALLETS TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for wallet.
userId	String	User ID of user
balance	String	Balance amount in the wallet
transactions	String	Type of transaction
amount	String	Amount in wallet
date	Date	Date of amount added

Table 3.4.12: Wallets Table

This table stores information regarding the amount earned by the users after playing the games. Users get a portion of the donated amount if they achieve the high score. It is stored and is claimed back.

13. REVIEW TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for review.
userId	String	User ID of user
review	String	Review about the application
rating	String	Rating about the application
postedDate	Date	Date of review added

Table 3.4.13: Review Table

This table stores the information about adding review and rating about the application and services. It is managed by admin.

14. TIC TAC TOE TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for tic tac toe.
score	String	User ID of user
createdAt	Date	Date of review added

Table 3.4.14: Tic Tac Toe Table

This table stores essential information about high score of the tic tac toe game and finds the high scorer each day.

15. SNAKE GAME TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for snake game.
score	String	User ID of user
createdAt	Date	Date of review added

Table 3.4.15: Snake Game Table

This table stores essential information about high score of the game and finds the high scorer.

16. QUIZ TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for quiz.
score	String	User ID of user
createdAt	Date	Date of review added

Table 3.4.16: Quiz Table

This table stores essential information about high score of the quiz game and finds the high scorer each day.

17. HANGMAN GAME TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for hangmen.
score	String	User ID of user
createdAt	Date	Date of review added

Table 3.4.17: Hangman Table

This table stores essential information about high score of the hangman game and finds the high scorer each day.

18. GUESS THE NUMBER GAME TABLE

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for game.
score	String	User ID of user
createdAt	Date	Date of review added

Table 3.4.18: Guess The Number Table

This table stores essential information about high score of the guess the number game and finds the high scorer each day.

3.3 ISSUES FACED AND REMEDIES TAKEN

3.3.1 ISSUES

During the development and implementation of the Panchayat Specific Charity Fund Collection Through Entertainment platform, several challenges emerged that affected system functionality, security, and user engagement. These challenges needed to be addressed to ensure seamless operations and a positive user experience. Integration of the payment gateway required extensive testing to ensure secure and successful transactions, especially in cases of network interruptions. Ensuring real-time synchronization of game scores and wallet updates also posed technical difficulties due to latency issues and concurrent user actions. Additionally, maintaining financial transparency while protecting sensitive user data required careful implementation of encryption and access control mechanisms.

1. **Payment Processing Delays and Failures:** The platform relied on online transactions for user participation and charity fund distribution. Initially, delays and occasional failures in processing payments were reported due to third-party gateway limitations and transaction verification issues. This affected user trust and the timely allocation of funds to charities.
2. **User Engagement and Retention:** Encouraging consistent user participation in gaming activities was a challenge. While initial engagement was high, retention rates dropped due to a lack of personalized incentives and interactive features. Users needed more motivation to continue contributing to charity through entertainment.
3. **Fraud Prevention and Security Risks:** As a financial platform handling user payments and donations, ensuring transaction security and preventing fraudulent activities were critical. Fake accounts and attempts to exploit the reward system were identified as potential risks that could impact financial transparency.
4. **Scalability and Performance Bottlenecks:** As the number of users, transactions, and game interactions increased, the platform experienced slower response times, especially when retrieving leaderboard data and processing reward allocations. The database structure and server capacity required optimization to accommodate future growth efficiently.
5. **Financial Transparency and Fund Allocation Clarity:** Since the platform operates on a revenue-sharing model, ensuring that users, charities, and administrators had clear visibility into fund distribution was a challenge. Users needed real-time updates on how their contributions were allocated to charities.

3.3.2 REMEDIES

To address these challenges, several strategic solutions were implemented to improve system efficiency, security, and user engagement. Robust authentication mechanisms and data encryption techniques were integrated to ensure secure user access and protect sensitive information. Real-time synchronization techniques and optimized backend logic were employed to minimize latency in game and wallet updates. Cross-platform compatibility and responsive UI design were also prioritized to enhance user experience across devices. Regular testing and feedback loops were incorporated to identify issues early and refine system performance continuously.

1. **Optimized Payment Gateway Integration:** To resolve transaction delays, the platform integrated multiple payment gateways with automatic fallback mechanisms. Secure and efficient APIs were used to reduce processing time, and transaction monitoring tools were implemented to detect and resolve failed payments in real time. Additionally, users were notified instantly about their payment status via automated email and SMS alerts.
2. **Enhanced User Engagement Strategies:** To boost retention, personalized rewards and gamification elements were introduced. A loyalty points system was integrated, allowing users to earn bonuses for consistent participation. Additionally, AI-driven recommendations suggested relevant games and charity campaigns based on user preferences, increasing engagement levels.
3. **Robust Fraud Detection and Security Measures:** Multi-layered authentication was implemented, including OTP verification for sign-ups and payments. AI-based fraud detection algorithms monitored unusual transaction patterns and flagged suspicious activities. Strict KYC (Know Your Customer) verification was enforced for large transactions and withdrawals to prevent abuse of the reward system.
4. **Performance and Scalability Enhancements:** Database indexing and caching techniques were applied to speed up query response times. Leaderboard and reward allocation processes were optimized using batch processing to reduce server load.
5. **Transparent Financial Reporting System:** A dedicated financial transparency dashboard was implemented, where users could track their contributions, reward earnings, and fund allocations in real time. Automated reports were generated to provide clear insights into revenue distribution (70% to charities, 20% to rewards, and 10% platform fees). Blockchain-based verification was explored to enhance transparency and build user trust in the system.

CHAPTER 4

RESULT

4.1 RESULT SCREENSHOT

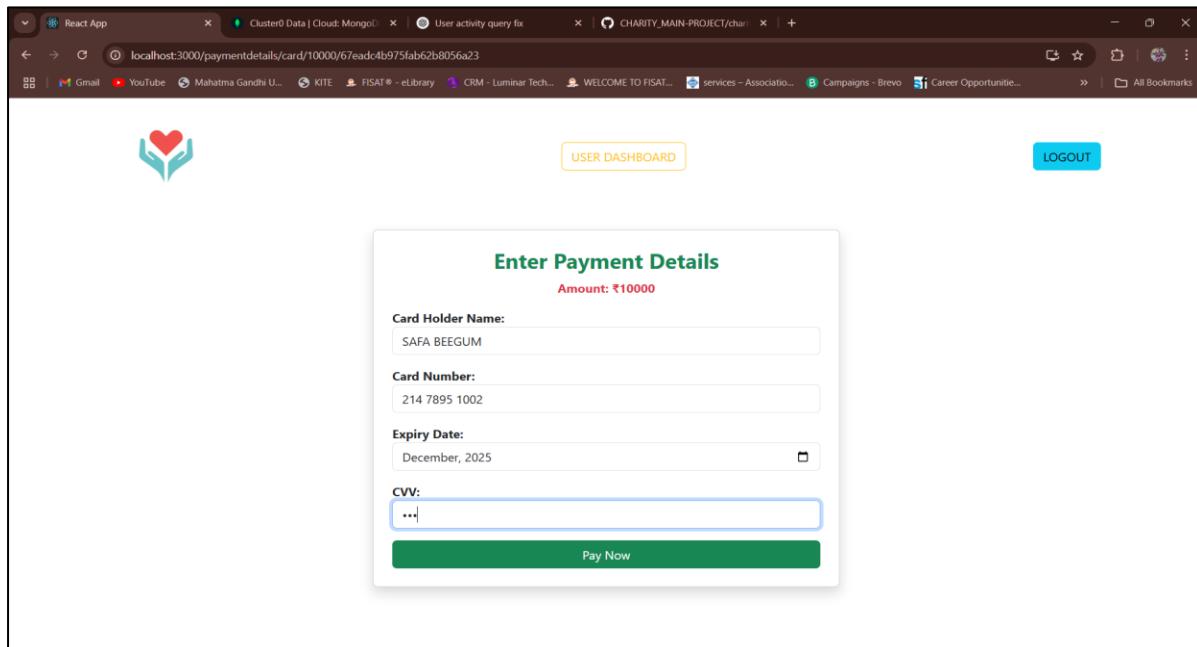


Fig 4.1: Donation Payment by User

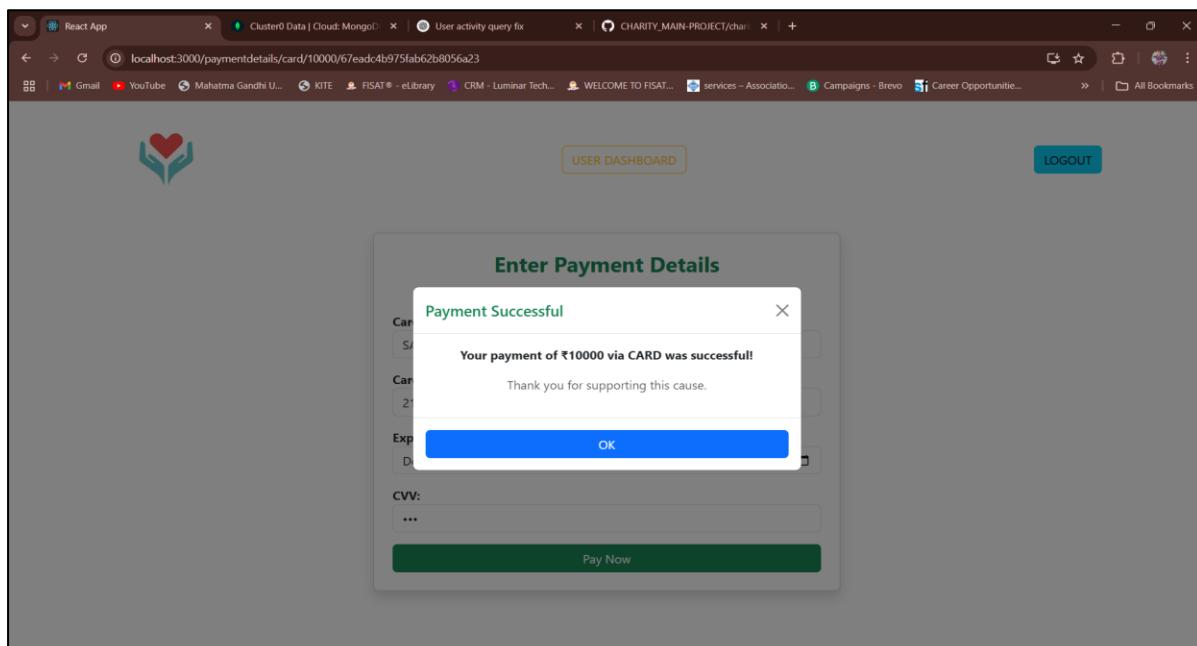


Fig 4.1: Payment Successful Alert

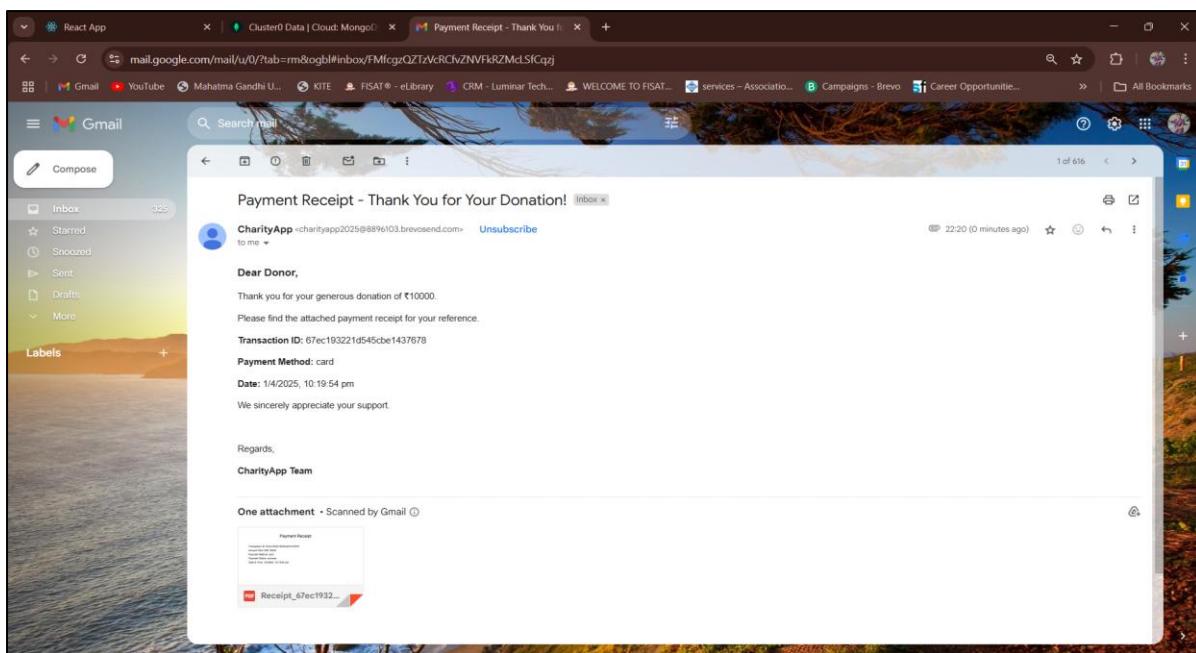


Fig 4.1: Email Receipt after Successful Payment

The payment confirmation system in the Panchayat Specific Charity Fund Collection Through Entertainment platform ensures secure and transparent financial transactions during game participation and charity contributions. Upon successful payment, the system automatically generates a confirmation notification, which is promptly sent via email to the user, detailing the complete transaction information.

This automated process eliminates the need for manual receipts, thereby improving operational efficiency and simplifying financial record-keeping within the platform. The instant notification reassures both users and administrators that the transaction has been securely processed, while the email receipt acts as verifiable proof of payment for future reference.

By integrating real-time alerts with a dependable digital trail, the system enhances financial transparency and builds trust among all stakeholders ensuring that every contribution, whether for gameplay or charity, is accurately recorded and acknowledged. Additionally, the system logs every transaction in a secure database for future auditing and reporting, further reinforcing accountability. Users also have access to a payment history dashboard within their profile, allowing them to review past payments, track wallet balance updates, and monitor donation allocations. This end-to-end transparency not only increases user confidence but also encourages repeat participation and ongoing support for local charitable initiatives.

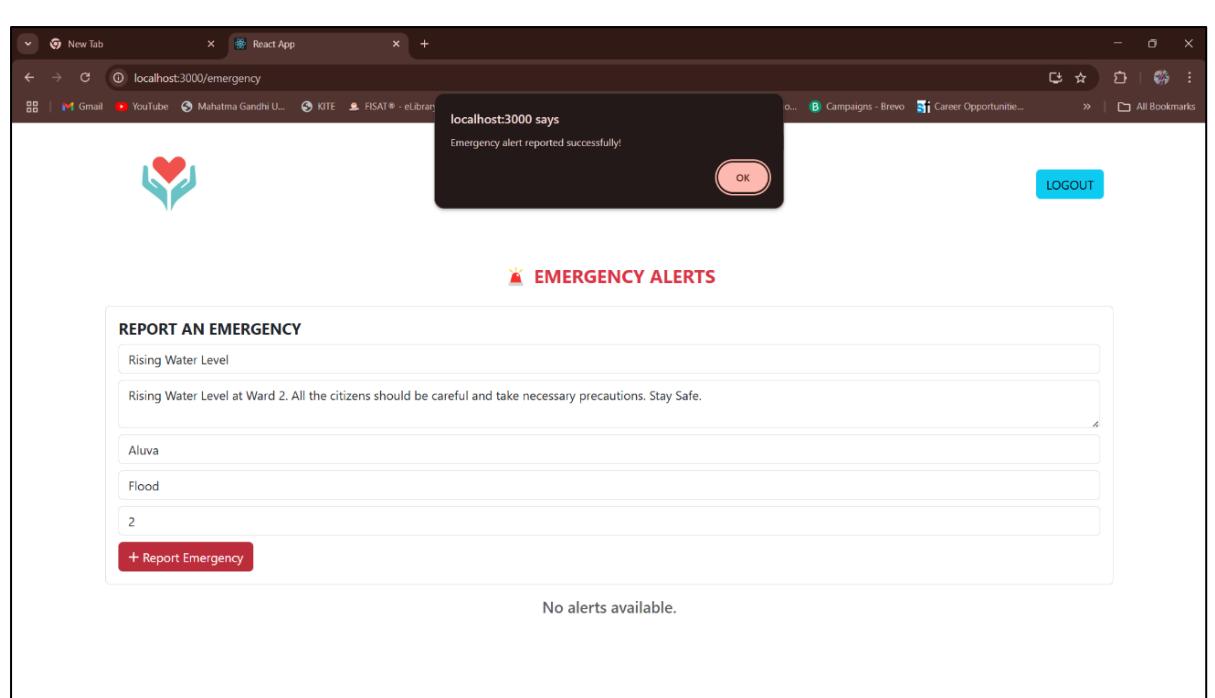


Fig 4.1: Adding Emergency Alert by Social Worker

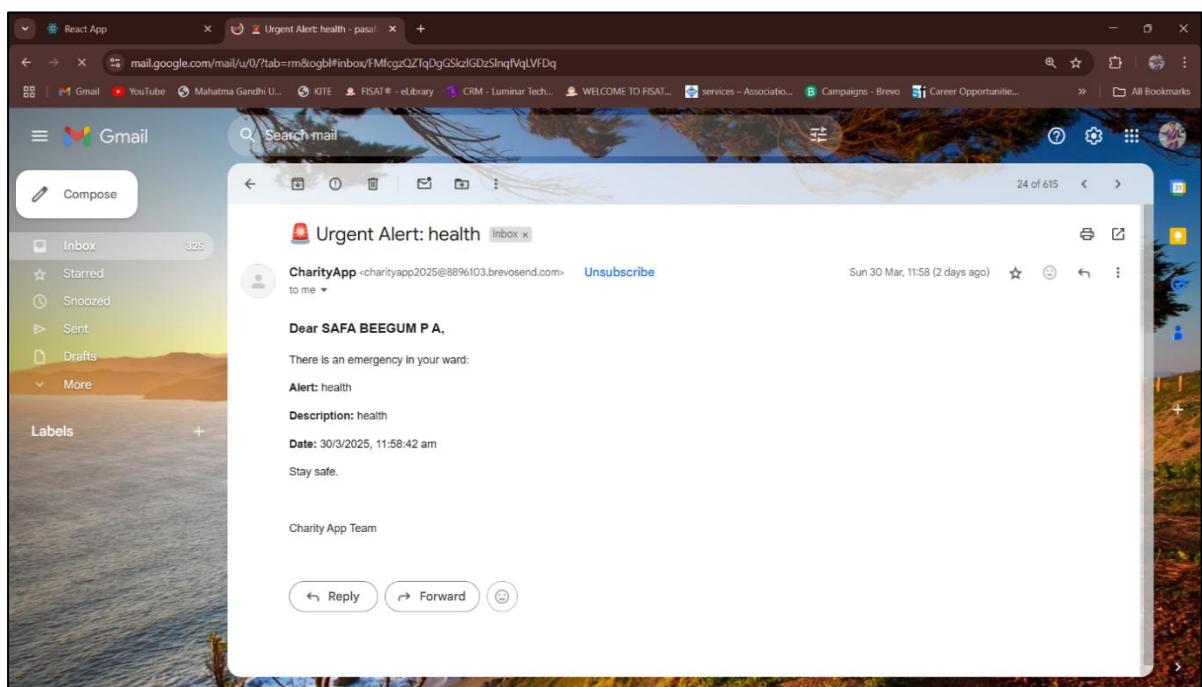


Fig 4.1: Email Notification for Emergency Alert

The Emergency Alert and Notification System instantly informs users of urgent needs within their Panchayat ward. When an alert is created, users with the same ward number receive detailed email notifications. These emails include emergency details, location, urgency, and suggested actions. The system ensures timely, targeted responses, strengthening community support and engagement.

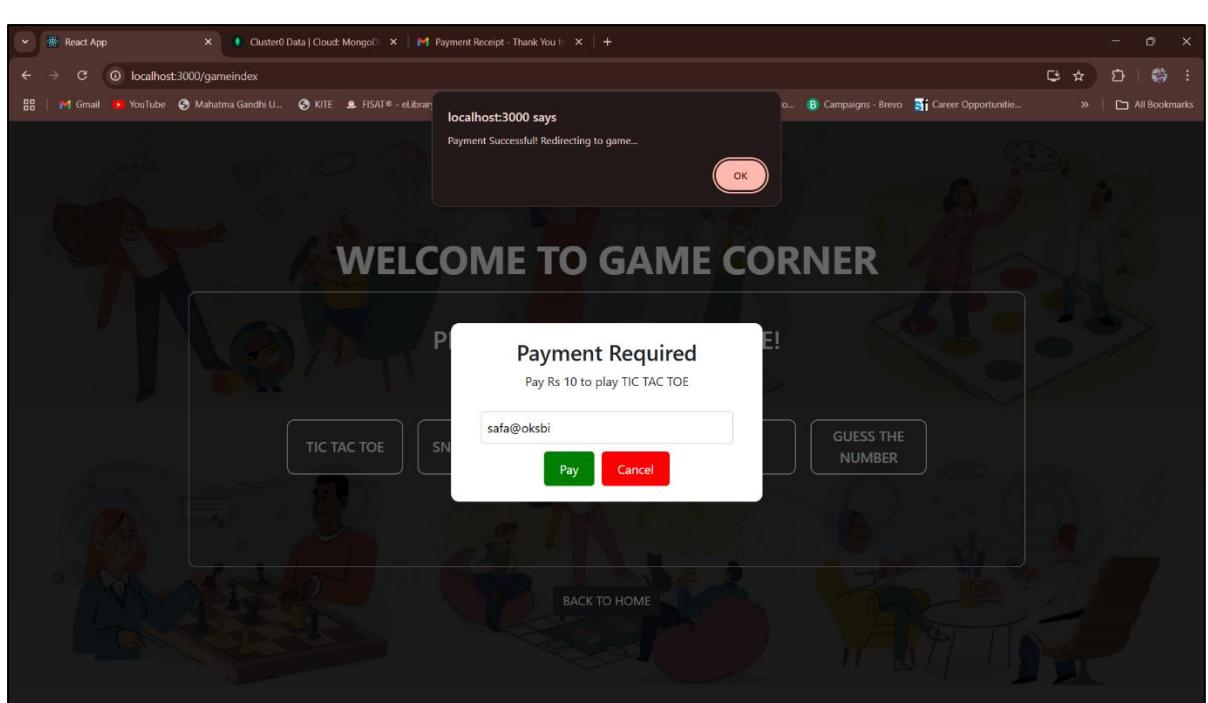


Fig 4.1: Game Payment Donation Interface

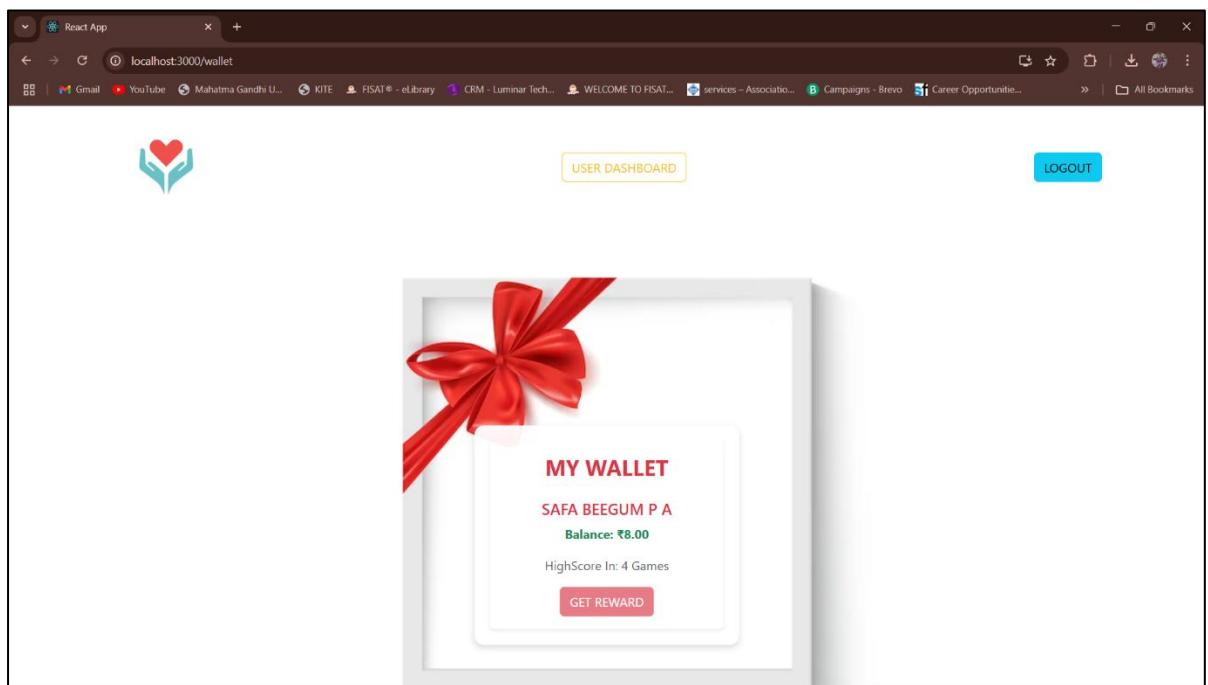


Fig 4.1: Wallet Amount Earned by User

The Game Play and Wallet Generation System rewards users for participating in games by crediting their in-platform wallet based on performance. Each win earns a fixed amount updated in real-time and visible on the user dashboard. Users can donate their wallet balance to verified causes or withdraw it as cash. This system boosts engagement while supporting transparent, gamified fundraising.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

The Panchayat Specific Charity Fund Collection Through Entertainment platform marks a significant step toward leveraging technology for social impact. By integrating entertainment with charitable contributions, the system ensures transparency, engagement, and seamless fund distribution. Replacing traditional donation methods with an interactive and gamified approach encourages more user participation and sustained contributions.

One of the most critical improvements of this system is the structured and automated financial tracking, ensuring that funds are allocated correctly—70% to verified charities, 20% as player rewards, and 10% as a platform fee. The wallet system enhances transparency by allowing users to track their earnings and donations in real time. This structured approach ensures that contributors have clear visibility into their impact, strengthening trust in the platform.

The use of the MERN stack (MongoDB, Express.js, React.js, Node.js) provides scalability and flexibility, making it easy to implement updates, introduce new games, and integrate additional features in the future. To ensure a seamless and secure experience, robust fraud prevention mechanisms, including multi-layered authentication and AI-driven transaction monitoring, were integrated to protect user data and funds.

Additionally, the leaderboard and reward system incentivize user participation by offering tangible benefits, boosting engagement, and encouraging continuous contributions. The financial transparency dashboard allows users to monitor fund distribution, making the donation process more accountable.

Overall, the Panchayat-Specific Charity Fund Collection Through Entertainment platform revolutionizes fundraising by making it more interactive, engaging, and transparent. By harnessing modern technology, the system fosters community-driven charity efforts while ensuring financial integrity and accountability. The success of this initiative paves the way for future enhancements, expanding its reach and impact on local communities in need.

5.2 FUTURE ENHANCEMENT

While the Panchayat-Specific Charity Fund Collection Through Entertainment platform effectively addresses current challenges in fundraising and community engagement, there is significant potential for future enhancements to improve its impact, security, and user experience. One of the major planned upgrades is the integration of a mobile application, allowing users to participate in games, track their earnings and donations, and receive real-time notifications directly from their smartphones. A dedicated app would make the platform more accessible and user-friendly.

To enhance communication, the platform can introduce SMS and WhatsApp alerts, ensuring that users receive timely updates about leaderboard rankings, fund allocations, and transaction confirmations, even if they do not regularly check emails. Another key improvement would be blockchain integration to provide a transparent and tamper-proof record of all transactions, ensuring donors and charities can verify the flow of funds with complete trust.

AI-powered analytics and recommendation systems can be integrated to personalize user experiences, suggesting relevant games, causes, and engagement opportunities based on past activity. Additionally, a cloud-based infrastructure can be implemented to enhance scalability, ensuring the system can handle increasing numbers of users, transactions, and game interactions efficiently.

The platform can also expand its accessibility by introducing multi-language support, making it easier for users from diverse linguistic backgrounds to navigate and engage with the system. Another critical enhancement would be fraud prevention mechanisms, incorporating AI-driven transaction monitoring and two-factor authentication to prevent unauthorized access and financial manipulation.

To further incentivize engagement, the introduction of community-driven events and seasonal challenges can keep users motivated while increasing contributions toward charitable causes. Gamification elements such as badges, achievements, and referral rewards can also be introduced to drive user participation and platform growth. With these future enhancements, the Panchayat-Specific Charity Fund Collection Through Entertainment platform can evolve into a more advanced, secure, and engaging system that maximizes social impact.

CHAPTER 6

APPENDIX

6.1 SOURCE CODE

Register.jsx

```
import axios from "axios";
import React, { useState } from "react";
import Nav from "./Nav";
const Register = () => {
  const [input, setInput] = useState({
    name: "",
    username: "",
    address: "",
    email: "",
    phone: "",
    ward_no: "",
    ward_name: "",
    password: "",
    confirmpass: ""
  });

  const inputHandler = (event) => {
    const { name, value } = event.target;

    if (name === "phone") {
      if (!/^d*$.test(value)) return;
      if (value.length > 10) return;
    }

    setInput({ ...input, [name]: value });
  };

  const readValue = async (event) => {
    event.preventDefault();

    const emailRegex = /^[^s@]+@[^s@]+\.[^s@]+$/;
    const phoneRegex = /^d{10}$/;
    const passwordRegex =
      /^(?=.*[0-9])(?=.*[@#$%^&*])[A-Za-z0-9!@#$%^&*]{6,}$/;

    // Check for empty fields
    for (let key in input) {
      if (input[key].trim() === "") {
        alert(`Please fill in the ${key.replace("_", " ")}.toUpperCase()} field.`);
      }
    }
  };
}
```

```

        }

    }

if (!emailRegex.test(input.email)) {
    alert("Please enter a valid email address!");
    return;
}

if (!phoneRegex.test(input.phone)) {
    alert("Please enter a valid 10-digit phone number!");
    return;
}

if (!passwordRegex.test(input.password)) {
    alert(
        "Password must be at least 6 characters, include 1 digit, and 1 special character!"
    );
    return;
}

if (input.password !== input.confirmpass) {
    alert("Passwords do not match!");
    return;
}

try {
    const response = await axios.post(
        "http://localhost:3030/register",
        input
    );

    console.log(response.data);
    if (response.data.status === "Success") {
        alert("Registered Successfully!!!");

        setInput({
            name: "",
            username: "",
            address: "",
            email: "",
            phone: "",
            ward_no: "",
            ward_name: "",
            password: "",
            confirmpass: ""
        });
    } else {
        alert("Email ID Already Exists!!!");
    }
} catch (error) {
    console.error("Error during registration:", error);
    alert("There was an error processing your request.");
}

```

```

    }
};

return (
<div>
<Nav />
<div className="container">
<div className="row">
<div className="col-lg-6">
<div className="card border-light mb-3">

</div>
</div>
<div className="col-lg-6">
<br><br>
<div className="card border-light mb-3">
<div className="row g-3">
<div className="col-6">
<label className="form-label">NAME</label>
<input
  type="text"
  className="form-control"
  name="name"
  value={input.name}
  onChange={inputHandler}
/>

</div>
<div className="col-6">
<label className="form-label">USERNAME</label>
<input
  type="text"
  className="form-control"
  name="username"
  value={input.username}
  onChange={inputHandler}
/>

</div>
<div className="col-12">
<label className="form-label">ADDRESS</label>
<textarea
  name="address"
  className="form-control"
  value={input.address}

```

```

        onChange={inputHandler}
    ></textarea>
</div>
<div className="col-6">
    <label className="form-label">EMAIL</label>
    <input
        type="text"
        className="form-control"
        name="email"
        value={input.email}
        onChange={inputHandler}
    />

</div>
<div className="col-6">
    <label className="form-label">CONTACT NUMBER</label>
    <input
        type="text"
        className="form-control"
        name="phone"
        maxLength="10"
        value={input.phone}
        onChange={inputHandler}
    />

</div>
<div className="col-6">
    <label className="form-label">WARD NUMBER</label>
    <select
        name="ward_no"
        className="form-control"
        value={input.ward_no}
        onChange={inputHandler}
    >
        <option value="">---SELECT HERE---</option>
        {[...Array(10).keys()].map((num) => (
            <option key={num + 1} value={num + 1}>
                {num + 1}
            </option>
        )))
    </select>
</div>
<div className="col-6">
    <label className="form-label">WARD NAME</label>
    <input
        type="text"
        className="form-control"
        name="ward_name"
        value={input.ward_name}
        onChange={inputHandler}
    />

```

```
>
</div>
<div className="col-6">
  <label className="form-label">PASSWORD</label>
  <input
    type="password"
    name="password"
    className="form-control"
    value={input.password}
    onChange={inputHandler}
  />
</div>
<div className="col-6">
  <label className="form-label">CONFIRM PASSWORD</label>
  <input
    type="password"
    name="confirmpass"
    className="form-control"
    value={input.confirmpass}
    onChange={inputHandler}
  />
</div>
<div className="col-6">
  <div className="d-grid gap-2">
    <button onClick={readValue} className="btn btn-success">
      REGISTER
    </button>
  </div>
</div>
<div className="col-6">
  <div className="d-grid gap-2">
    <a href="/login" className="btn btn-secondary">
      LOGIN PAGE
    </a>
  </div>
</div>
</div>
</div>
</div>
</div>
);
};
```

MakePayment.jsx

```
import React, { useState } from "react";
import { useNavigate, useParams } from "react-router-dom";
import UserNavbar from "./UserNavbar";
const MakePayment = () => {
  const [amount, setAmount] = useState("");
  const [method, setMethod] = useState("");
  const navigate = useNavigate();
  const { postId } = useParams();
  const handleNext = (e) => {
    e.preventDefault();
    const userId = sessionStorage.getItem("userId");
    if (!userId) {
      alert("Session expired! Please log in again.");
      navigate("/login");
      return;
    }
    if (!amount || amount <= 0) {
      alert("Enter a valid amount");
      return;
    }
    sessionStorage.setItem("userId", userId);
    navigate(`/paymentdetails/${method}/${amount}/${postId}`);
  };
  return (
    <div
      style={{ {
        minHeight: "100vh",
        background: "linear-gradient(to right, #f8f9fa, #eef2f3)",
      }}>
      <UserNavbar />
      <div
        className="d-flex justify-content-center align-items-center"
        style={{ { minHeight: "65vh" } }}>
        <div
          className="card p-4 shadow-lg border-0 rounded"
          style={{ { width: "550px", backgroundColor: "white" } }}>
          <h3 className="text-center mb-3 text-danger">Make a Payment</h3>
          <form onSubmit={handleNext} className="payment-form">
            <label className="fw-bold">Enter Amount (₹):</label>
            <input type="number" value={amount}>
            <input type="text" value="1000" style={{ width: "100px" }}>
            <label>Select Payment Method:</label>
            <select>
              <option value="Card">Card</option>
              <option value="UPI">UPI</option>
              <option value="Bank Transfer">Bank Transfer</option>
            </select>
          </form>
        </div>
      </div>
    </div>
  );
}
```

```

<option value="" disabled>
    ---SELECT PAYMENT METHOD---
</option>
<option value="card"> Credit / Debit Card</option>
<option value="upi"> UPI</option>
<option value="bank"> Bank Transfer</option>
</select>
<button type="submit" className="btn btn-danger w-100">
    Next
</button>
</form>
</div>
</div>
</div>
);
};

export default MakePayment;

```

CompletedPosts.jsx

```

import React, { useEffect, useState } from "react";
import axios from "axios";
import AdminNavbar from "./AdminNavbar";
import TransactionModal from "./TransactionModal";
const CompletedPosts = () => {
    const [completedPosts, setCompletedPosts] = useState([]);
    const [selectedPost, setSelectedPost] = useState(null);
    const [showModal, setShowModal] = useState(false);
    const [completedTransactions, setCompletedTransactions] = useState({ });
    useEffect(() => {
        fetchCompletedPosts(), [];
    }
    const fetchCompletedPosts = async () => {
        try {
            const response = await axios.get("http://localhost:3030/completedposts");
            console.log("Completed Posts Response:", response.data);
            setCompletedPosts(response.data);
        } catch (error) {
            console.error("Failed to fetch completed posts", error);
        }
    };
    const handleOpenModal = (post) => {
        setSelectedPost(post);
        setShowModal(true);
    };
    const handleTransactionSuccess = async () => {
        await fetchCompletedPosts();
        setShowModal(false);
    };
    return (

```

```

<div>
  <AdminNavbar />
  <div className="container mt-4">
    <h3 className="text-center fw-bold mb-4 text-success">TRANSACTION</h3>
    {completedPosts.length === 0 ? (
      <p className="text-center">No completed posts available.</p>
    ) : (
      <div className="row">
        {completedPosts.map((post) => (
          <div key={post._id} className="col-md-4 mb-4">
            <div className="card shadow-sm border-success">
              <div className="card-body">
                <h5 className="card-title text-success">{post.title}</h5>
                <p className="card-text">
                  <strong>Name:</strong> {post.name} </p>
                  <p><strong>Required:</strong> ₹{post.requiredAmount}</p>
                  <p> <strong>Collected:</strong> ₹{post.currentDonationsReceived}</p>
                  <p className="text-success fw-bold">✓ Fully Funded</p>
                  {post.status === "success" ? (
                    <p className="text-success fw-bold">Transaction Successful </p>
                  ) : (
                    <button className="btn btn-success" onClick={() =>
                      handleOpenModal(post)}> Perform Transaction</button>
                  )
                </div>
              </div></div>
            )})
          </div>)}
        </div>
      {selectedPost && (
        <TransactionModal show={showModal} handleClose={() => setShowModal(false)}
          post={selectedPost} onTransactionSuccess={handleTransactionSuccess}
        />
      )}
    </div>
  );
};

export default CompletedPosts;

```

GamePayments.jsx

```

import React, { useEffect, useState } from "react";
import axios from "axios";
import { Table, Container } from "react-bootstrap";
import AdminNavbar from "./AdminNavbar";
const GamePayments = () => {
  const [gamedonation, setPayments] = useState([]);
  useEffect(() => {
    fetchPayments();
  }, []);

```

```

const fetchPayments = async () => {
  try {
    const response = await axios.get("http://localhost:3030/gamepayments");
    setPayments(response.data);
  } catch (error) {
    console.error("Failed to fetch payments", error);
  }
};

return (
  <div>
    <AdminNavbar />
    <Container className="mt-4">
      <h3 className="text-center fw-bold mb-4">GAME PAYMENTS</h3>
      <Table striped bordered hover>
        <thead className="bg-primary text-white">
          <tr>
            <th>#</th>
            <th>User ID</th>
            <th>Amount (₹)</th>
            <th>Method</th>
            <th>Status</th>
            <th>Date</th>
          </tr>
        </thead>
        <tbody>
          {gamedonation.length === 0 ? (
            <tr>
              <td colSpan="6" className="text-center">
                No payments found
              </td>
            </tr>
          ) : (
            gamedonation.map((gamedonation, index) => (
              <tr key={gamedonation._id}>
                <td>{index + 1}</td>
                <td>{gamedonation.userId}</td>
                <td>₹{gamedonation.amount}</td>
                <td>{gamedonation.method}</td>
                <td className={
                  gamedonation.status === "pending"
                    ? "text-warning"
                    : "text-success"
                }>{gamedonation.status}</td>
                <td>{new Date(gamedonation.createdAt).toLocaleString()}</td>
              </tr>
            )));
          )
        </tbody>
      </Container>
    </div>
  );
};

export default GamePayments;

```

```

const Express = require("express");
const Mongoose = require("mongoose");
const Cors = require("cors");
const Bcrypt = require("bcrypt");
const jwt = require("jsonwebtoken");
const { PDFDocument, rgb, StandardFonts } = require('pdf-lib');
const nodemailer = require('nodemailer');
const fs = require('fs');
const multer = require('multer');
const path = require('path');
const SibApiV3Sdk = require('sib-api-v3-sdk');
const EventEmitter = require('events');
const paymentEvents = new EventEmitter();
const cron = require('node-cron');

const userModel = require("./models/users");
const adminModel = require("./models/admin");
const socialworkersModel = require("./models/socialworkers");
const reviewModel = require("./models/review");
const paymentModel = require("./models/payment");
const postModel = require("./models/post");
const gameDonationModel = require("./models/gamedonation");
const platformEarningModel = require('./models/platformearning');
const guessTheNumberModel = require("./models/guessthenumber");
const quizModel = require("./models/quiz");
const ticTacToeModel = require("./models/tictactoe");
const snakeGameModel = require("./models/snakegame");
const hangmanModel = require("./models/hangman");
const transactionModel = require("./models/transaction");
const walletModel = require('./models/wallet');
const rewardModel = require("./models/reward");
const announcementModel = require("./models/announcement");
const emergencyModel = require("./models/emergency");
const GameModels = [quizModel, guessTheNumberModel, ticTacToeModel, snakeGameModel, hangmanModel];
let app = Express();
app.use(Express.json());
app.use(Cors());

//Register
app.post("/register", async(req,res)=> {
    let input = req.body
    let hashedPassword = Bcrypt.hashSync(req.body.password, 10)
    req.body.password = hashedPassword
    let check = await userModel.find({email:req.body.email})
    if(check.length>0)
    {
        res.json({"status":"Email ID already exists"})
    }
    else
}

```

```
{
    let result = new userModel(input)
    await result.save()
    res.json({ "status": "Success" })
}

// Make Payment
app.post("/makepayment", async (req, res) => {
    const { postId, amount, method } = req.body;
    console.log("Received Payment Request:", req.body);
    const token = req.headers.authorization?.split(" ")[1];
    if (!token) {
        return res.status(401).json({ status: "Error", message: "Token is missing! Unauthorized access" });
    }
    const decoded = jwt.verify(token, "CharityApp");
    const userId = decoded.userId;
    if (!postId || !mongoose.Types.ObjectId.isValid(postId)) {
        console.log("Invalid Post ID:", postId);
        return res.status(400).json({ status: "Failed", message: "Invalid or missing Post ID" });
    }
    if (!amount || isNaN(amount) || amount <= 0) {
        console.log("Invalid Amount:", amount);
        return res.status(400).json({ status: "Failed", message: "Invalid or missing amount" });
    }
    if (!method) {
        console.log("Missing Payment Method:", method);
        return res.status(400).json({ status: "Failed", message: "Payment method is required" });
    }
    try {
        const post = await postModel.findById(postId);
        if (!post) {
            console.log("Post Not Found:", postId);
            return res.status(404).json({ status: "Failed", message: "Post not found" });
        }
        if (post.currentDonationsReceived + Number(amount) > post.requiredAmount) {
            return res.status(400).json({
                status: "Failed",
                message: "Donation exceeds required amount. Try a smaller amount."
            });
        }
        const payment = await paymentModel.create({
            userId,
            postId,
            amount,
            method,
            status: 'pending'
        });
    }
})
```

```

console.log("Payment Created Successfully:", payment);
return res.status(200).json({
    status: "Success",
    message: "Payment Initiated Successfully",
    paymentId: payment._id
});
} catch (error) {
    console.error("Payment Error:", error);
    return res.status(500).json({
        status: "Failed",
        message: "Payment Failed",
        error: error.message
    });
}
});

// Process Payment
app.post("/processpayment", async (req, res) => {
    const { paymentId } = req.body;
    const token = req.headers.authorization?.split(" ")[1];
    if (!token) {
        return res.status(401).json({
            status: "Error",
            message: "Token is missing! Unauthorized access"
        });
    }
    try {
        const decoded = jwt.verify(token, "CharityApp");
        const payment = await paymentModel.findById(paymentId);
        if (!payment) {
            return res.status(404).json({
                status: "Error",
                message: "Payment not found!"
            });
        }
        if (payment.status === "success") {
            return res.status(200).json({
                status: "Success",
                message: "Payment already processed!"
            });
        }
        const post = await postModel.findById(payment.postId);
        if (!post) {
            return res.status(404).json({
                status: "Error",
                message: "Post not found!"
            });
        }
        if (post.currentDonationsReceived + Number(payment.amount) > post.requiredAmount)
    {

```

```

return res.status(400).json({
  status: "Error",
  message: "Donation target already reached or this payment exceeds the required
amount."
});
}
payment.status = "success";
await payment.save();
post.currentDonationsReceived += Number(payment.amount);
if (post.currentDonationsReceived >= post.requiredAmount) {
  post.currentDonationsReceived = post.requiredAmount;
  post.status = 'approved';
}
await post.save();
const pdfDoc = await PDFDocument.create();
const page = pdfDoc.addPage([600, 400]);
const { width, height } = page.getSize();
const font = await pdfDoc.embedFont(StandardFonts.Helvetica);
const fontSize = 12;
page.drawText('Payment Receipt', {
  x: 200,
  y: height - 50,
  size: 20,
  font,
  color: rgb(0, 0, 0)
});
page.drawText(`Transaction ID: ${payment._id}`, { x: 50, y: height - 100, size: fontSize,
font });
page.drawText(`Amount Paid: INR ${payment.amount}`, { x: 50, y: height - 120, size:
fontSize, font });
page.drawText(`Payment Method: ${payment.method}`, { x: 50, y: height - 140, size:
fontSize, font });
page.drawText(`Payment Status: ${payment.status}`, { x: 50, y: height - 160, size:
fontSize, font });
page.drawText(`Date & Time: ${new Date(payment.createdAt).toLocaleString()}`, { x:
50, y: height - 180, size: fontSize, font });
const pdfBytes = await pdfDoc.save();
const pdfBuffer = Buffer.from(pdfBytes);
res.status(200).json({
  status: "Success",
  message: "Payment successfully processed! Receipt will be sent via email.",
  paymentId: payment._id
});
try {
  const apiInstance = new SibApiV3Sdk.TransactionalEmailsApi();
  const sendSmtpEmail = new SibApiV3Sdk.SendSmtpEmail();
  sendSmtpEmail.subject = "Payment Receipt - Thank You for Your Donation!";
  sendSmtpEmail.htmlContent =
    `

### Dear Donor,</h3> <p>Thank you for your generous donation of ₹${payment.amount}.</p> `


```

```

<p>Please find the attached payment receipt for your reference.</p>
<p><strong>Transaction ID:</strong> ${payment._id}</p>
<p><strong>Payment Method:</strong> ${payment.method}</p>
<p><strong>Date:</strong> ${new Date(payment.createdAt).toLocaleString()}</p>
<p>We sincerely appreciate your support.</p>
<br>
<p>Regards,</p>
<p><strong>CharityApp Team</strong></p>
`;
sendSmtpEmail.sender = { name: "CharityApp", email: "charityapp2025@gmail.com" };
sendSmtpEmail.to = [{ email: decoded.email }];

sendSmtpEmail.attachment = [
  content: pdfBuffer.toString('base64'),
  name: `Receipt_${payment._id}.pdf`
];
await apiInstance.sendTransacEmail(sendSmtpEmail);
console.log(`Email sent successfully to ${decoded.email}`);
} catch (emailError) {
  console.error("Email Sending Error:", emailError);
}
} catch (error) {
  console.error("Payment Processing Error:", error);
  return res.status(500).json({
    status: "Error",
    message: "Failed to process payment. Something went wrong."
  });
}
});

//Download Receipt
app.post("/downloadreceipt", async (req, res) => {
let { paymentId } = req.body;
let token = req.headers.authorization?.split(" ")[1];
if (!token) return res.status(401).send("Token Missing!");
try {
  const decoded = jwt.verify(token, "CharityApp");
  const user = await userModel.findOne({ email: decoded.email });
  if (!user) return res.status(404).send("User Not Found");
  const payment = await paymentModel.findById(paymentId);
  if (!payment) return res.status(404).send("Payment Not Found");
  const pdfDoc = await PDFDocument.create();
  const page = pdfDoc.addPage([600, 400]);
  const { width, height } = page.getSize();
  const font = await pdfDoc.embedFont(StandardFonts.Helvetica);
  const fontSize = 12;
  page.drawText('Payment Receipt', {
    x: 200,
  }
);
}
});
```

```

y: height - 50,
size: 20,
font,
color: rgb(0, 0, 0)
});
page.drawText(`Transaction ID: ${payment._id}`, { x: 50, y: height - 100, size: fontSize,
font });
page.drawText(`Amount Paid: INR ${payment.amount}`, { x: 50, y: height - 120, size:
fontSize, font }); // Avoid ₹
page.drawText(`Payment Method: ${payment.method}`, { x: 50, y: height - 140, size:
fontSize, font });
page.drawText(`Payment Status: ${payment.status}`, { x: 50, y: height - 160, size:
fontSize, font });
page.drawText(`Date & Time: ${new Date(payment.createdAt).toLocaleString()}`, { x:
50, y: height - 180, size: fontSize, font });
const pdfBytes = await pdfDoc.save();
res.setHeader('Content-Type', 'application/pdf');
res.setHeader('Content-Disposition', `attachment;
filename=Receipt_${payment._id}.pdf`);
res.send(Buffer.from(pdfBytes));
} catch (error) {
console.error("Failed to download receipt:", error);
res.status(500).send("Failed to download receipt");
}
});

//Transaction
app.post("/transaction", async (req, res) => {
try {
const { postId } = req.body;
const post = await postModel.findById(postId);
if (!post) {
return res.status(404).json({ error: "Post not found" });
}
const transactionData = {
postId: post._id,
requiredAmount: post.requiredAmount,
accountName: post.accountName,
accountNo: post.accountNo,
ifsc: post.ifsc,
bankName: post.bankName,
status: "success",
};
const newTransaction = new transactionModel(transactionData);
await newTransaction.save();
await postModel.findByIdAndUpdate(postId, { status: "success" });
res.status(201).json({ message: "Transaction successful", transaction: newTransaction
});
} catch (error) {
console.error("Transaction Error:", error);
res.status(500).json({ error: "Server error" });
}
});

```

6.2 SCREENSHOTS

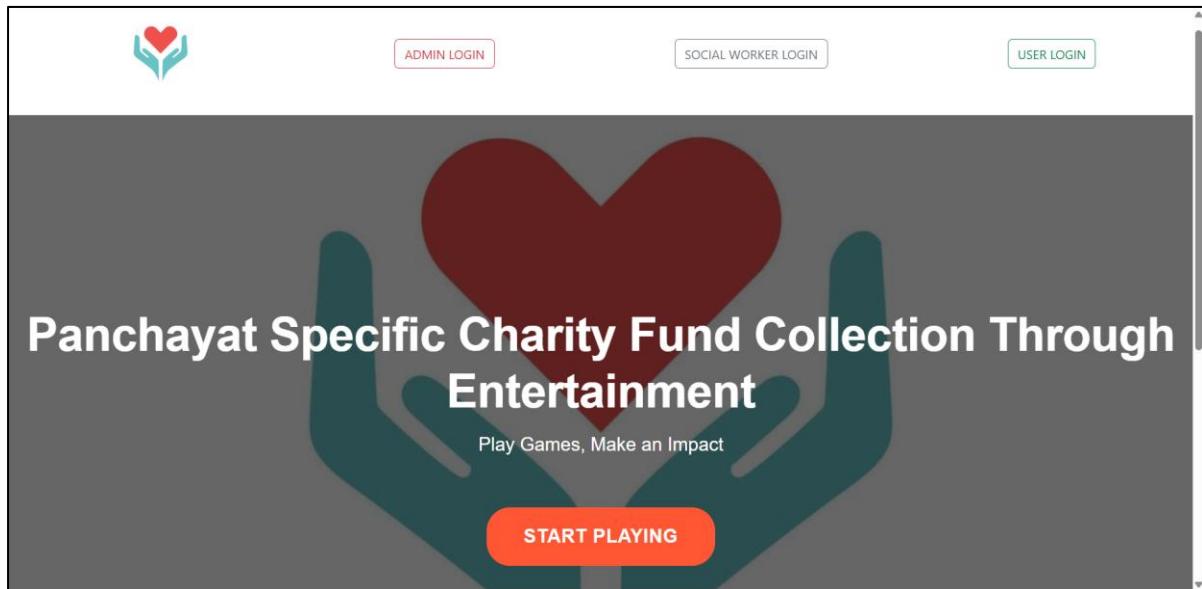


Fig 6.2: Home Page

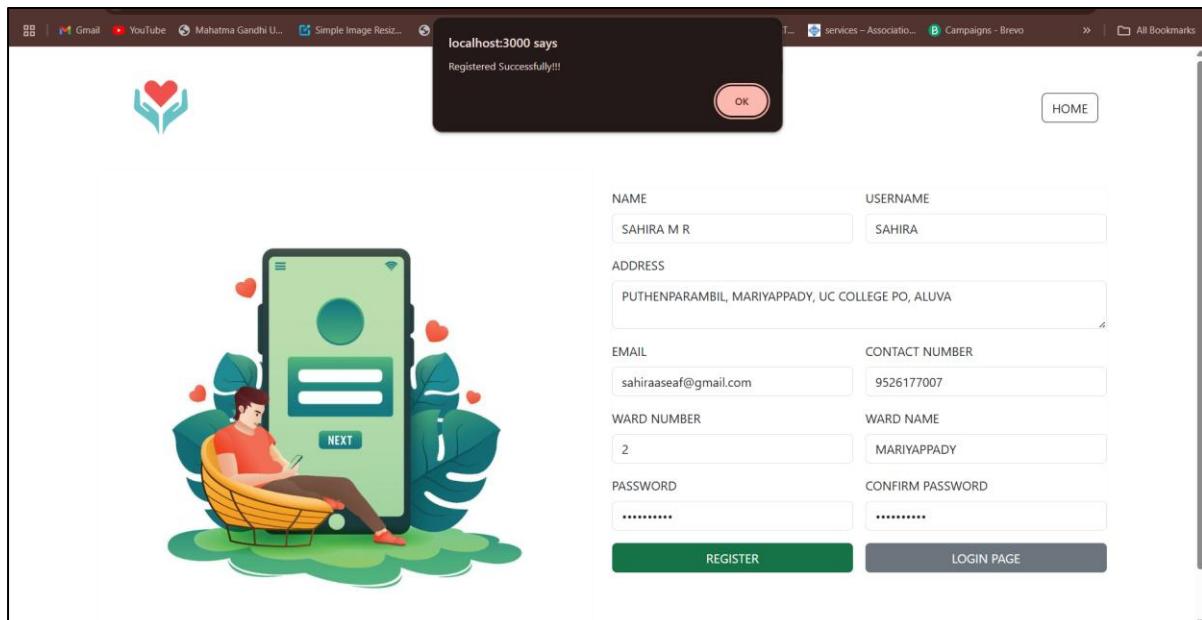


Fig 6.2: Register Page

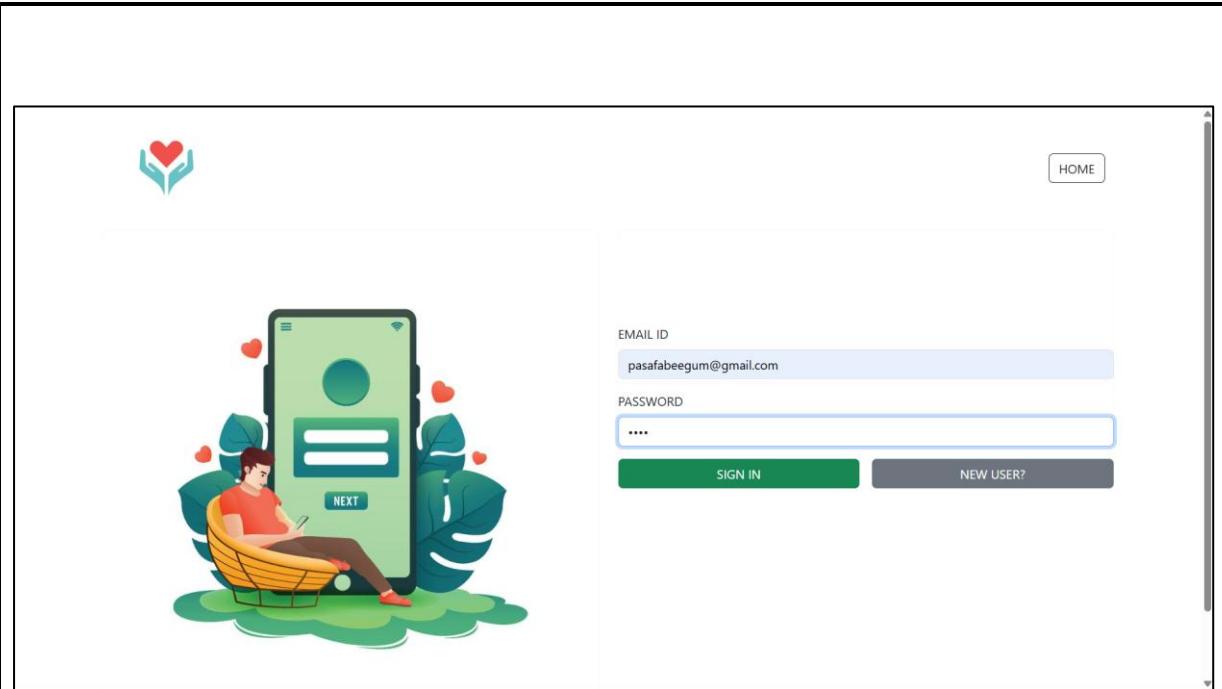


Fig 6.2: Login Page

A screenshot of a web browser showing the 'SOCIAL WORKERS DASHBOARD'. The URL is 'localhost:3000/socialworkersdashboard'. The dashboard has a sidebar on the left with links: HOME, ADD POSTS, MANAGE POSTS, VIEW REPORTS, COMMUNITY ANNOUNCEMENT, and EMERGENCY ALERT. The main area shows 'RECENT POSTS' with two entries: 'Medical Emergency for 15 year old' (Name: John Doe, Required: ₹10000, Status: completed, Collected: ₹10000.00) and 'Medical Emergency' (Name: Sam Dominic, Required: ₹50000, Status: pending, Collected: ₹0.00). A 'LOGOUT' button is at the top right.

Fig 6.2: Social Worker Dashboard

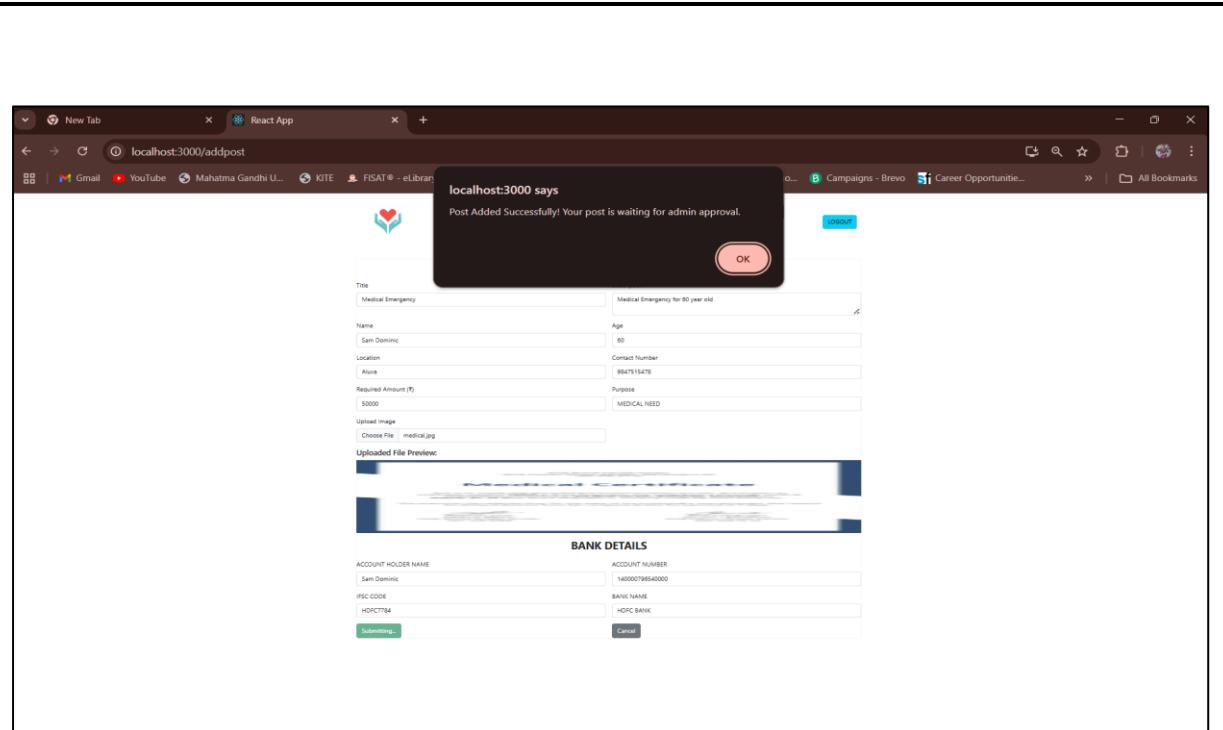


Fig 6.2: Add Post

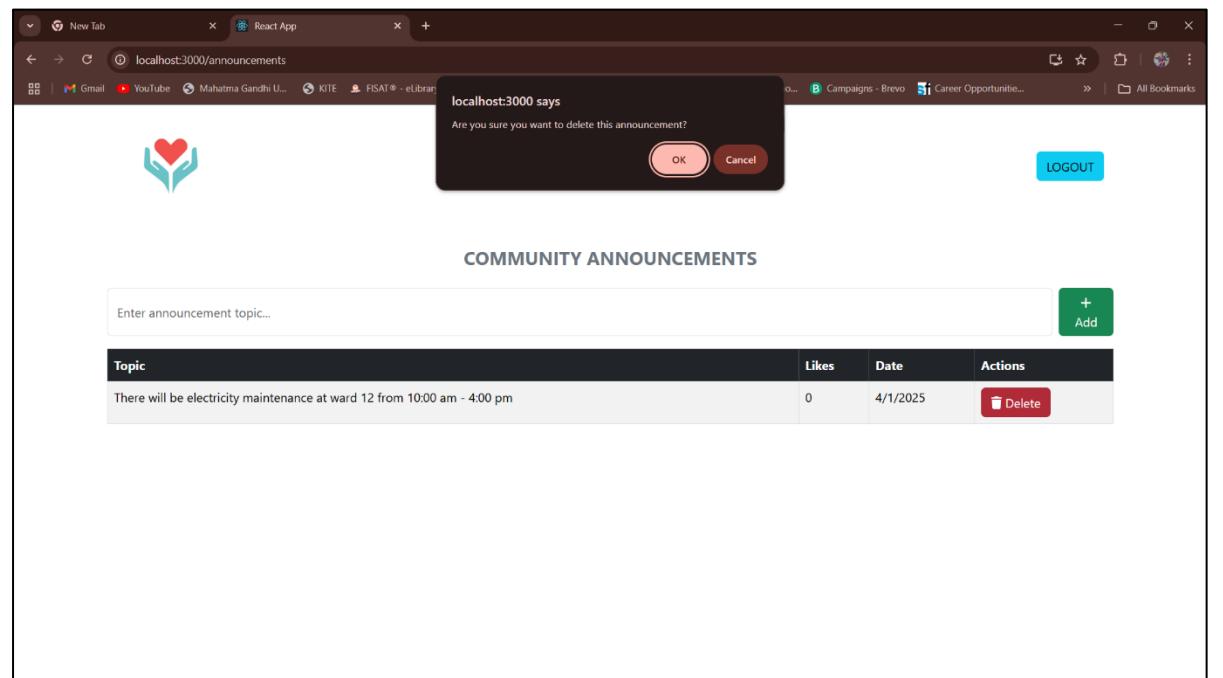


Fig 6.2: Add Community Announcement

Panchayat Specific Charity Fund Collection Through Entertainment

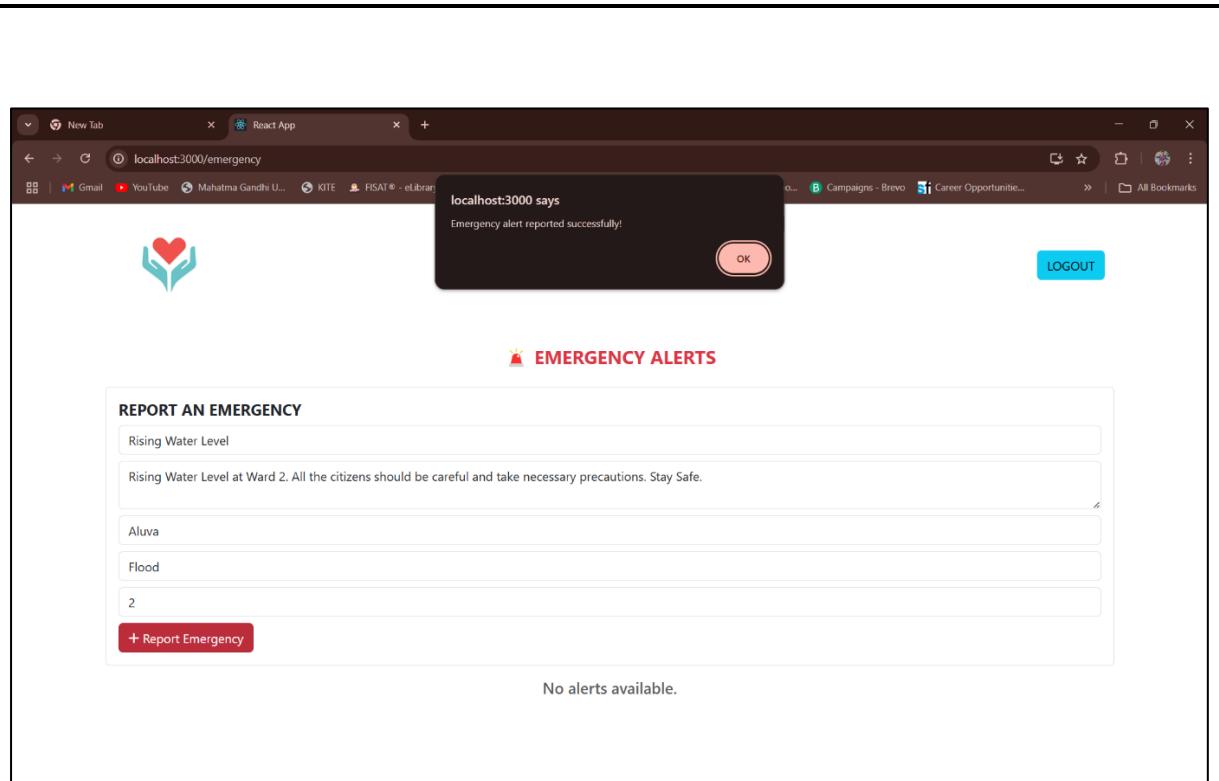


Fig 6.2: Add Emergency Alert

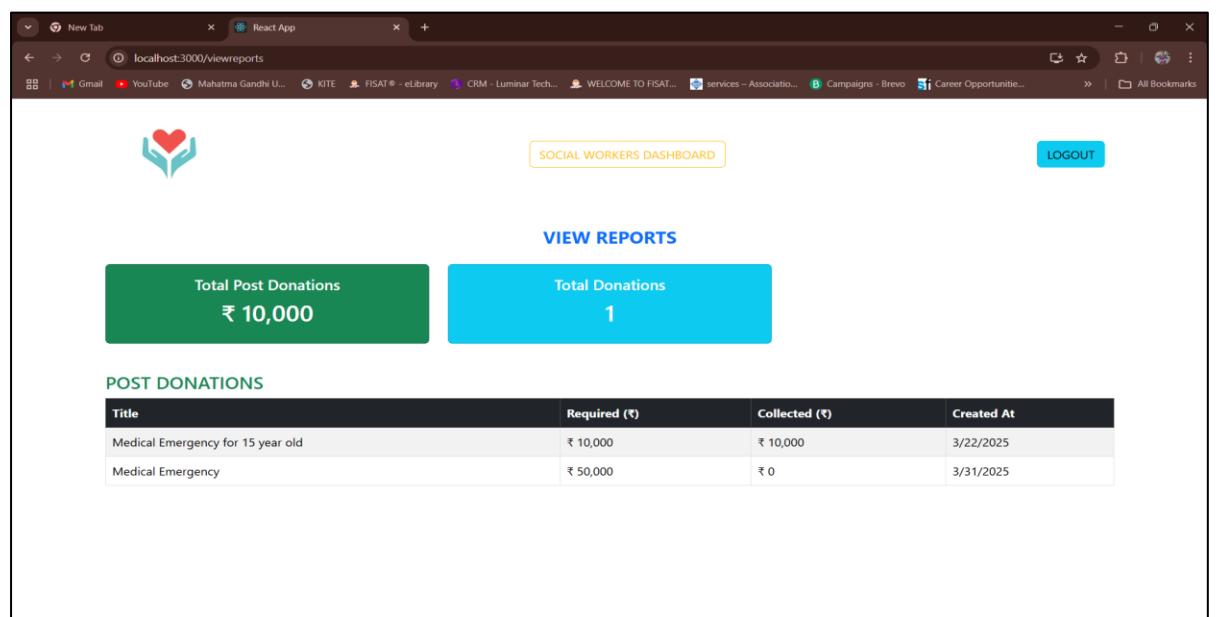


Fig 6.2: View Reports

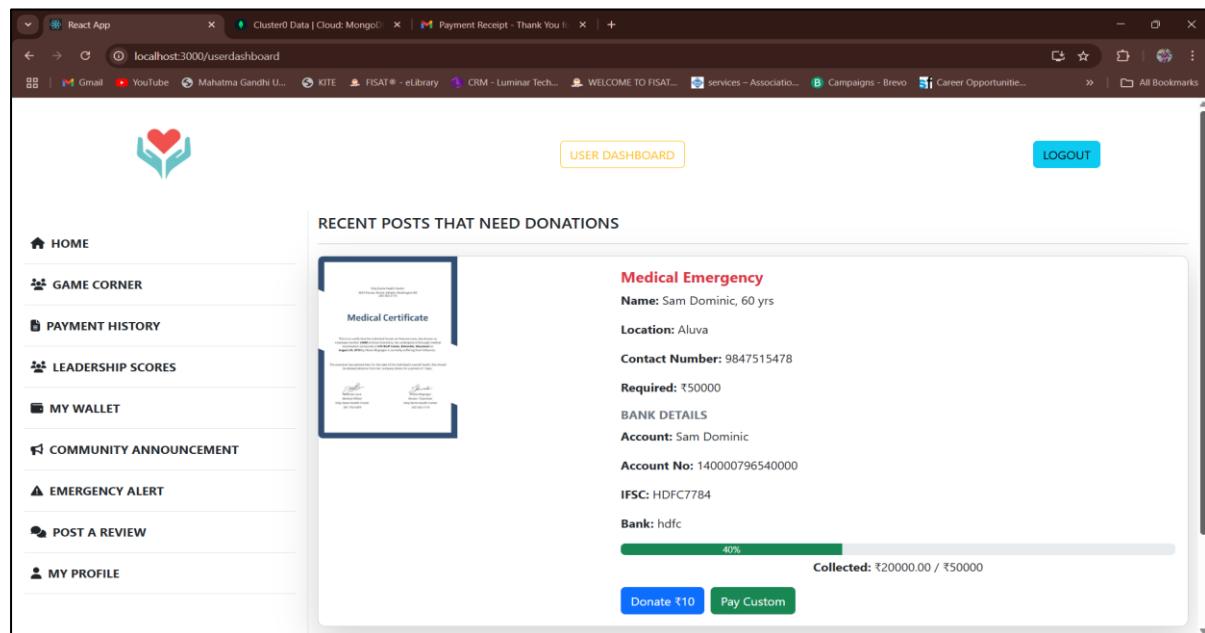


Fig 6.2: User Dashboard

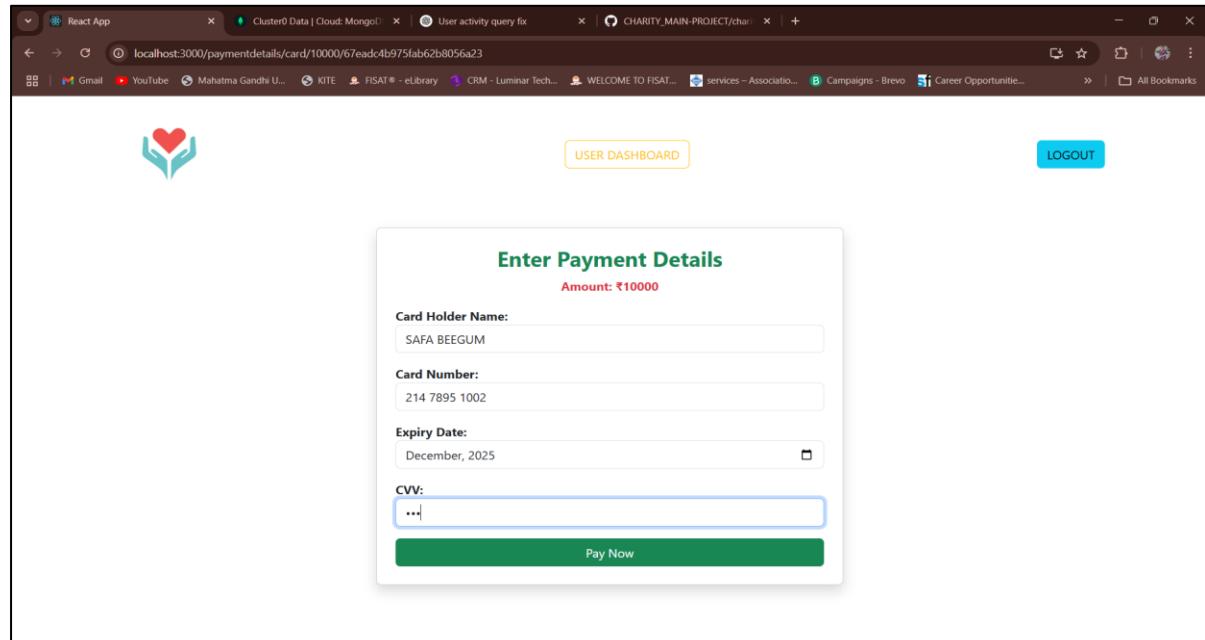


Fig 6.2: Make Custom Payment

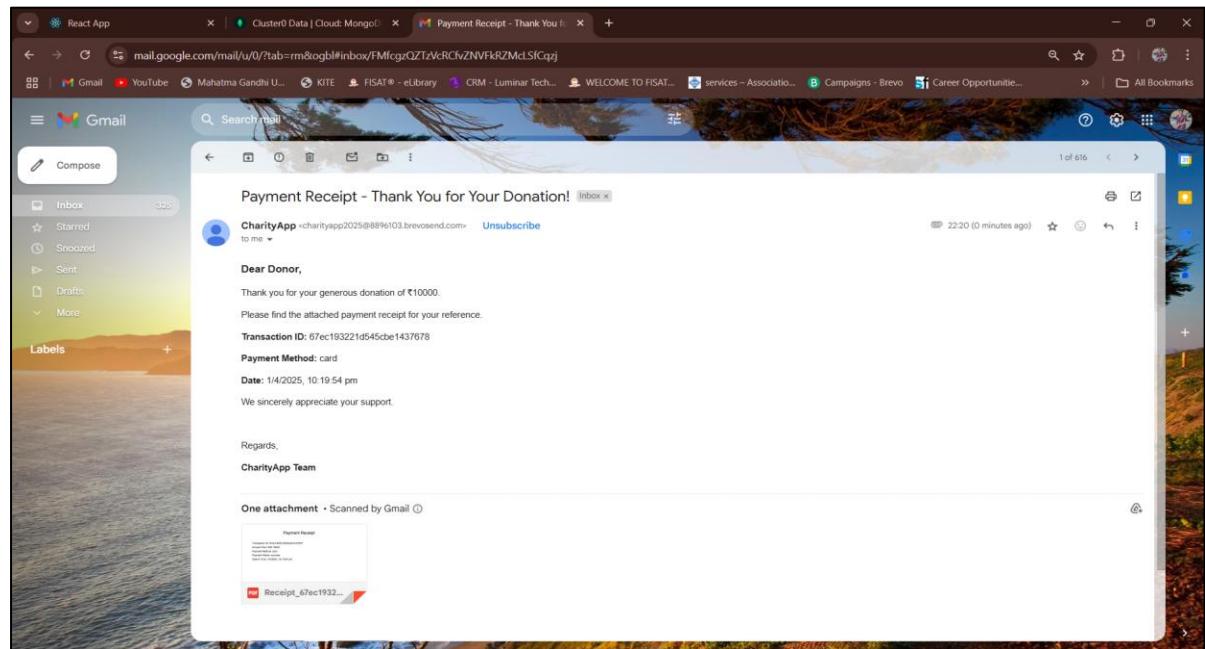


Fig 6.2: Payment Notification

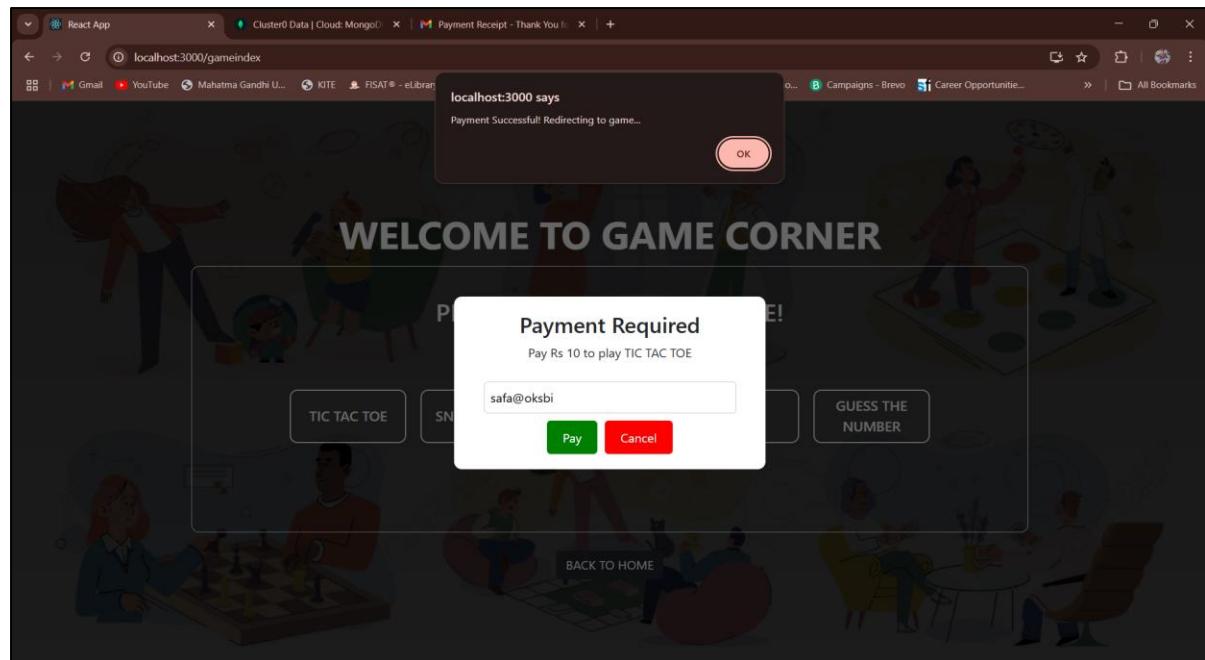


Fig 6.2: Make Game Payment

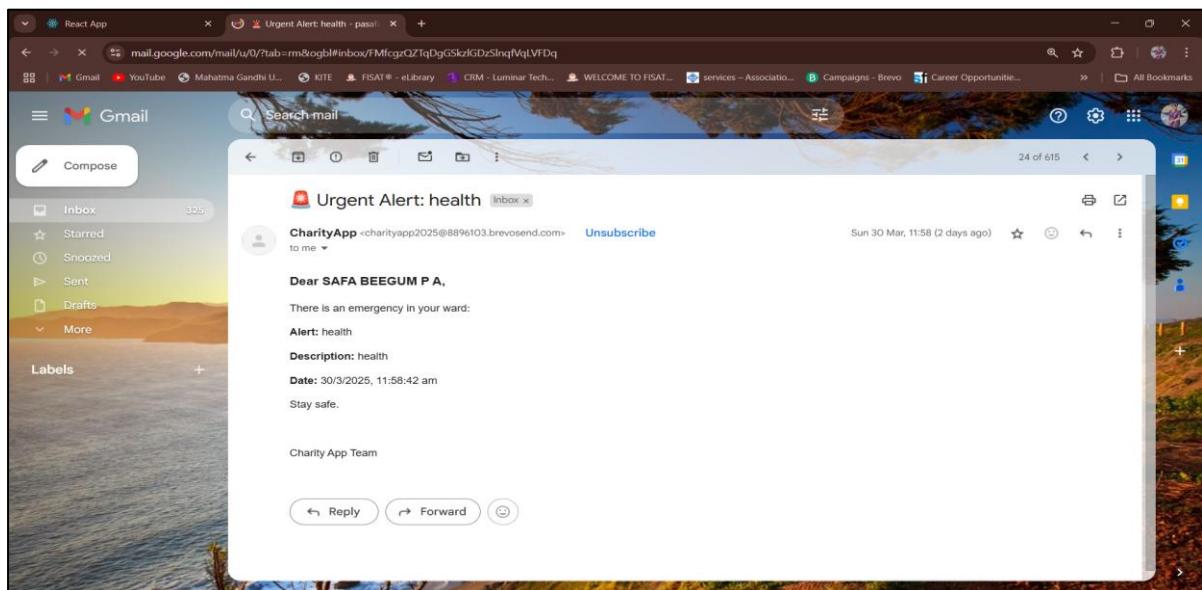


Fig 6.2: Emergency Alert Notification

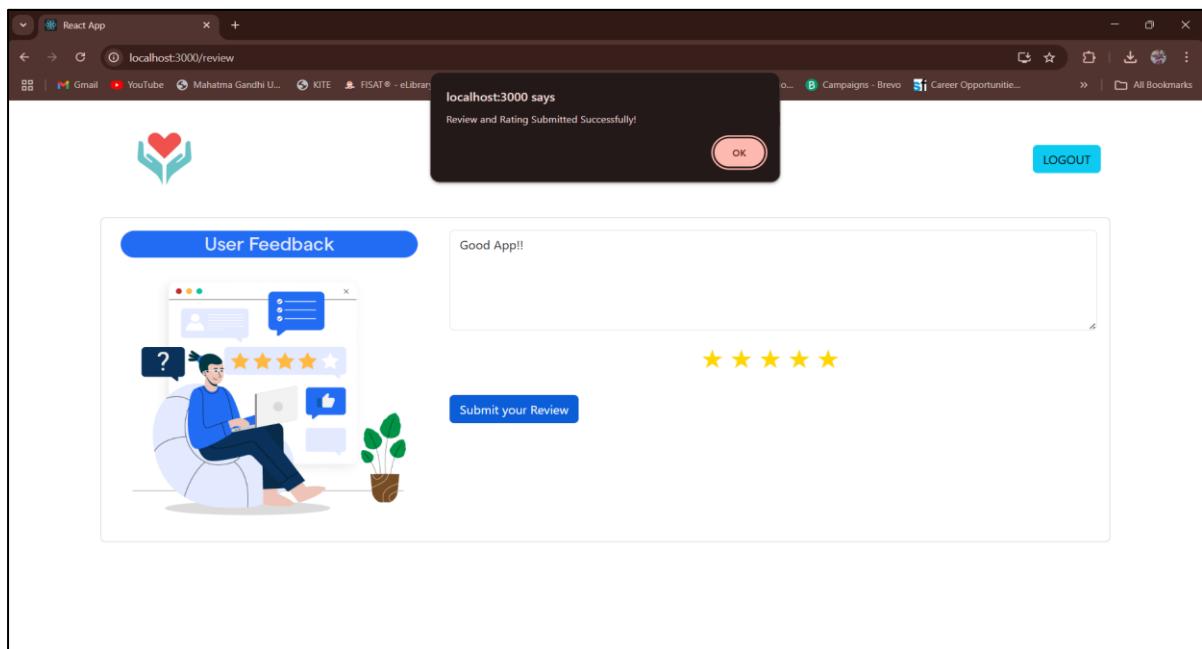


Fig 6.2: Add Feedback and Rating

Panchayat Specific Charity Fund Collection Through Entertainment

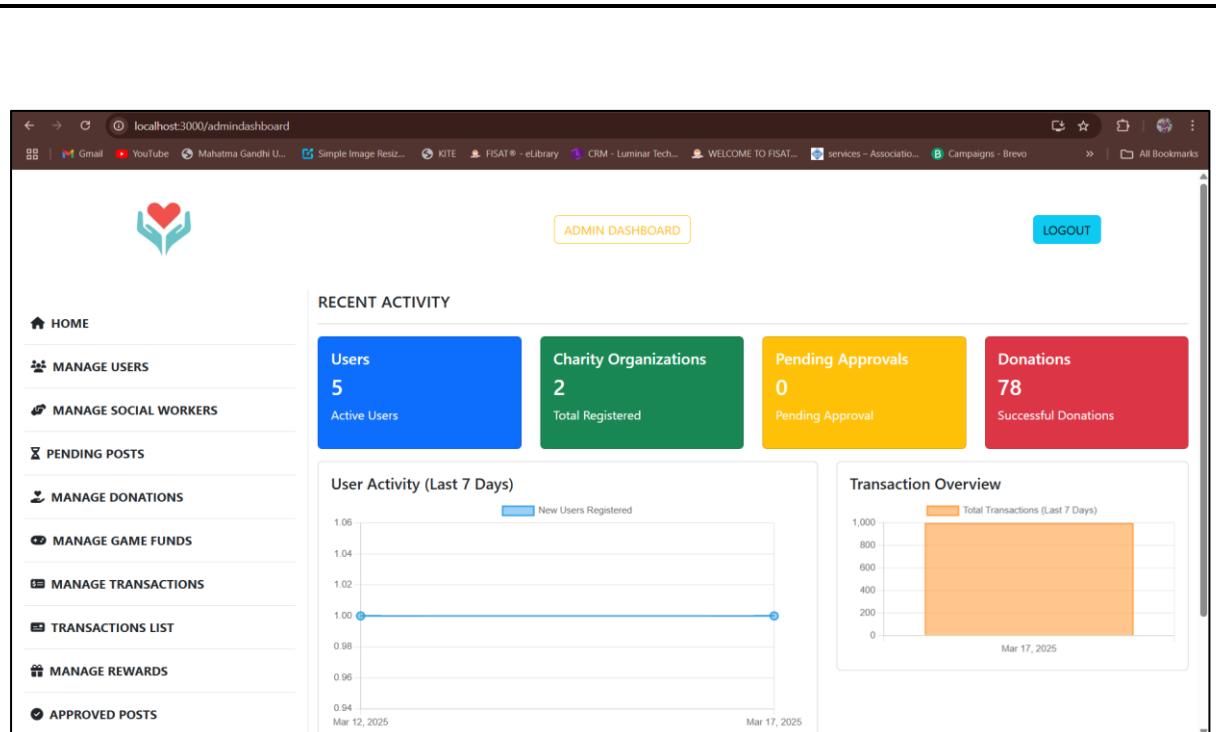


Fig 6.2: Admin Dashboard

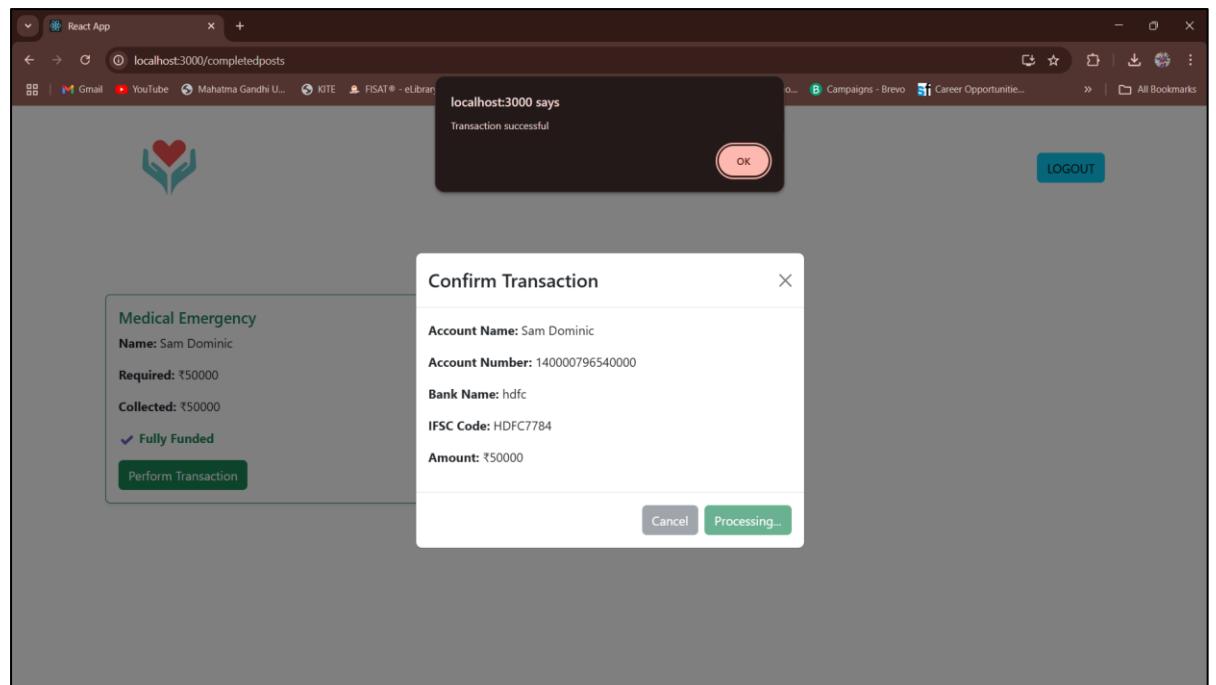


Fig 6.2: Transaction

CHAPTER 7

REFERENCES

- [1] Sommerville, I. (2015). *Software Engineering* (10th Edition). Pearson.
- [2] Pressman, R. S. (2019). *Software Engineering: A Practitioner's Approach* (9th Edition). McGraw Hill.
- [3] Rajan, V., & Thomas, R. (2022). "Gamification in Crowdfunding Platforms: Engaging Users for Social Good." *International Journal of Interactive Multimedia and Artificial Intelligence*, 7(5), 90-97.
- [4] Kaur, R., & Bansal, A. (2020). "Game-Based Fundraising: Leveraging Gamification for Non-Profit Engagement." *Journal of Social Impact Technology*, 4(3), 67-75.
- [5] Dey, A., & Ramesh, S. (2021). "Gamified Platforms for Community Development: A Case Study Approach." *Journal of Community Informatics*, 17(1), 23-39.
- [6] Singh, M., & Verma, P. (2019). "Digital Games for Social Good: Merging Play and Purpose." *Journal of Human-Centered Computing*, 11(4), 75-83.
- [7] Kumar, N., & Patel, D. (2020). "A Review on Secure Payment Gateways for Web Applications." *Journal of Web Engineering*, 19(3), 205-220.
- [8] Singh, M., & Verma, P. (2019). "Wallet-Based Reward Systems in Digital Platforms." *International Journal of Computer Trends and Technology*, 67(6), 45-49.
- [9] MongoDB Official Documentation - <https://www.mongodb.com/docs/>
- [10] React.js Official Documentation - <https://react.dev/>
- [11] Node.js Official Documentation - <https://nodejs.org/en/docs>
- [12] Express.js Guide - <https://expressjs.com/>



Department of Computer Applications

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)[®]

Hormis Nagar, Mookkannor, Angamaly, Ernakulam - 683577

Phone : 0484 2725272, Email : mail@fisat.ac.in