Cash Crafter

A PROJECT REPORT
for
Mini Project- I (K24MCA18P)
Session (2024-25)

Submitted by
Devanshi Singhal
202410116100060
Aanchal
202410116100002

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Ms. Divya Singhal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS

KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(DECEMBER- 2024)

CERTIFICATE

Certified that **Devanshi Singhal (202410116100002), Aanchal (202410116100002), has**/ have carried out the project work having "**Cash Crafter**" (**Mini Project-I, K24MCA18P**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms. Divya Singhal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Kr. Tripathi
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Cash Crafter
Devanshi Singhal, Aanchal

ABSTRACT

In the digital age, effective financial management is crucial for individuals and small businesses alike. Our project, "Cash Crafter," aims to provide a comprehensive financial management platform that simplifies tracking income, expenses, and overall financial health. Inspired by the success of platforms like Khatabook, Cash Crafter is designed to empower users with intuitive tools for managing their finances at their fingertips.
It allows users to register and create personalized accounts, where they can input essential financial details. The platform supports various categories for income and expenses, enabling users to categorize their transactions for better clarity and analysis. Users can easily log their daily financial activities, set reminders for upcoming payments, and generate insightful reports that reflect their financial status over time.
It is a user-friendly financial management website that aims to simplify the complexities of personal finance. By providing essential tools and resources, we empower users to take control of their financial well-being, fostering a culture of financial literacy and responsibility.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Devanshi Singhal

Aanchal

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The rise of digital transformation has reshaped how businesses and individuals manage their financial operations. In today's fast-paced economy, manual transaction management methods, such as handwritten ledgers or Excel sheets, have become outdated and inefficient. **Cash Crafter** addresses these challenges by providing an innovative, easy-to-use web-based application for recording, analyzing, and managing financial transactions.

Cash Crafter is specifically designed for small businesses, freelancers, and financial agents who need a streamlined solution for transaction management. The application combines features like automated notifications, financial insights through dynamic graphs, and secure record-keeping, ensuring a comprehensive and user-friendly experience.

Its core functionalities—transaction recording, client communication, and financial reporting—are built to save time, minimize human error, and enhance decision-making. With customizable themes, mobile responsiveness, and real-time updates, Cash Crafter prioritizes user experience while providing valuable tools for financial management.

1.2 PURPOSE OF THE PROJECT

The primary purpose of **Cash Crafter** is to simplify and automate the process of transaction management for users who rely on manual or semi-automated systems. This project ensures that users can easily store, access, and analyze their financial records, enabling efficient decision-making and reducing operational overhead.

Key goals include:

- **Centralized Record Management**: Providing a single platform to record all transactions securely.
- **Automation**: Automating repetitive tasks like generating reports and notifying clients.
- **User Empowerment**: Offering users actionable insights into their financial data for better decision-making.
- **Accessibility**: Ensuring the application is accessible from anywhere, on any device.

By meeting these objectives, Cash Crafter bridges the gap between manual bookkeeping methods and expensive enterprise-level solutions, offering an affordable yet powerful alternative.

1.3 BACKGROUND

In traditional financial management, small business owners and freelancers have relied on physical ledgers, spreadsheets, or simple accounting software. While effective for basic needs, these methods often lack features like automation, data visualization, and real-time notifications.

For instance:

1. A **freelancer** may forget to notify a client of a pending payment, leading to delayed transactions.
2. A **financial agent** might struggle to consolidate monthly reports manually, wasting time on repetitive calculations.
3. A **small business owner** may face challenges in tracking and categorizing revenue sources.

Cash Crafter addresses these challenges by:

- **Digitizing Transaction Records**: Users can create, edit, and retrieve financial data anytime, anywhere.
- **Automating Notifications**: The system sends payment alerts to clients automatically, ensuring timely communication.
- **Providing Insights**: Cash Crafter generates dynamic reports to visualize revenue trends, pending payments, and monthly progress.

The application's user-friendly interface ensures accessibility for users with minimal technical expertise, while its scalable backend architecture makes it suitable for growing businesses.

1.4 PROBLEM STATEMENT

Manual transaction management methods, while widely used, present several challenges:

1. **Prone to Errors**: Human error in entering, calculating, or retrieving financial data can lead to inaccurate records and financial losses.
2. **Time-Consuming**: Sorting and managing transactions manually is tedious, especially for businesses dealing with high transaction volumes.
3. **Lack of Automation**: Manual systems require users to remember tasks like sending payment reminders to clients, leading to missed opportunities or delayed payments.
4. **Limited Insights**: Without tools for data visualization, users struggle to analyze trends or identify potential bottlenecks in their financial workflows.

These limitations highlight the need for an efficient, automated system like **Cash Crafter**, which combines the power of technology with user-centric design to overcome these challenges. By automating repetitive tasks, providing dynamic reports, and ensuring secure record-keeping, Cash Crafter enables users to focus on their core responsibilities rather than administrative overhead.

1.5 SIGNIFICANCE OF THE PROJECT

The significance of **Cash Crafter** lies in its ability to enhance financial management for small businesses, freelancers, and agents through automation and insights. Its value can be summarized in the following points:

1. **Enhanced Productivity**
    a. Automating notifications, report generation, and data storage saves significant time, allowing users to focus on strategic tasks rather than administrative ones.
2. **Error Reduction**
    a. By reducing reliance on manual entry, the system minimizes the risk of errors that could otherwise disrupt financial workflows or lead to inaccurate reports.
3. **Improved Decision-Making**
    a. Real-time data analysis and visual insights empower users to make informed decisions regarding their finances, such as identifying periods of high revenue or overdue payments.
4. **Client Transparency**
    a. Automated notifications and receipts improve client trust and satisfaction by ensuring clear and timely communication.
5. **Cost-Effective Solution**
    a. Unlike expensive ERP systems, Cash Crafter offers a cost-effective alternative, making advanced transaction management tools accessible to smaller businesses and freelancers.

The project's ability to integrate with SMS/Email APIs further enhances its value, ensuring smooth communication and adaptability for a variety of user needs.

1.6 SCOPE OF THE PROJECT

The scope of **Cash Crafter** includes core functionalities such as:

1. **Transaction Management**
    a. Users can add, view, edit, and delete transactions.
    b. Data is stored securely in a relational database, ensuring persistence and easy retrieval.

2. **Client Notifications**
    a. Automated SMS and Email notifications alert clients when payments are received or due.
    b. Customizable messages enhance communication efficiency.
3. **Financial Insights and Reports**
    a. Users can access dynamic reports, including revenue breakdowns, pending payments, and monthly trends.
    b. Data visualization tools like charts and graphs make reports easy to understand.
4. **User Management and Customization**
    a. Users can manage their profiles, update details, and customize themes (dark, light, or default).
5. **Future Expansion**
    a. Potential for additional features like multi-user access, PDF/Excel export of reports, and integration with accounting software.

1.7 TARGET AUDIENCE

Cash Crafter is tailored for:

1. **Small Businesses**: Business owners seeking a simplified transaction management system without investing in expensive ERP tools.
2. **Freelancers**: Individuals who need to track payments and expenses while ensuring transparency with their clients.
3. **Financial Agents**: Professionals managing multiple clients' transactions and requiring automated tools to simplify their workflows.

1.8 BENEFITS OF CASH CRAFTER

1. **Simplified Workflows**
    a. Streamlines transaction management and reduces administrative burdens.
2. **Scalable Architecture**
    a. The system is designed to grow with users' needs, accommodating more transactions and features as required.
3. **User-Friendly Interface**
    a. A responsive design ensures ease of use on desktops, tablets, and mobile devices.
4. **Real-Time Features**
    a. Instant notifications and real-time financial updates keep users informed and in control.

CHAPTER 2

FEASIBILITY STUDY/LITERATURE REVIEW

A feasibility study evaluates whether a proposed project is viable, achievable, and sustainable across various dimensions. For **Cash Crafter**, this involves analyzing technical capabilities, financial implications, operational readiness, and the societal impact of the system. This section also includes a literature review of existing solutions to understand gaps and opportunities.

2.1 TECHNICAL FEASIBILITY

Cash Crafter is built on modern, scalable technologies that ensure its functionality and efficiency. The system's technical feasibility involves evaluating the hardware, software, and tools required for development and deployment.

1. **Frontend Development**

2. The user interface will be created using HTML, CSS, Bootstrap, and JavaScript. These technologies are lightweight, responsive, and compatible with most modern web browsers.

    a. **Advantages**: Quick loading times, mobile-friendly design, and minimal dependencies.
    b. **Scalability**: The frontend can be expanded to include additional pages or features as the application evolves.

3. **Backend Development**

    The backend system will use **Node.js** or **Flask (Python)** for server-side operations. Both technologies are well-suited for handling database interactions, processing user requests, and integrating APIs for notifications.

    a. **Node.js**: Non-blocking I/O makes it efficient for handling multiple concurrent requests, such as transaction updates and notifications.
    b. **Flask**: Lightweight and flexible for web development, ensuring quick prototyping and deployment.

4. **Database**
    a. **MySQL**: Relational database management system (RDBMS) is ideal for storing structured data such as user details, transaction records, and notification logs.

b. **Scalability**: MySQL supports large datasets, making it suitable for growing businesses.

5. **Notification APIs**

Integrating **Twilio** (for SMS notifications) and **SendGrid** (for email notifications) ensures reliable client communication. These APIs are widely used, secure, and offer extensive documentation for developers.

6. **Hosting and Accessibility**

The application will be hosted on platforms like AWS or Heroku, ensuring high uptime and accessibility across devices.

a. **Cloud Hosting**: Offers cost-effective and scalable hosting options for startups and small businesses.

2.2 ECONOMIC FEASIBILITY

Economic feasibility evaluates the cost-benefit analysis of implementing the project. For **Cash Crafter**, the primary goal is to deliver a cost-effective solution without compromising on quality or performance.

1. **Development Costs**
   a. **Hardware**: Development can be done on standard hardware, such as personal computers with 4GB+ RAM.
   b. **Software**: Open-source tools (e.g., Node.js, Flask, MySQL, Bootstrap) significantly reduce software costs.
   c. **APIs**: Pay-as-you-go APIs (e.g., Twilio, SendGrid) require minimal upfront investment.
2. **Operational Costs**
   a. Cloud hosting services like AWS or Heroku offer affordable options for deploying the application, with costs proportional to usage.
   b. **Example**: Heroku's free tier is suitable during development, with scaling options starting at $7/month.
3. **Return on Investment (ROI)**
   a. Increased efficiency in transaction management reduces labor hours spent on manual tasks, saving businesses money.
   b. Automated notifications improve client satisfaction, potentially leading to higher client retention and revenue.
4. **Revenue Opportunities**
   a. A freemium model can be implemented, offering basic features for free and charging a subscription fee for premium features like advanced reporting and multi-user access.

2.3 OPERATIONAL FEASIBILITY

Operational feasibility examines whether the system can be effectively implemented and used by its intended audience.

1. **Target Audience**
    a. **Small Businesses**: Owners who need a simple, affordable tool for managing transactions.
    b. **Freelancers**: Professionals requiring client payment tracking and automated communication.
    c. **Financial Agents**: Agents handling multiple clients and needing secure, centralized records.
2. **Ease of Use**
    a. A user-friendly interface ensures that individuals with limited technical expertise can use the system effortlessly.
    b. Features like guided workflows and intuitive navigation enhance usability.
3. **System Maintenance**
    a. Routine database backups ensure data integrity and recovery.
    b. Regular updates (e.g., bug fixes, new features) can be easily deployed using cloud-based infrastructure.
4. **Training Requirements**
    a. Minimal training is needed due to the system's straightforward design. Video tutorials and help guides can assist users in understanding core functionalities.

2.4 SOCIAL FEASIBILITY

Social feasibility assesses the system's acceptance and impact on its users and society.

1. **User Impact**
    a. By automating repetitive tasks, the application frees up time for users to focus on strategic goals, enhancing their productivity and satisfaction.
    b. Clear and timely communication builds trust between users and their clients.
2. **Environmental Impact**
    a. Digital record-keeping reduces the reliance on paper-based systems, contributing to sustainability efforts.
3. **Social Adoption**
    a. The system's affordability and ease of use make it accessible to a wide audience, promoting financial literacy and efficiency across small-scale industries.
4. **Long-Term Benefits**

a. As more users adopt **Cash Crafter**, it could foster a community of small business owners and freelancers sharing best practices in financial management.

2.5 LITERATURE REVIEW

A literature review evaluates existing systems and highlights gaps that **Cash Crafter** addresses:

1. **Existing Solutions**
    a. **Manual Methods**: While commonly used, physical ledgers and spreadsheets are prone to errors and lack automation.
    b. **ERP Systems**: Tools like QuickBooks and Tally offer robust features but are often expensive and complex, making them unsuitable for small businesses.
2. **Gaps Identified**
    a. Lack of affordable, user-friendly solutions for small-scale users.
    b. Absence of automated client notifications in traditional systems.
    c. Limited customization and reporting features in existing tools.
3. **Research Studies**

Studies indicate that automating financial workflows can reduce errors by up to 60% and increase operational efficiency by 40%. Moreover, timely notifications enhance client satisfaction by 30%.

4. **Proposed Solution**

Cash Crafter bridges the gap by providing an affordable, automated system tailored to small businesses and freelancers. It integrates notifications, insights, and user-friendly design to address the limitations of existing solutions.

CHAPTER 3

PROJECT OBJECTIVE

The objectives of the Cash Crafter project are designed to provide a clear framework for the development and implementation of the financial management platform. These objectives outline the goals that the project aims to achieve, ensuring that the final product meets the needs of its target audience—freelancers, small businesses, and agents. The objectives can be categorized into primary and secondary objectives.

3.1 PRIMARY OBJECTIVES

1. Develop a User-Friendly Financial Management Platform

- Goal: To create an intuitive and accessible web-based application that simplifies financial management for users with varying levels of financial literacy.
- Details: The platform will feature a clean and responsive design, ensuring ease of navigation and usability across different devices. User experience (UX) research will be conducted to gather feedback and make iterative improvements.

2. Provide Comprehensive Transaction Management

- Goal: To enable users to efficiently track and manage their income and expenses in real-time.
- Details: The platform will allow users to categorize transactions, log recurring payments, and generate reports on their financial activities. This feature will help users maintain accurate financial records and gain insights into their spending habits.

3. Implement Budgeting and Goalsetting tools

- Goal: To empower users to set financial goals and create budgets that align with their financial objectives.
- Details: Users will be able to allocate budgets for different categories, monitor their spending against these budgets, and receive alerts when they approach their limits. This feature will promote financial discipline and help users achieve their financial goals.

4. Offer Real-Time Reporting and Analytics

- Goal: To provide users with actionable insights into their financial health through detailed reporting and analytics.
- Details: The platform will generate customizable reports that summarize income, expenses, and budget performance. Visualizations such as charts and graphs will enhance users' understanding of their financial trends over time.

5. Ensure Data Security and Privacy

- Goal: To implement robust security measures that protect user data and ensure confidentiality.
- Details: The platform will utilize encryption, secure access protocols, and options for two-factor authentication to safeguard sensitive financial information. Users will be informed about data privacy policies and practices.

3.2 SECONDARY OBJECTIVES

1. Foster Community Engagement and Support

- Goal: To create a supportive community where users can share experiences, tips, and strategies for effective financial management.
- Details: The platform will include a forum for discussion board where users can interact, ask questions, and provide feedback. This feature will encourage collaboration and knowledge sharing among users.

2. Facilitate Automated Notifications and Reminders

- Goal: To implement a notification system that keeps users informed about important financial events.
- Details: Users will receive automated reminders for upcoming payments, budget limits, and financial goals. This feature will help users stay organized and proactive in managing their finances.

3. Explore Future Enhancements and Integrations

- Goal: To plan for future scalability and enhancements that will add value to the platform.
- Details: The project will consider potential integrations with banking services, mobile application development, and advanced analytics features. User feedback will guide the prioritization of future enhancements.

4. Conduct Market Research and User Testing

- Goal: To gather insights into user needs and preferences through market research and user testing.
- Details: Surveys, interviews, and usability testing will be conducted to understand the challenges faced by the target audience. This research will inform the design and functionality of the platform, ensuring it meets user expectations.

5. Promote Financial Literacy and Responsibility

- Goal: To contribute to the financial literacy of users by providing educational resources and tools.
- Details: The platform will include articles, tutorials, and tips on financial management best practices. By promoting financial literacy, Cash Crafter aims to empower users to make informed financial decisions.

CHAPTER 4

HARDWARE AND SOFTWARE REQUIREMENTS

For Desktops/Laptops:

- **Computer to access the application by the user:**
    - Processor: Intel Core i3 or equivalent (recommended: i5 or better)
    - RAM: Minimum 4 GB (recommended 8 GB or more)
    - Storage: SSD preferred for faster performance (minimum 256 GB).
    - Operating System: Windows 10 or higher, macOS, or a suitable Linux distribution.

For Server:

- **Server Specifications**:
    - Processor: Multi-core processor (Intel Xeon or equivalent)
    - RAM: Minimum 8 GB (recommended 16 GB or more)
    - Storage: RAID setup preferred for redundancy (minimum 500 GB SSD).
    - Network: High-speed internet connection with good bandwidth.
    - Backup Solutions: External drives or cloud-based solutions for data backup.

Backend Technologies:

- Server-side languages: Node.js, Python, PHP.
- Frameworks: Express.js.

Database:

- MySQL for data management.

Frontend Technologies:

- HTML, CSS, Bootstrap, JavaScript.
- Frameworks: React

Browser Support:

- Chrome, Firefox, Safari, and Edge (latest versions).

CHAPTER 5

PROJECT FLOW

The application will follow a structured workflow, ensuring the key functionalities are seamlessly integrated:

1. **User Authentication:** Sign Up, Login, and Logout.
2. **Transaction Management:** Record and manage transactions.
3. **Dashboard & Reports:** Analyze data through graphs and insights.
4. **Notifications:** Automated alerts for clients.
5. **Settings & Preferences:** Manage user profile, themes, and preferences.

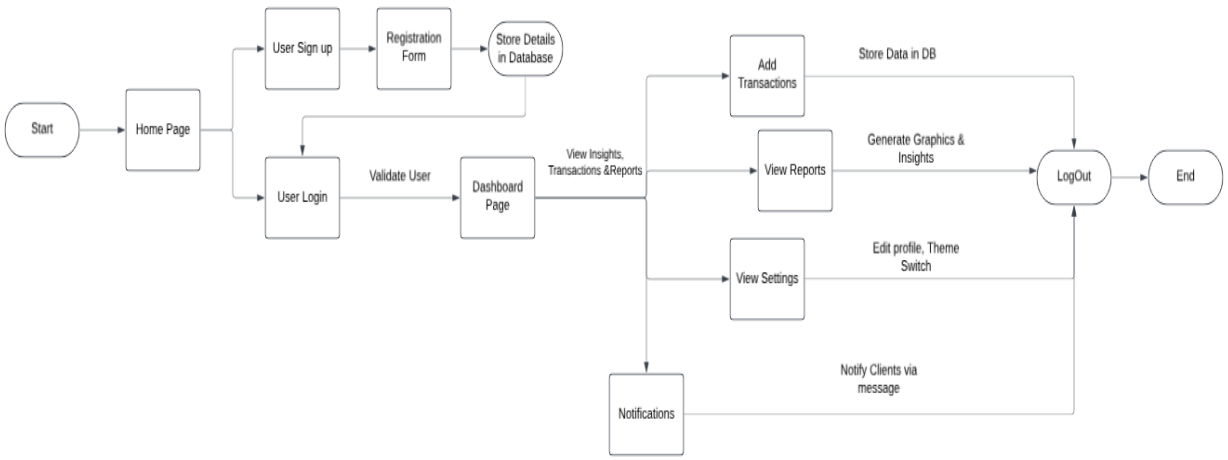

Fig 5.1: This figure represents the work flow of Cash Crafter.

Detailed User Workflow as shown in the Fig 5.1:

a. User Authentication Flow

- **Home Page**
  Navbar: Options → Login, Sign Up, About Us, Contact Us.
  Feature Highlights: "Transaction Tracking", "Client Communication", etc.
  CTA Buttons: "Get Started".

- **Sign Up Process**
  Collect user data: Name, Email, Password, Phone Number, Address (Flat No, Street, City).
  Validate input and store it in the database.
  Redirect to **Dashboard** upon successful signup.
- **Login Process**
  User enters email and password.
  Backend validates credentials against the database.
  Redirect to **Dashboard**.
- **Logout**
  User clicks Logout from the navbar dropdown.
  Session is destroyed, and user is redirected to the **Home Page**.

b. Dashboard Flow

- **Dashboard Page**
  Greeting: "Welcome, [Username]!"
- **Key Cards**:
  Total Transactions
  Total Amount Collected
  Monthly Revenue
  Navbar options → Transactions | Reports | Settings | Logout.
- **Navigation Flow**
- **Transactions Page**
  Input fields to add client transactions: Name, Amount, Date, Payment Type, Payment Mode, and Month.
  A table displays added transactions with sorting and search features.
  Transaction data is saved to the database.
- **Reports Page**
  Graphical representation of:
  Total Amount vs Pending Amount.
  Monthly Payment Insights.
  Data is fetched dynamically from the database.
- **Settings Page**

User Profile: Update name, email, mobile number, etc.
Theme Settings: Options for Dark, Light, and Default modes.
Registration Details: Option to add more information.

c. Transaction Management Flow

- **Adding Transactions**
  Fields to enter:
  Client Name
  Amount Collected

Date of Payment
Payment Type (Credit/Debit)
Payment Mode (Cash/Online)
Payment Month

- **Viewing Transactions**
  Table: Displays saved transaction records.
  Option to sort, search, or filter transactions.
  Data persistence ensured in the database.
- **Automated Notifications**
  When a transaction is added, a notification is triggered:
  SMS or Email sent to the client, confirming receipt of payment.

d. Reports and Analysis Flow

- **Data Visualization**
  Dynamic graphs displaying:
  Total Amount vs Pending Amount.
  Monthly and Yearly Payment Breakdown.
  Data fetched in real-time from the database.
- **Insights**
  Insights on revenue, pending dues, and total transactions.

e. Settings & Preferences Flow

- **User Profile Management**
  Update profile information (name, email, address, password).
- **Theme Preferences**
  Dark Mode, Light Mode, or Default Mode.
  Save user preference locally or in the database.
- **Additional Registration**

Option to add further details about the agent or client.

f. Key Backend Functionalities

- **Database Setup:** Store user credentials, transactions, and settings.
- **APIs:** Handle user authentication, data submission, and fetching transactions dynamically.
- **Notifications:** Integrate SMS or Email APIs to alert clients.
- **Data Validation:** Ensure accurate and secure input handling.

g. Tools & Technologies

- **Frontend:** HTML, CSS, Bootstrap, JavaScript.
- **Backend:** Node.js with Express, or Python with Flask/Django.
- **Database:** MySQL, MongoDB, or PostgreSQL.
- **APIs:** Twilio (SMS), SendGrid (Email).
- **Libraries:** Chart.js or Apex Charts for data visualization.

h. Future Enhancements

- Add **export options** to download reports as PDF or Excel.
- Include advanced **user authentication** (OTP, multi-factor authentication).
- Build a **mobile-friendly version** or a mobile app.
- Add **role-based access control** for different users (e.g., Admin vs Agent)

Furthermore, the project flow can be structured using requirements shown in Fig. 5.2 below. The figure shows the details of entities and their relationships among them.

Entities represent the main objects or concepts in the system. For **Cash Crafter**, the entities are:

1. **User:** Represents the system users (admin, agents, or business owners).
2. **Transactions:** Represents financial transactions recorded in the system.
3. **Notifications:** Represents messages sent to clients when transactions occur.
4. **Reports:** Represents the generated financial summaries and insights.

Attributes describe the properties of each entity. Here's a breakdown:

1. **User**:
    a. User id (Primary Key)
    b. Name
    c. Email
    d. Password
    e. Phone
    f. Address
2. **Transaction**:
    a. Transaction id (Primary Key)
    b. User id (Foreign Key referencing User)
    c. Client name
    d. Amount
    e. Payment date
    f. Payment type (e.g., advance or installment)
    g. Payment mode (e.g., cash, card, online)

    h. Payment month
3. **Notification**:
    a. Notification id (Primary Key)
    b. User id (Foreign Key referencing User)
    c. Transaction id (Foreign Key referencing Transaction)
    d. Notification type
    e. Sent time
4. **Report**:
    a. Report id (Primary Key)
    b. User id (Foreign Key referencing User)
    c. Total transactions
    d. Total amount
    e. Pending amount
    f. Generated date

The relationships between the entities is as follows:

1. **User → Transaction**
    a. A user can record multiple transactions.
    b. **Relationship**: One-to-Many (1: N)
2. **Transaction → Notification**
    a. Each transaction can trigger a notification.
    b. **Relationship**: One-to-One (1:1) or One-to-Many (1: N)
3. **User → Notification**
    a. A user can receive multiple notifications.
    b. **Relationship**: One-to-Many (1: N)
4. **User → Report**
    a. A user can generate multiple reports.
    b. **Relationship**: One-to-Many (1: N)
5. **Transaction → Report**
    a. A report includes multiple transactions.
    b. **Relationship**: One-to-Many (1: N)

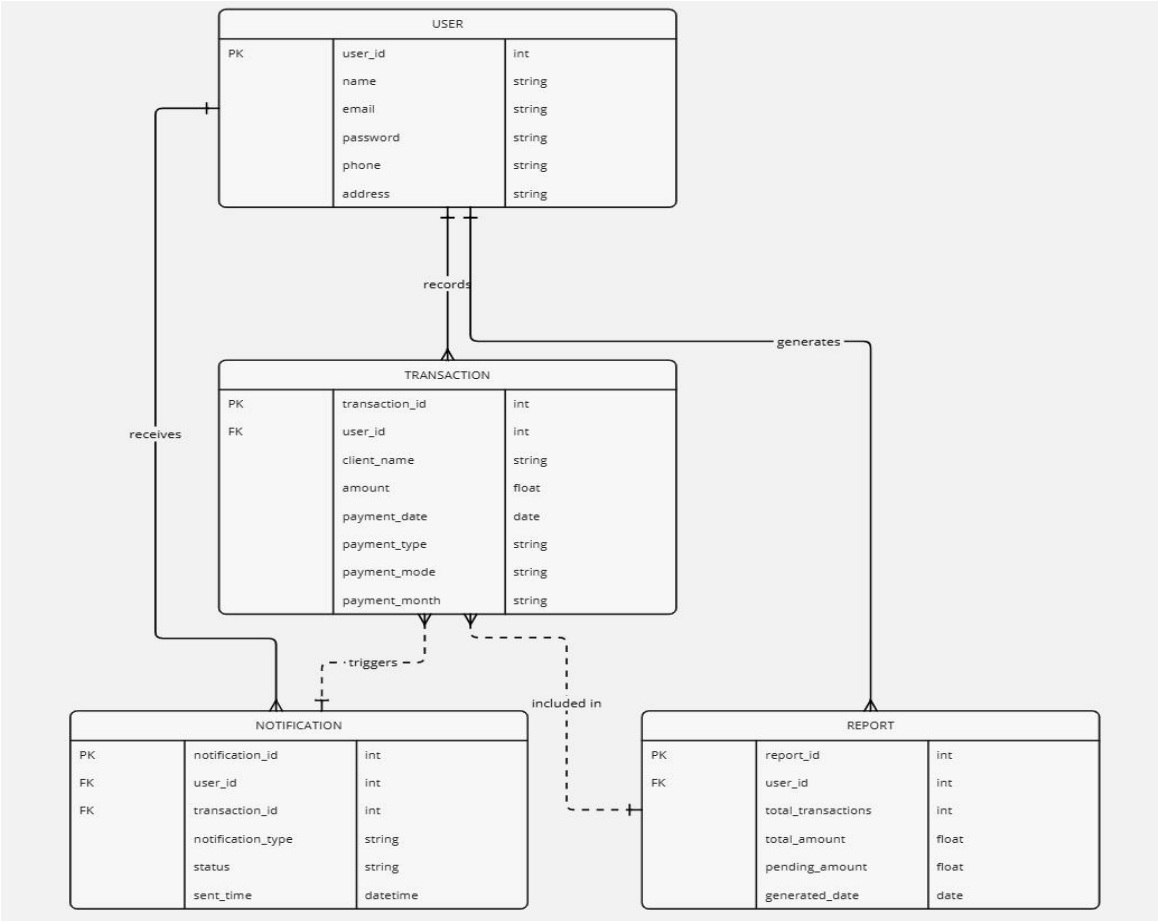

Fig. 5.2: ER Diagram which represents the relationship between entities and their attributes.

CHAPTER 6

PROJECT OUTCOMES

- **Improved Transaction Management**

  Users can add, view, and organize transactions efficiently.

- **Automated Notifications**

  Clients receive timely payment updates, reducing manual follow-ups.

- **Financial Insights**

  Dynamic graphs and tables provide clarity on revenue, expenses, and pending dues.

- **Enhanced User Experience**

  Intuitive UI with theme customization improves usability.

- **Scalability and Future Enhancements**

  Scope to add new features like PDF/Excel exports and advanced analytics.

Sign Up Page

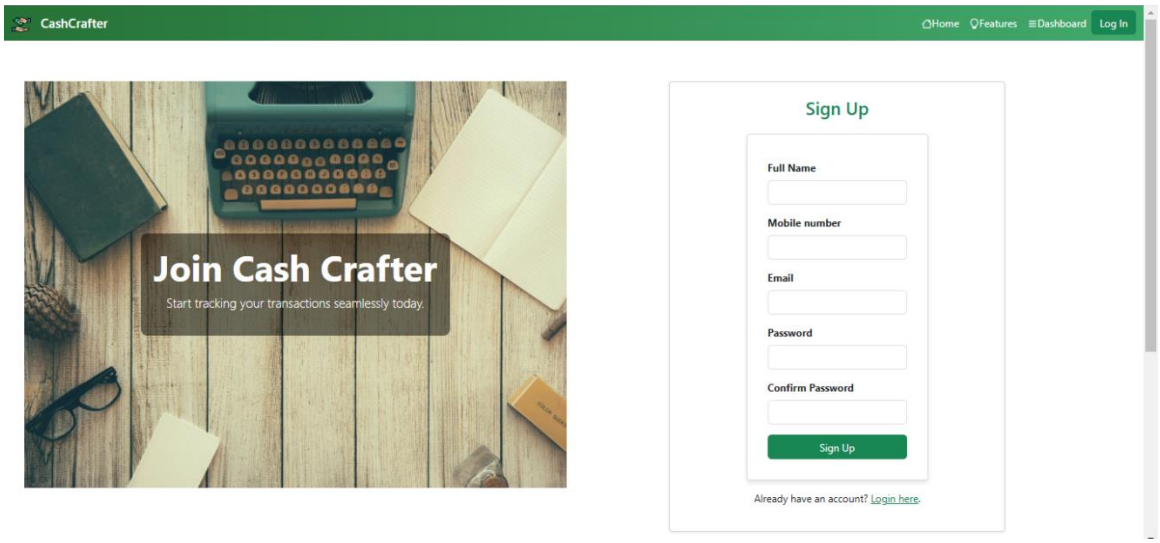

Fig 6.1: The above figure shows the Sign-Up page which starts the User Authentication module of the application.

Login Page

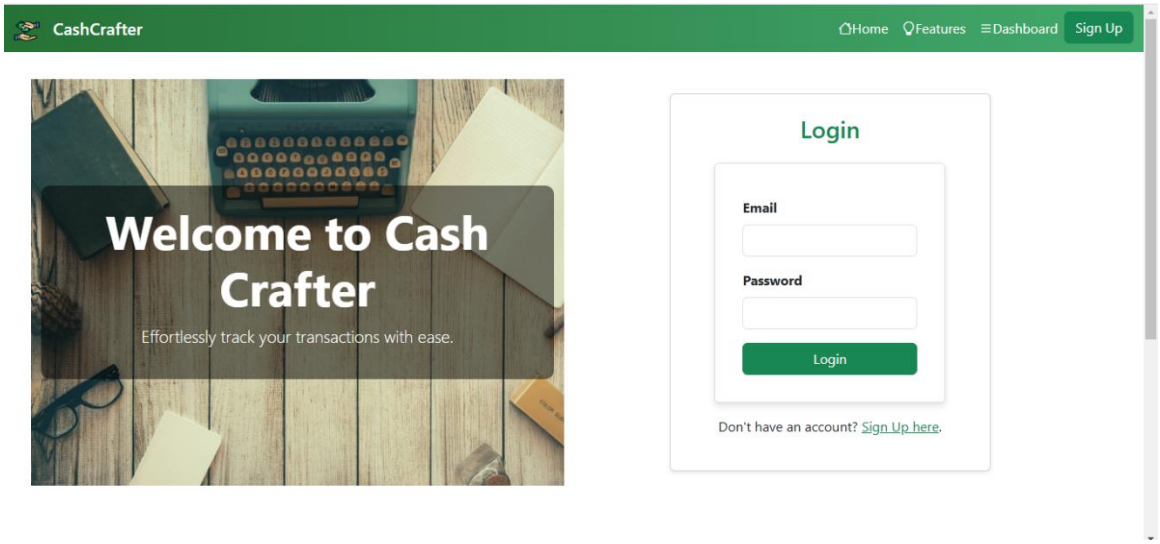

Fig 6.2: The above figure shows the Login Page which is second part of User Authentication Module.

Home Page

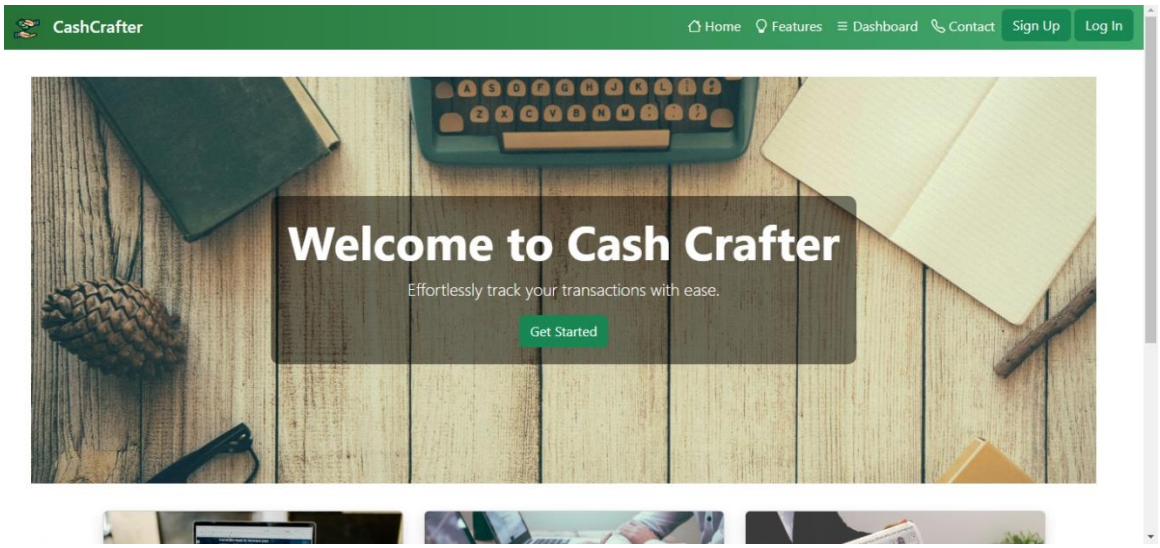

Fig 6.3: The above figure shows the landing page describing the main functionality of the application and attracting the target audience to use it.

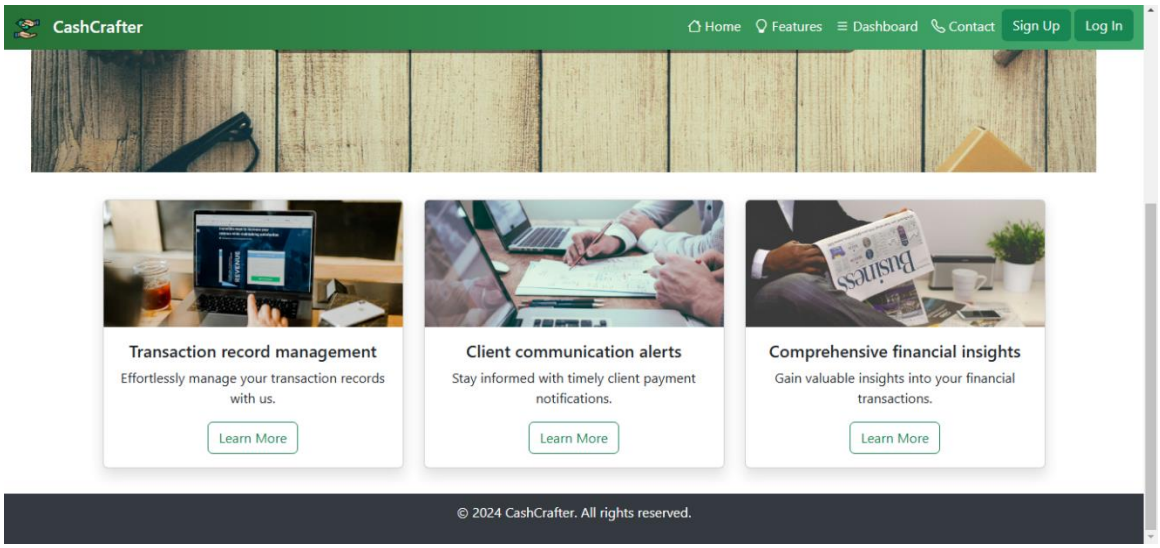

Fig 6.4: This figure is the continuing part of Home page where all the functionalities can be accessed and can be moved to next page to perform tasks.

Features Page

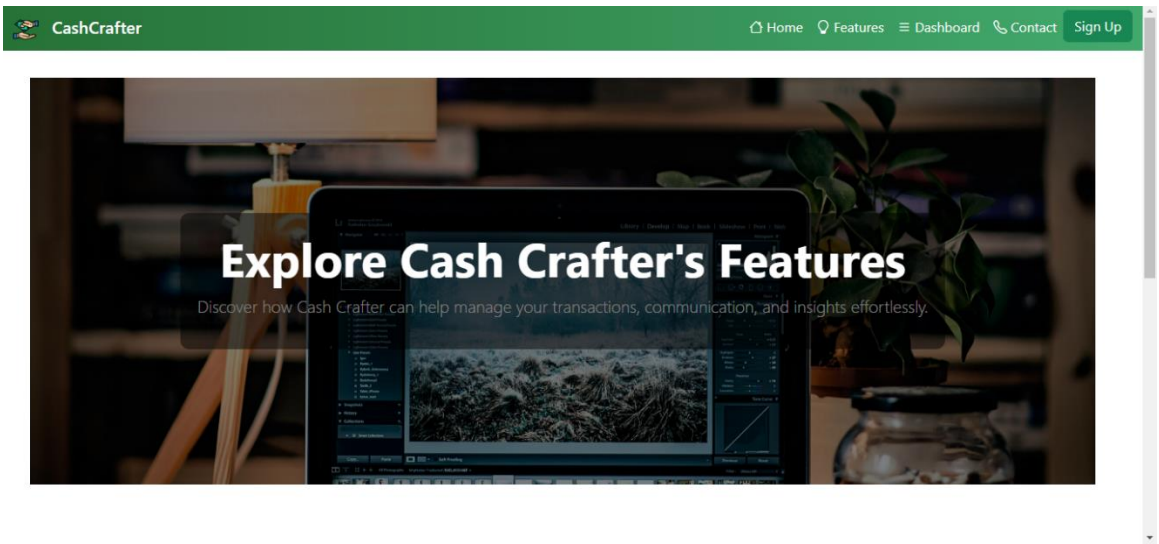

Fig 6.5: The above figure shows the features page which describes users how to use the functionalities provided in the application.

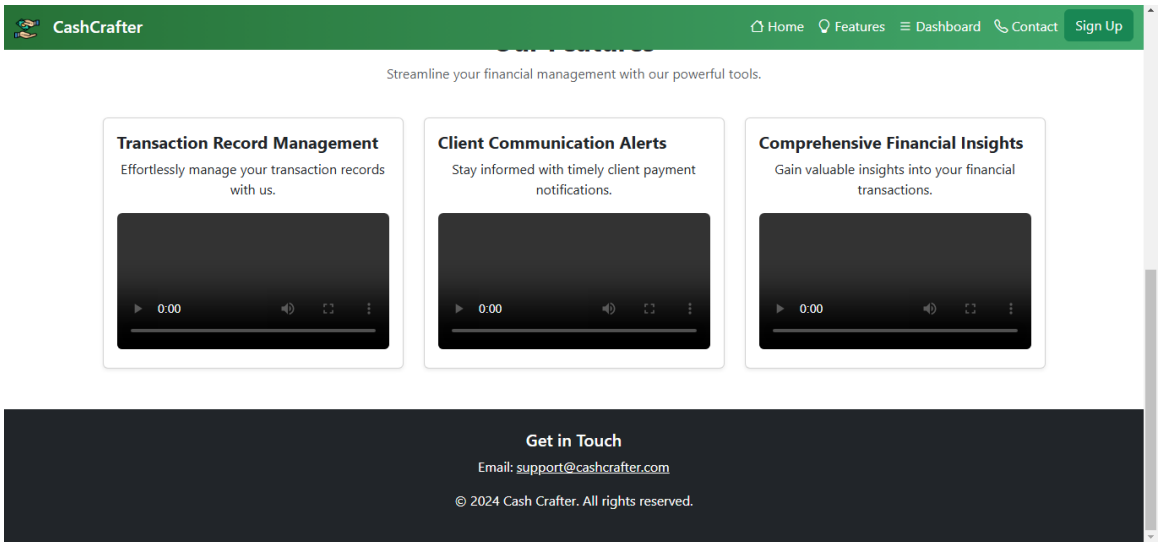

Fig 6.6: This figure is the continuing part of features page which has videos related to each functionality.

Dashboard Page

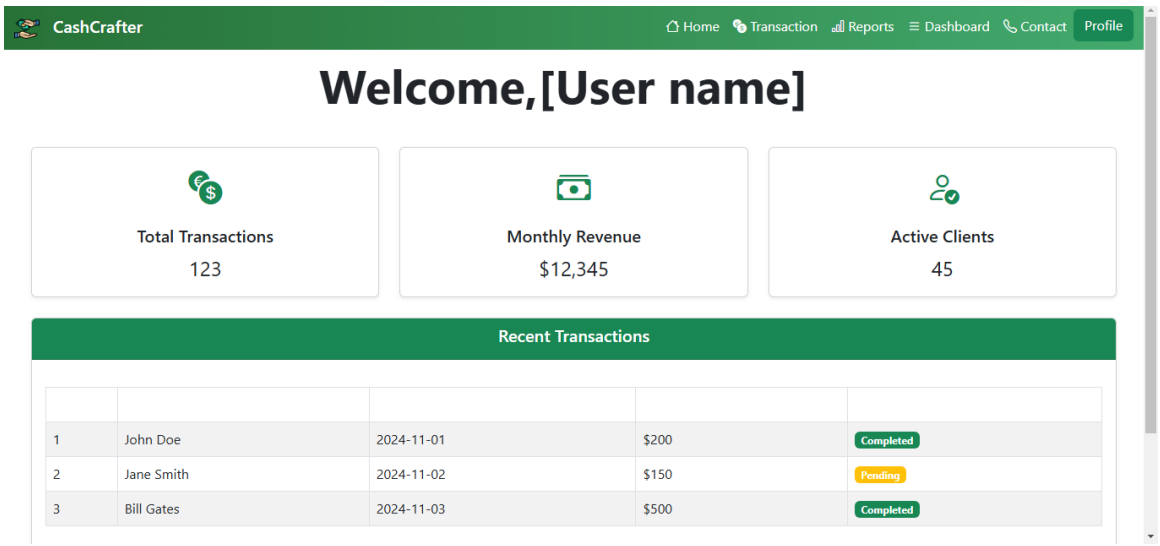

Fig 6.7: Dashboard page in this figure shows the transaction summary and welcomes user after successful login.

Transaction Page

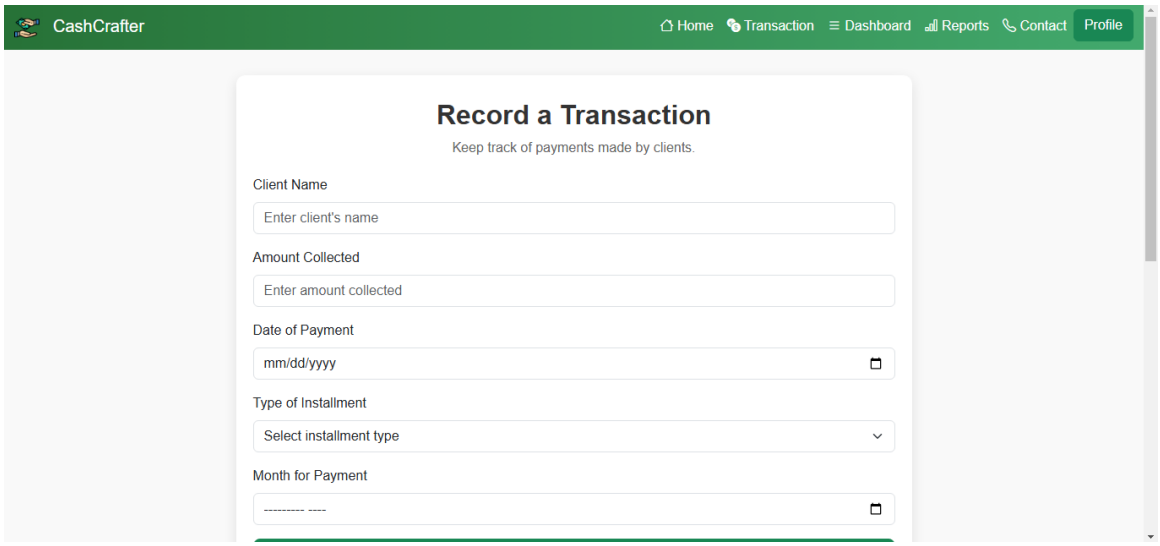

Fig 6.8: Transaction page, which has a form to enter details of the client from whom the payment is collected and needs to be recorded.

Transaction records on transaction page

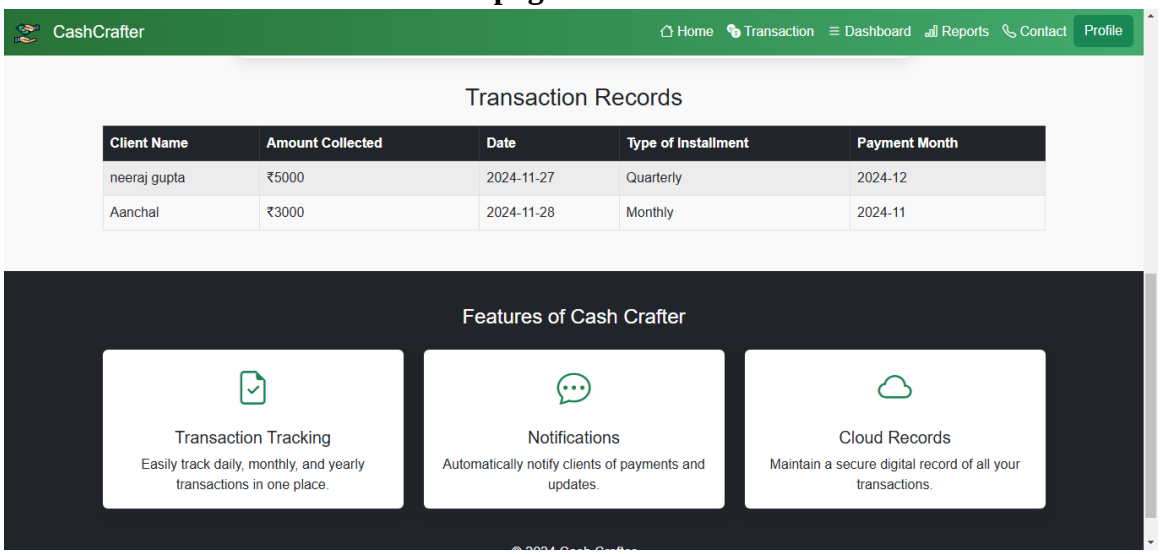

Fig 6.9: This is the extended part of Transaction page, where the records are shown in the form of list.

Contact Page

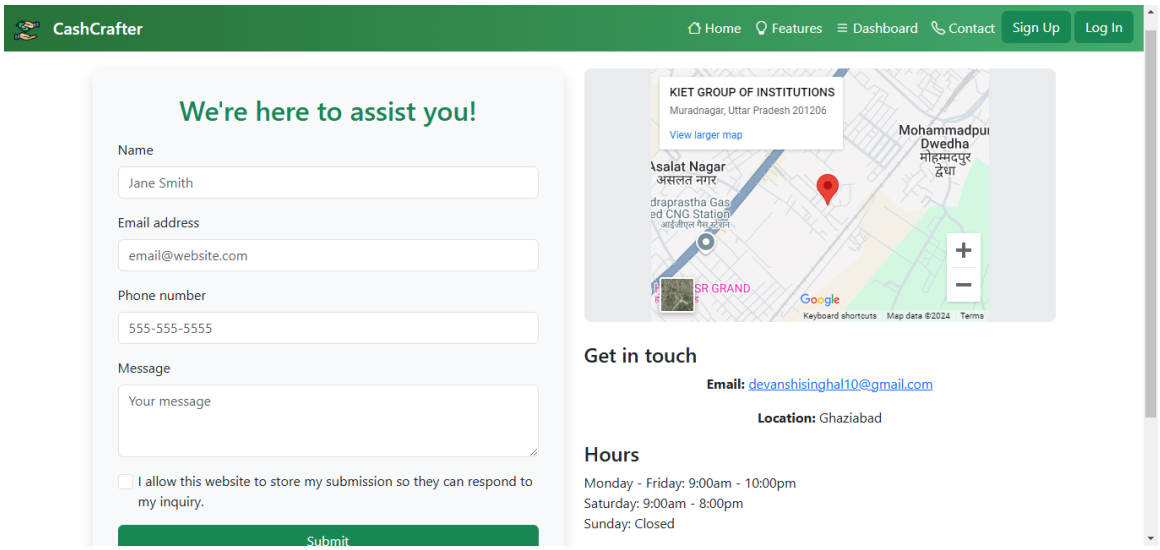

Fig 6.10: Contact Page which collects the feedback and provides access to contact admin whenever needed.

REFERENCES

1. Web Resources

- https://www.twilio.com/docs/sms
- https://sendgrid.com/docs
- https://www.chartjs.org/
- https://getbootstrap.com/
- https://dev.mysql.com/doc/
  https://www.linkedin.com/learning/certificates/1b40ff509afa7a85ea80e427c20f98153236764ae3da51f363d220ad7ab1aad2?trk=share_certificate

2. Books

1. Casciaro, M., & Mammino, L. (2020). *Node.js design patterns* (3rd ed.). Packt Publishing.
2. Grinberg, M. (2018). *Flask web development: Developing web applications with Python* (2nd ed.). O'Reilly Media.
3. Zammetti, F. (2021). *Modern full-stack development: Using TypeScript, React, Node.js, Webpack, and Docker*. Apress.

3. Research Papers

1. Smith, J., & Brown, A. (2021). Enhancing financial efficiency through automation. *International Journal of Finance and Technology*, 8(2), 45-62.
2. Lee, K., & Chang, Y. (2020). The impact of real-time client notifications on financial transparency. *Proceedings of the International Conference on Digital Communication Systems*, 2020, 89-95.
3. Johnson, R., & Williams, D. (2019). User-centered design for financial applications. *UX Design Trends in Technology*, 12(4).

4. APIs and Tools

- https://www.twilio.com/
- https://sendgrid.com/
- https://github.com/

5. Software and Frameworks

- https://nodejs.org/
- https://getbootstrap.com/