Akorn

**A Project Report**

Submitted by

**BHARATH KUMAR M**

(Reg no: MJAWMCA012)

In partial fulfilment of requirements

For the Award of the Degree of

**Master of Computer Application (MCA)**

**University of Calicut**

****

**Centre for Computer Science & Information Technology**

**(University of Calicut)**

**MANJERI**

**2022-2024**

Akorn

**A Project Report**

Submitted by

**BHARATH KUMAR M**

(Reg no: MJAWMCA012)

In partial fulfilment of requirements

For the Award of the Degree of

**Master of Computer Application (MCA)**

**University of Calicut**

****

**Centre for Computer Science & Information Technology**

**(University of Calicut)**

**MANJERI**

**2022 - 2024**

**CERTIFICATE**

This is to certify that the project report entitled Akorn submitted by Bharath kumar M (Register Number: MJAWMCA012) to University of Calicut for the award of the degree of Master of Science (MCA) in Computer Application is a bonafide record of the project work carried out by him/her under my supervision and guidance. The content of the report, in full or parts have not been submitted to any other Institute or University for the award of any other degree or diploma.

**Mr. Rinshad K**

**Assistant Professor**

**CCSIT MANJERI**

**Mrs. SHALINI NP**

**Assistant Coordinator**

**CCSIT MANJERI**

**Place:**

**Date:**

Certified that the candidate was examined by us in the Project Viva Voce Examination held on ………………………………………………… and his Register Number is ………………………………………..

**Examiners:**

**1.**

**2.**

**Declaration**

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person or material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

**Bharath kumar M (MJAWMCA012)**

**Place:**

**Date:**

**ACKNOWLEDGEMENT**

The success of the project depends upon the effort invested. It’s my duty to acknowledgement and thanks the individuals who have the contributed in the successful completion of the project. First and foremost, I praise the God Almighty for the grace he showered on me during my studies as well as day to day life activities.

I take this opportunity to express my profound and wholehearted thanks to **Ms. Shalini NP** (Associate Coordinator, CCSIT Manjeri) for providing ample facilities made to undergo my project successfully. I express my heartfelt gratitude to our internal guide **Mr. Rinshad K** (Assistant professor, CCSIT Manjeri) for their constant encouragement throughout the course of study. They have taken pain to go through the project and make necessary correction as and when needed. I am also thankful to my department faculties for their continuous motivation for the successful completion of my project.

I wish to express my love and respect for my parents, for their support, contribution and encouragement which helped me a lot to complete this project successfully. I finally give a bunch of thanks to my friends and all my well-wishers who had supported me directly and indirectly throughout the project with their constructive criticisms.

Abstract

The "Akorn" is a dynamic and user-centric web application tailored to facilitate the realization of creative projects and innovative business ideas. The platform serves as a virtual launchpad, connecting visionaries and entrepreneurs with a global community of backers eager to support and invest in compelling ventures. The key components of the platform include project creation, funding goals, and backer rewards. "Akorn" is designed to be a catalyst for innovation, providing a digital ecosystem where creativity thrives and dreams find wings. By combining a user-friendly interface with robust features, the platform empowers creators to transform their ideas into reality while fostering a supportive community that believes in the power of collective impact.

**Table of Contents**

[1. INTRODUCTION 1](#_Toc171989030)

[2. SYSTEM ANALYSIS 3](#_Toc171989031)

[3. SYSTEM STUDY 8](#_Toc171989032)

[4. SYSTEM SPECIFICATION 17](#_Toc171989033)

[5. SYSTEM ARCHITECTURE 20](#_Toc171989034)

[7. SYSTEM DESIGN AND METHODOLOGY 23](#_Toc171989035)

[8. SYSTEM DEVELOPMENT 34](#_Toc171989036)

[9. SYSTEM TESTING 37](#_Toc171989037)

[10. CONCLUSION 40](#_Toc171989038)

[11. IMPACT AND FUTURE ENHANCEMENTS 42](#_Toc171989039)

[12. SCREENSHOTS 44](#_Toc171989040)

[13. REFERENCES 50](#_Toc171989041)

1. INTRODUCTION

Introduction

Welcome to **AKORN**, the visionary project incubator. In a landscape where innovation meets opportunity, **AKORN** stands as a beacon for dreamers and creators seeking to bring their boldest ideas to life. With a dynamic fusion of cutting-edge technology and community-driven support, **AKORN** redefines the investment experience, empowering both visionaries and backers alike to embark on transformative journeys together.

At the heart of **AKORN** lies a revolutionary platform designed to transcend the limitations of traditional investment systems. Gone are the days of generic project presentations and disconnected support networks. Instead, Akorn offers a vibrant ecosystem where creativity thrives and collaboration flourishes.

With **AKORN** innovative features seamlessly integrated, **AKORN** presents an unparalleled opportunity for creators to showcase their projects with unparalleled depth and detail. From captivating multimedia presentations to flexible funding goals tailored to individual project needs, **AKORN**’s comprehensive project creation interface empowers creators to captivate and inspire.

Transparency and accountability form the bedrock of **AKORN**’s ethos. Backers are granted real-time visibility into funding progress, ensuring a clear understanding of how their contributions propel projects towards success. With diverse funding options and personalized reward tiers, backers are invited to engage with projects on a deeper level, forging meaningful connections with the creators they support

The app we propose addresses these crucial issues head-on, building a bridge of transparency between government projects and citizens. By empowering both sides, we can unlock a future of responsible resource management, trustworthy governance, and improved project outcomes for all.

2. SYSTEM ANALYSIS

System Analysis

**1. Introduction:**

This section outlines the system analysis for Akorn, a web application designed to revolutionize investment in India. Akorn aims to bridge the gap between investors and founders.

**2. Problem Definition:**

The Akorn project aims to establish a revolutionary investment platform to transform the traditional investment experience. This platform seeks to address several challenges present in current investment systems, particularly focusing on improving the connection between investors and entrepreneurs.

**3. Existing System Landscape:**

Current investment platforms often lack personalized project presentations, and seamless integration these limitations hinder user experience and project success rates.

**4. Requirements Analysis:**

The requirement analysis for the Akorn project involves assessing the needs and expectations for a investment platform based on the MERN stack (MongoDB, Express.js, React, Node.js). This includes reviewing existing literature on investment platforms, analyzing user expectations, and understanding technological advancements in web development.

**a. Functional Requirements**

1. **User Management:**
   * Registration and login with email or social media accounts.
   * Profile management for creators and backers.
   * Role-based access control for administrators, creators, and backers.
2. **Project Management:**
   * Project creation with multimedia presentations (images, videos, text).
   * Flexible funding goals.
   * Real-time visibility of funding progress and backer interaction.
3. **Financial Transactions:**
   * Dummy payment gateway integration for transactions.
   * Compliance with financial regulations and data protection laws.

##### **b. Non-functional Requirements**

1. **Performance:**
   * Response time optimization for seamless user interaction.
   * Scalability to handle increasing user traffic and project submissions.
2. **Security:**
   * Encryption of sensitive data (user credentials).
3. **Usability:**
   * Intuitive user interface design for easy navigation and project discovery.
   * Accessibility features to accommodate diverse user needs.

**6. System Design:**

Akorn will be a web-based application accessible through various devices (smartphones, tablets, laptops) to ensure wider accessibility across India.

* **Front-End:** A user-friendly and responsive interface likely developed using a popular JavaScript framework like React.js.
* **Back-End:** A server-side application built with the MERN stack (MongoDB, Express.js, React.js, Node.js) for scalability and performance.
* **Database:** MongoDB, a document-oriented database, will likely be used for its flexibility and ease of handling project data.

**7. System Interfaces:**

Akorn will interact with the following external systems:

* **Payment Gateway:** If managing contractor expenses, a secure payment gateway integration might be needed. In its current state, the web app is not using any payment gateway as the amount of money that is to be transferred for a project is much higher than that may be supported by most systems that are freely available.
* **Graph Service:** Integrating a Graph service like rechart.js allow visualizing project. They also make it simpler to view and understand.

8. Feasibility Study

The feasibility study assesses the viability of implementing the Akorn project. This study evaluates various aspects, including technical, economic, operational, and legal feasibility, to determine the project's potential success.

1. Technical Feasibility

* Platform Development: Developing the Akorn platform requires advanced web development skills, including front-end and back-end development, database management. API’s and existing infrastructure provides a strong foundation for technical implementation.
* Scalability: The platform must accommodate a growing user base and increasing project submissions without compromising performance. Scalability measures, such as cloud-based infrastructure and optimized code, are necessary to ensure seamless operation.

2. Economic Feasibility

* Cost Analysis: The project involves initial development costs, including software development, infrastructure setup.
* Market Potential: The demand for investment platforms is significant, with an increasing number of entrepreneurs and creatives seeking funding for innovative projects. Akorn's unique features and user-centric approach position it favorably in the market.

3. Operational Feasibility

* User Experience: Akorn's success depends on providing a seamless and intuitive user experience for both creators and backers. Operational feasibility hinges on the platform's ability to facilitate project creation, funding managemen.
* Administrative Processes: Effective management of user accounts, project submissions, and financial transactions is essential for operational success. Streamlined administrative processes, supported by robust backend systems, ensure smooth platform operation.

4. Legal Feasibility:

* Regulatory Compliance: Compliance with relevant regulations, such as data protection laws and financial regulations, is critical to operating a investment platform. Legal feasibility requires thorough adherence to legal requirements in all jurisdictions where Akorn operates.
* Intellectual Property: Protecting intellectual property rights for project creators and ensuring compliance with copyright laws is essential to mitigate legal risks and maintain trust within the community.

**Conclusion:**

The feasibility study indicates that the Akorn project is technically, economically, operationally, and legally feasible. With proper planning, execution, and ongoing management, Akorn has the potential to become a leading investment platform, fostering innovation, creativity, and meaningful connections among creators and backers.

3. SYSTEM STUDY

System Study

**1. Architecture and Technologies**

* **Frontend:** Built using React.js for dynamic and interactive user interfaces. Ensures responsive design for optimal viewing across devices.
* **Backend:** Utilizes Node.js for server-side logic, handling requests, and integrating with MongoDB.
* **Database:** MongoDB, a NoSQL database, chosen for scalability and flexibility in managing diverse data types.
* **API Integration:** Custom-built APIs using Express.js to manage project creation, user authentication, and transaction handling.
* **Cloud Infrastructure:** AWS or similar cloud services for scalability, reliability, and performance optimization.

**2. Functional Components**

**User Management**

* **Registration and Authentication:** Secure user registration and login via email or social media accounts.
* **Profile Management:** Allows creators and backers to manage their profiles and preferences.
* **Role-Based Access Control:** Admin, creator, and backer roles with specific permissions for each role.

**Project Management**

* **Project Creation:** User-friendly interface for creators to create detailed project profiles, upload multimedia content (images, videos), and set funding goals.
* **Funding Options:** Flexible funding goals and real-time visibility of funding progress for backers.

**Financial Transactions**

* **Payment Gateway Integration:** Dummy payment gateway initially for transaction simulation.

**Administrative Tools**

* **Dashboard:** Comprehensive dashboard for administrators to manage user accounts, oversee project submissions, and monitor platform activities.

**3. Non-Functional Requirements**

* **Performance:** Optimized for fast response times and scalability to handle increasing user traffic and project submissions.
* **Security:** Encryption of sensitive data, secure API endpoints, and regular security audits.
* **Usability:** Intuitive UI/UX design for easy navigation and seamless user experience.

**4. Environmental Details**

* **Hardware:** Standard servers or cloud-based infrastructure to host MongoDB, Node.js server, and React frontend.
* **Software:** Compatible operating systems (Linux, Windows) for Node.js deployment, MongoDB for data storage, and development tools (e.g., Visual Studio Code).

**Conclusion**

The system design for Akorn incorporates modern web development technologies to create a robust platform that fosters innovation and collaboration between creators and backers. By leveraging the MERN stack's capabilities, Akorn ensures scalability, security, and a seamless user experience, positioning itself as a leading investment platform in the industry.

This design provides a foundational framework for further development, ensuring that Akorn meets its objectives of transforming the investment landscape through technology-driven solutions and community-driven support.

**Proposed System: Akorn**

**1. Architecture Overview**

Akorn will utilize a microservices architecture to ensure scalability, modularity, and maintainability. Key components include:

**Frontend:** Developed using React.js, ensuring a responsive and interactive user interface.

**Backend Services:** Built with Node.js and Express.js, providing robust API endpoints for data management and business logic.

**Database:** MongoDB, a NoSQL database, chosen for its flexibility and scalability.

**2. Functional Components**

**User Management**

**Authentication and Authorization:** Secure registration and login using email or social media accounts. Role-based access control for administrators, creators, and backers.

**Profile Management:** Users can manage their profiles.

**Project Management**

**Project Creation:** Founders can create detailed project profiles with multimedia presentations (images, videos, text). Includes options for flexible funding goals and project milestones.

**Funding Management:** Real-time visibility of funding progress, backer interaction, and pledge management.

**Financial Transactions**

**Administrative Tools**

**Dashboard:** Comprehensive dashboard for administrators to manage user accounts, monitor project submissions, and oversee platform activities.

**3. Non-Functional Requirements**

**Performance:** Optimized for high performance and scalability to handle concurrent user interactions and project submissions.

**Security:** Data encryption, secure API endpoints, and regular security audits to protect user information and financial transactions.

**Usability:** Intuitive UI/UX design with responsive layouts for seamless navigation across devices.

**4. Environmental Details**

**Hardware:** Utilization of cloud-based infrastructure for scalability, including servers for hosting MongoDB, Node.js server instances, and frontend React applications.

**Software:** Operating system compatibility (Linux, Windows) for Node.js deployment, MongoDB for data storage, and development tools (e.g., Visual Studio Code) for coding and debugging.

**5. Potential Challenges:**

Implementing a project like Akorn, which aims to revolutionize the investment platform landscape, can face several potential challenges across technical, operational, economic, and regulatory domains. Here are some key challenges to consider:

**1. Technical Challenges**

* + **Scalability:** Ensuring the platform can handle a growing number of users, projects, and transactions without compromising performance.
  + **Integration Complexity:** Integrating with various third-party APIs (e.g., payment gateways, social media APIs) for seamless functionality and user experience.
  + **Data Security:** Implementing robust security measures to protect sensitive user data, financial transactions, and intellectual property rights.
* **Technology Stack Updates:** Managing updates and patches for the MERN stack components (MongoDB, Express.js, React, Node.js) to maintain compatibility and security.

**2. Operational Challenges**

* + **User Engagement:** Encouraging consistent user engagement from both creators and backers to ensure active participation and project success.
  + **Content Management:** Effectively moderating and managing project content to maintain quality, legality, and adherence to platform guidelines.
  + **Customer Support:** Providing responsive and effective customer support to address user queries, technical issues, and disputes.
  + **Platform Stability:** Ensuring high availability and reliability of the platform with minimal downtime or disruptions.

**3. Economic Challenges**

* + **Monetization Strategy:** Developing and refining a sustainable monetization strategy that aligns with user expectations and industry standards.

**4. Regulatory Challenges**

* + **Compliance:** Navigating complex regulatory frameworks related to financial transactions, data protection, and intellectual property rights in multiple jurisdictions.
  + **Legal Risk Management:** Mitigating legal risks associated with user agreements, terms of service, copyright infringement, and dispute resolution.
  + **Monitoring and Reporting:** Implementing systems for monitoring platform activities and reporting requirements to regulatory authorities as necessary.

**5. Strategic and Execution Challenges**

* + **User Education:** Educating users (both creators and users) about the platform's features, benefits, and best practices to maximize engagement and success rates.
  + **Partnership Development:** Establishing strategic partnerships with investors, accelerators, and industry influencers to enhance platform credibility and visibility.
  + **Adaptability:** Staying agile and responsive to market trends, user feedback, and technological advancements to continuously improve the platform.

**5. Mitigation Strategies:**

To address the potential challenges faced by Akorn in implementing its revolutionary investment platform, here are some mitigation strategies across technical, operational, economic, and regulatory aspects:

* 1. **Technical Mitigation Strategies**
* **Scalability:**
  + Implement cloud-based infrastructure (e.g., AWS) with auto-scaling capabilities to handle spikes in user traffic and project submissions.
  + Use microservices architecture for modular development, allowing independent scaling of different components.
* **Integration Complexity:**
  + Conduct thorough API compatibility and performance testing during development and updates.
  + Maintain clear documentation and provide developer support for integrating with third-party APIs.
* **Data Security:** 
  + Implement end-to-end encryption for sensitive data in transit and at rest.
  + Regularly conduct security audits and penetration testing to identify and mitigate vulnerabilities.
* **Technology Stack Updates:**
  + Establish a proactive schedule for updating components of the MERN stack, ensuring compatibility and security patches are applied promptly.
  + Use version control and staging environments for testing updates before deployment to production.

1. **Operational Mitigation Strategies**

* **User Engagement:**
  + Implement gamification strategies (e.g., badges, rewards) to incentivize active participation from both creators and backers.
  + Foster a community-driven approach with forums, webinars, and networking events to enhance engagement.
* **Content Management:**
  + Deploy automated and manual moderation tools to ensure compliance with platform guidelines and legal requirements.
  + Establish clear submission and review processes to maintain quality and authenticity of project content.
* **Customer Support:**
  + Offer customer support (email) with prompt response times.
* **Platform Stability:**
  + Use monitoring tools to track platform performance metrics (e.g., uptime, response time).
  + Implement disaster recovery plans and regular backups to minimize downtime and data loss.

1. **Economic Mitigation Strategies**
   * **Financial Sustainability:**
   * Conduct regular financial analyses and adjust pricing models (e.g., transaction fees, premium features) based on market trends and user feedback.
   * Diversify revenue streams (e.g., sponsorship opportunities, partnerships) to reduce reliance on transaction fees alone.
   * **Market Competition:**
   * Focus on unique selling propositions (USPs) such as superior user experience, transparency, and community engagement.
   * Continuously monitor and analyze competitor strategies to differentiate and adapt accordingly.
2. **Strategic and Execution Mitigation Strategies**
   * **User Education:**
   * Develop comprehensive user guides, tutorials, and FAQs to educate users about platform features, best practices, and compliance requirements.
   * Offer webinars, workshops, and online courses to empower users with knowledge and skills relevant to crowdfunding and investments.
   * **Partnership Development:**
   * Cultivate strategic partnerships with industry influencers, venture capitalists, and accelerators to enhance credibility and attract high-quality projects and backers.
   * Leverage partnerships for co-marketing opportunities and access to broader networks of investors and innovators.
   * **Adaptability:**
   * Maintain agile development practices to quickly iterate and adapt platform features based on user feedback and market dynamics.
   * Foster a culture of innovation within the development team to explore emerging technologies and trends that could enhance platform functionality and user experience.

4. SYSTEM SPECIFICATION

System Specifications

**1. System Overview:**

Akorn employs a modern microservices architecture to ensure scalability, modularity, and ease of maintenance. Key architectural components include:

* **Frontend:** Developed using React.js, providing a responsive and interactive user interface (UI) for creators, backers, and administrators.
* **Backend Services:** Built with Node.js and Express.js, serving as the core business logic layer and providing RESTful APIs for data management and integration.
* **Database:** MongoDB, a NoSQL database, chosen for its flexibility and scalability to store diverse data types related to users, projects, transactions, and administrative data.

**2. Hardware Requirements:**

* **Server-Side:**
  + A reliable web server with good processing power and memory (scalable based on anticipated user traffic).
  + Ample storage space to accommodate user data, project information, and application files.

**3. Software Requirements:**

* **Client-Side:**
  + Modern web browser (Chrome, Firefox, Safari, Edge) with JavaScript enabled.
  + Internet connection (broadband or mobile data).
* **Server-Side:**
  + **MERN Stack:**
    - Node.js (latest stable version)
    - MongoDB (database server)
    - Express.js (web framework)
    - React.js (front-end library)
  + **Additional Considerations:**
    - Recharts.js (for data visualization)

**4. System Interfaces:**

* **User Interface (UI):** A user-friendly web interface accessible through various devices (smartphones, tablets, laptops) for citizen and government user interaction.
* **Database Interface:** The application will interact with a MongoDB database to store user data, project information, reports, and other relevant data.
* **External APIs:** Rechart.js for visualization for data as graphs.

**5. Performance Requirements:**

* The system should be responsive and provide fast loading times for a seamless user experience.
* The application should be scalable to accommodate a growing user base and handle increasing data volumes efficiently.

**6. Security Requirements:**

* Secure user authentication and authorization mechanisms to protect user data.
* Secure communication channels using HTTPS to encrypt data transmission.
* Regular security updates and vulnerability patching for the server and application components.

**7. Non-Functional Requirements:**

* **Accessibility:** The application should be accessible to a broad audience across India's diverse population. This necessitates a user-friendly interface that functions well on various devices and internet speeds.
* **Maintainability:** The code should be well-documented, modular, and follow coding best practices to facilitate future maintenance and updates.
* **Scalability:** The system architecture should be designed to scale horizontally to accommodate a growing user base and increasing data volumes.

5. SYSTEM ARCHITECTURE

System Architecture

**Frontend (Client-Side):**

React.js: At the heart of Akorn user interface lies React.js, a popular JavaScript library renowned for its component-based architecture. This approach breaks down the UI into reusable components, promoting clean and maintainable code. React also boasts a virtual DOM (Document Object Model), ensuring efficient UI updates that translate into a smooth and responsive user experience for citizens interacting with the platform.

JavaScript: Since React primarily relies on JavaScript for component development, Akorn extensively utilizes this language for front-end functionality. This not only fosters streamlined development but also reduces the need for developers to switch between different programming languages, minimizing context-switching burdens.

**Additional Libraries:**

Data Visualization with Recharts.js: Akorn empowers users to understand project progress and key metrics through interactive visualizations. To achieve this, libraries like Recharts.js, built on top of React, play a crucial role. Recharts.js simplifies the creation of charts and graphs, presenting complex data in an easy-to-understand format.

**Backend (Server-Side):**

**Node.js**: Powering the Akorn server-side is Node.js, a JavaScript runtime environment. A key advantage of Node.js is its ability to efficiently handle asynchronous requests, making it ideal for web applications like Akorn that anticipate a significant user base. Node.js efficiently manages multiple user interactions simultaneously, ensuring smooth operation even with high traffic.

**Express.js**: Built on top of Node.js, Express.js acts as a web framework that streamlines back-end development. Express.js provides a robust set of features for handling API requests, routing, and middleware. These functionalities simplify tasks like handling user authentication, managing data flow, and defining how different parts of the application communicate with each other.

**MongoDB**: Akorn departs from traditional relational databases and embraces MongoDB, a NoSQL document database. MongoDB's strength lies in its flexible data storage capabilities. It stores project data, user information, and reports in a document-oriented format, which is efficient for handling diverse data types. This flexibility proves advantageous as Akorn scales to accommodate a growing user base and potentially even more complex data structures in the future.

**Integrated Communication: API - The Bridge Between Frontend and Backend**

Akorn utilizes an Application Programming Interface (API) to establish seamless communication between the React.js front-end and the Node.js/Express.js back-end. The API acts as a bridge, allowing the front-end to make requests to the back-end for various purposes. These requests might involve retrieving project data, submitting citizen reports, or facilitating two-way communication between citizens and government agencies.

**Synergy of Technologies: The MERN Stack Advantage**

The decision to leverage the MERN stack for Akorn was not arbitrary. Here's a closer look at the key benefits this combination offers:

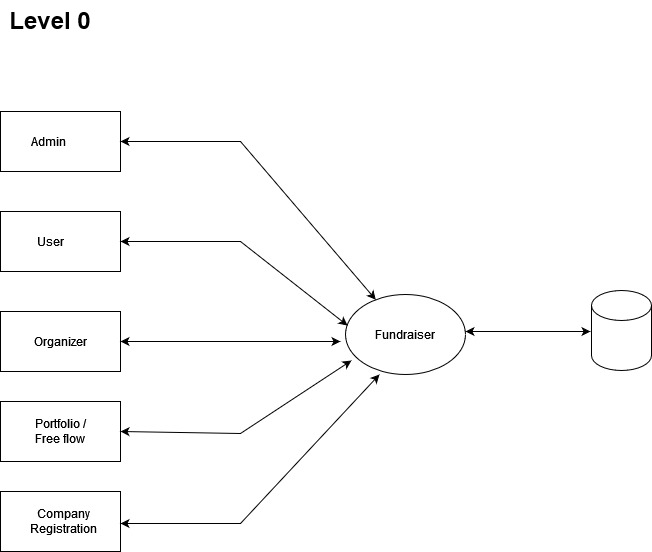
* **Scalability and Performance**: The MERN stack is renowned for its ability to handle large datasets and high user traffic. As Akorn gains traction and attracts more users, the MERN stack ensures the platform can scale efficiently to accommodate this growth.
* **Full-Stack Development**: A significant advantage of the MERN stack is its ability to provide a comprehensive set of technologies for both front-end and back-end development. This streamlines the development process as developers can leverage their JavaScript expertise across the entire application stack.
* **Open-Source and Community-Driven**: All components of the MERN stack are open-source, making development cost-effective. Additionally, this open-source nature fosters large developer communities, allowing Akorn to benefit from a wealth of knowledge, resources, and ongoing improvements within these communities.

7. SYSTEM DESIGN AND METHODOLOGY

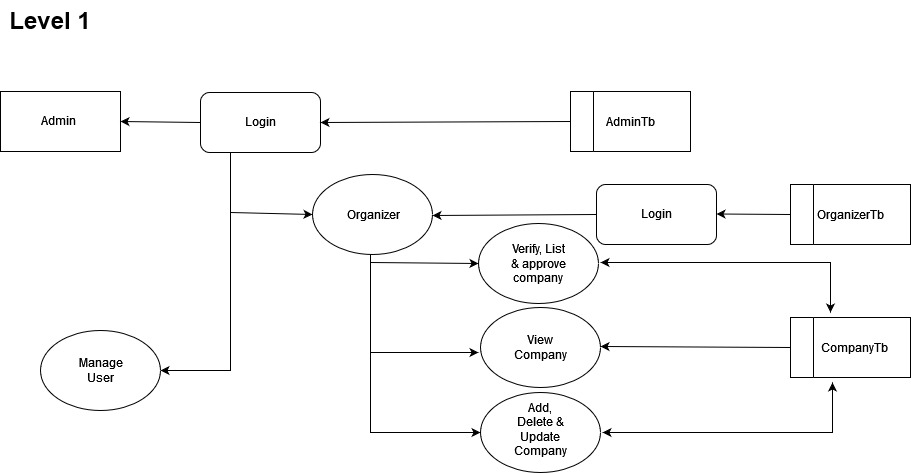
System Design and Methodology

DFD

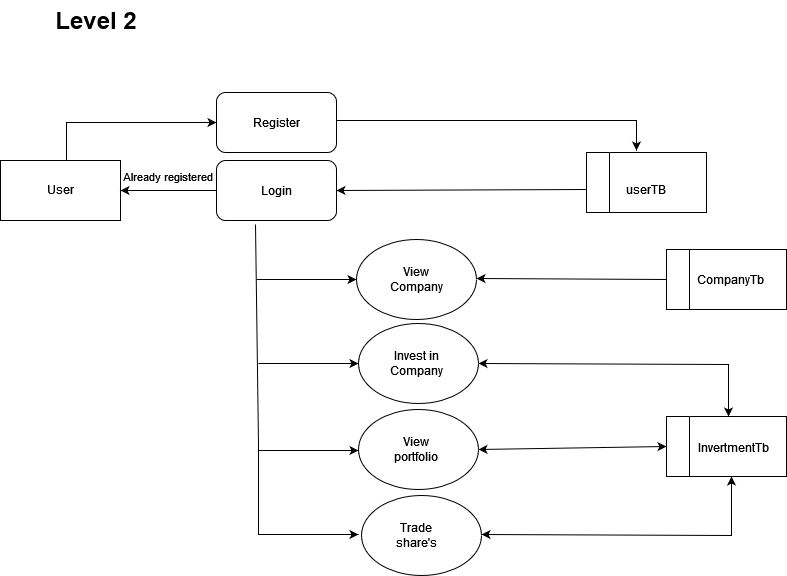
In Akorn there are mainly three level of Data flow diagrams which interact with data mentioned below in different levels .

****

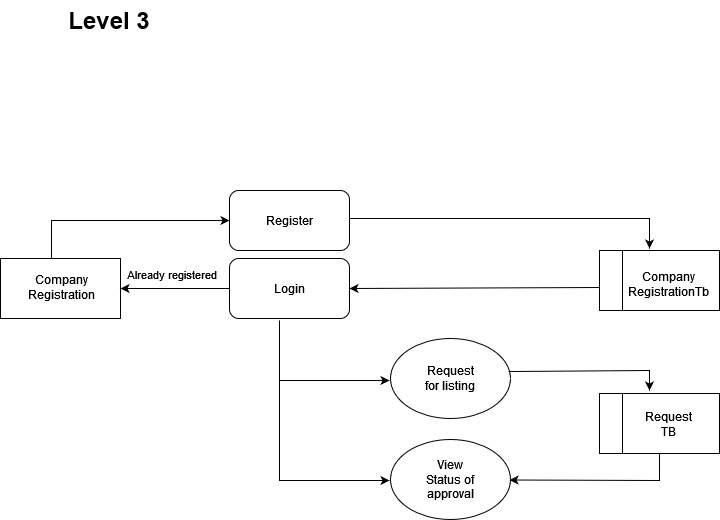
**Figure 1 : DFD Level 0**

****

**Figure 2 : DFD Level 1**

****

**Figure 3 : DFD Level 2**

****

**Figure 4 : DFD Level 3**

**Agile Methodology for Akorn**

Akorn will adopt an Agile development methodology for the following reasons:

**Rapid Prototyping and Iterative Development**

Agile allows Akorn to create functional prototypes early in the development process. These prototypes can be showcased to stakeholders and user groups for feedback, facilitating iterative development and continuous improvement. This ensures that the final platform aligns closely with user needs and expectations.

**Flexibility and Adaptability to Change**

Investment platform requirements and market dynamics can evolve during development. Agile's iterative approach allows Akorn to incorporate changes efficiently, adapting to new requirements through incremental updates and continuous stakeholder feedback.

**Improved Communication and Collaboration**

Agile promotes close collaboration between Akorn's development team, stakeholders, and potential users. This fosters transparency in the development process, ensures alignment with project goals, and allows for regular feedback loops. This collaboration is crucial for refining features and enhancing platform usability.

**Faster Time-to-Market**

Agile's focus on incremental delivery enables Akorn to release functional features early and often throughout the development process. Stakeholders can see tangible progress, provide timely feedback, and influence the direction of subsequent iterations, leading to a faster time-to-market for the platform.

**Agile Scrum Framework for Akorn**

Akorn will utilize Agile Scrum framework due to its suitability for complex projects like an investment platform:

**Key Elements of Agile Scrum**

* + **Short Iterations (Sprints):** Sprints in Scrum are time-boxed iterations lasting 1-4 weeks. Each sprint focuses on delivering a set of prioritized features or functionalities, allowing Akorn to demonstrate tangible progress and receive feedback early in the development cycle.
  + **Product Backlog:** A prioritized list of features and tasks maintained by the Product Owner (representing stakeholders). Each sprint, Akorn's team selects items from the backlog to work on, ensuring alignment with project goals and stakeholder priorities.

**Roles:**

* + **Product Owner:** Represents stakeholders, prioritizes backlog items, and ensures the team delivers value aligned with business objectives.
  + **Scrum Master:** Facilitates Scrum ceremonies (e.g., sprint planning, daily stand-ups, sprint reviews, retrospectives), removes impediments, and ensures adherence to Agile principles.
  + **Development Team:** Cross-functional team members responsible for delivering working software. They collaborate closely during sprints to achieve sprint goals.

**Sprint Planning and Retrospective:**

* + **Sprint Planning:** Held at the beginning of each sprint to define sprint goals, select backlog items, and create a sprint backlog.
  + **Sprint Retrospective:** Held at the end of each sprint to review what went well, identify areas for improvement, and adjust processes for the next sprint.

**Reasons for Choosing Agile Scrum for Akorn**

* + **Project Complexity:** Akorn's investment platform involves multiple functionalities and dynamic user interactions. Scrum's iterative approach allows Akorn to break down complex tasks into manageable increments, ensuring continuous progress and adaptability.
  + **Evolving Requirements:** The investment landscape and user expectations may evolve. Scrum's flexibility enables Akorn to respond to changing requirements quickly, adjusting priorities and delivering incremental value with each sprint.
  + **Stakeholder Collaboration:** Akorn values stakeholder feedback, including creators and backers. Scrum's emphasis on collaboration and regular feedback loops ensures that Akorn incorporates user insights throughout development, enhancing platform usability and satisfaction.

**Database Design Considerations: MongoDB for Akorn**

Akorn will utilize MongoDB, a NoSQL database, for its flexibility, scalability, and performance characteristics:

**MongoDB Features**

* + **Document-Oriented Storage:** MongoDB stores data in flexible JSON-like documents, accommodating complex data structures with nested objects and arrays. This flexibility is ideal for storing diverse data types related to projects, users, transactions, and multimedia content on Akorn.
  + **Schema-Less Design:** Unlike traditional relational databases, MongoDB imposes fewer constraints on schema design. Documents within a collection can have varying structures, enabling Akorn to evolve data models and adapt to changing business requirements without downtime or schema migrations.
  + **Horizontal Scalability:** MongoDB scales horizontally by distributing data across multiple servers (sharding), accommodating Akorn's growing user base and increasing data volumes. This scalability ensures high availability and performance under heavy load conditions.

**Benefits of MongoDB for Akorn**

* + **Flexibility:** MongoDB's document model allows Akorn to store and retrieve complex data structures naturally, supporting dynamic project profiles, multimedia content, and diverse user interactions.
  + **Scalability:** As Akorn expands its user base and project submissions, MongoDB's horizontal scalability ensures the platform can handle increasing data throughput and concurrent user interactions efficiently.
  + **Performance:** MongoDB's optimized query engine and indexing capabilities provide fast read and write operations, essential for delivering responsive user experiences and real-time updates on the Akorn platform.

By adopting Agile Scrum methodology and leveraging MongoDB's capabilities, Akorn can effectively develop and deploy its innovative investment platform, ensuring rapid iteration, stakeholder collaboration, and scalability to meet evolving market demands. This approach positions Akorn to deliver a robust, user-centric platform that fosters creativity, collaboration, and transformative project funding experiences.

**Collections and Fields:**

**lotsData**

|  |  |
| --- | --- |
| companyId (FK) | References companyregistration.\_id |
| companyName | Name of the company the user invested in |
| Email | User's email (redundant, consider removing) |
| Lots | Number of investment lots purchased by the user |
| amountInvested | Total amount invested by the user (default: 0) |
| createdAt | Timestamp of investment record creation |
| updatedAt | Timestamp of investment record update |

**CompanyRegistration**

|  |  |
| --- | --- |
| Name | Name of the user / founder |
| Email | Email of the user / founder |
| linkedin\_founder | LinkedIn of the founder |
| linkedin\_company | LinkedIn of the company |
| Sector | Sector of the company |
| market\_cap | Market cap of the company |
| Address | Address of the company |
| pan\_number | Pan number of the company |
| Website | Website of the company |
| previous\_fundraising\_rounds\_discription | Previous fundraising round description |
| product\_discription | Product description |
| Traction | Traction |
| revenue\_and\_making | Revenue of the company |
| team\_size | Number of Employees |
| community\_fund\_raising\_reason | Reason for fund raising |
| where\_you\_learn\_about\_us | Where they found out about us |
| existing\_commitments | Is there any existing commitment for Founder |
| Pitch | Pitch as in video format |
| Financials | Financial Documents |
| amount\_expected\_to\_raise | Amount expected to raise |
| Excel | Financials in excel format |
| Status | Status of Approval |

**excelData**

|  |  |
| --- | --- |
| company\_id (FK) | References companyregistration.\_id |
| quaters | Array of strings representing quarters |
| incomeExpenses | Array of objects with metric (string) and values (array of numbers) |

**Userkyc**

|  |  |
| --- | --- |
| name | User's name |
| email | User's email |
| linkedin\_url | User's LinkedIn profile URL |
| address | User's address |
| pan | User's PAN number |
| aadhar | User's Aadhar number |
| contact\_number | User's contact number |
| whattsapp | User's whatsapp number |
| about\_yourself | User's description |
| anual\_income | User's annual income |
| bank\_account\_number | User's bank account number |
| bank\_account\_photo | URL or path to bank account photo |
| where\_you\_learn\_about\_us | How the user found the platform |
| existing\_commitments | User's existing funding commitments |
| avatar | URL or path to user's avatar |
| ID\_proof | URL or path to user's ID proof |
| Status | User KYC verification status (default: 'null') |

**investmentData**

|  |  |
| --- | --- |
| companyId (FK) | References companyregistration.\_id |
| lots | Number of investment lots |
| amount | Total investment amount |
| companyName | Name of the company |
| companySector | Company's industry sector |
| email | User's email (redundant, consider removing) |
| timestamp | Timestamp of investment record creation (singular) |

**UserWallet**

|  |  |
| --- | --- |
| money | User's wallet balance |
| email | User's email (unique) |
| createdAt | Timestamp of wallet record creation |
| updatedAt | Timestamp of wallet record update |

**CompanyWithdrawals**

|  |  |
| --- | --- |
| companyId (FK) | References companyregistration.\_id (Likely) |
| amount | Withdrawal amount |
| email | User's email (redundant, consider removing) |
| status | Withdrawal request status (default: "null") |

8. SYSTEM DEVELOPMENT

Modularity Criteria and User Interface Layout

Designing the user interface (UI) layout for Akorn's platform involves several key considerations to ensure usability, engagement, and effectiveness. Here’s a structured approach to defining the UI layout:

Header

* + Logo: Clearly identifiable Akorn logo for brand recognition.
  + Navigation: Intuitive menu options

Hero Section

* + Eye-catching Banner: Featuring a compelling project or success story to attract attention.
  + Call-to-Action (CTA) Buttons: Prominently placed buttons for creators to start a project and for backers to explore projects.

Main Content Area

* + Project Categories: view of project categories for easy browsing.
  + Featured Projects: Highlighted projects, brief descriptions, funding progress, and funding goals.
  + Filters and Sorting Options: Enables users to filter projects and categories, etc.

Project Details Page

- Project Title and Description: Clearly defined with visuals and a compelling narrative.

Footer

- Legal Information: Terms of Service, Privacy Policy, and other legal disclosures.

Design Principles

* + Responsive Design: Ensures compatibility and usability across various devices (desktop, tablet, mobile).
  + Visual Consistency: Unified color scheme, typography, and iconography for a cohesive brand identity.
  + Intuitive Navigation: Seamless navigation flow with clear hierarchy and intuitive user interactions.

Users of the System

In the context of Akorn, "Users of the System" refers to the various stakeholders or individuals who interact with and utilize the Akorn platform. These users typically include:

* + Creators (Entrepreneurs or Visionaries):
  + Creators are individuals or teams who propose projects or business ideas on the Akorn platform. They use the system to create detailed project profiles, set funding goals, and interact with backers.
  + Backers and Users (Investors or Supporters):
  + Backers are individuals or organizations interested in supporting and funding projects proposed by creators. They browse projects on Akorn, pledge funds to support projects they find compelling, and may receive rewards based on their contribution level.
  + Administrators (Platform Managers):
  + Administrators are responsible for managing and overseeing the Akorn platform. They monitor project submissions, ensure compliance with platform policies, handle user support issues, and maintain the overall functionality and security of the system.

9. SYSTEM TESTING

System Testing

This outlines the testing phase for the Akorn platform, encompassing test plans, unit testing, integration testing, system testing, and implementation changeover plans.

**1. Test Plans**

The test plan for Akorn focuses on validating the functionality, performance, and reliability of the platform. It includes:

**Testing Objectives:** Ensuring all features meet specified requirements.

**Testing Scope:** Covering user authentication, project management and user interactions.

**Testing Schedule:** Milestones for unit testing, integration testing, and system testing phases.

**Risk Assessment:** Identifying potential risks and mitigation strategies during testing.

**2. Unit Testing**

**a. Test Items (Test Cases)**

**Backend Unit Testing:**

**User Authentication:**

Test case: Verify login with valid credentials.

Test case: Verify token expiration handling.

**Project Management:**

Test case: Create a new project and validate database entry.

Test case: Update project details and verify database update.

**Frontend Unit Testing:**

**User Interface Components:**

Test case: Validate form inputs for project creation.

Test case: Test UI rendering for project listings and user profiles.

**3. Integration Testing**

Integration testing ensures that components of the Akorn platform work together seamlessly:

**Backend Integration Testing:**

Test interaction between authentication middleware and API endpoints.

**Frontend Integration Testing:**

Validate interaction between React components.

Test asynchronous behavior of Axios requests for data fetching and updating.

**4. System Testing**

**a. Test Items (Test Cases)**

System testing validates the overall functionality and performance of the Akorn platform:

**User Workflow Testing:**

Test user registration, login, and logout processes.

Test end-to-end flow from project creation to funding and update notifications.

**Performance Testing:**

Assess platform responsiveness under varying user loads.

Test database query performance during peak traffic times.

**Security Testing:**

Verify encryption of sensitive user data (password).

Test for vulnerabilities in authentication mechanisms and API endpoints.

10. CONCLUSION

Conclusion

The Akorn platform emerges not only as a technological innovation but also as a game changer when it comes to investment platform. Akorn is built using the MERN stack – MongoDB, Express.js, React, and Node.js and aims to change how creators interact with their backers in the investment space. In all its stages of development and implementation, Akorn has been driven by one vision: promoting transparency, nurturing creativity and ensuring a smooth user experience.

**Reflection on Objectives and Achievements**

From its early days, Akorn has sought to overcome limitations inherent in other platforms that provide crowd funding services by concentrating on achieving specific goals: creating secure authentication schemes and optimizing user interaction through smart UI/UX design. However, these goals were not merely achieved but exceeded with JWT used for safe user login being implemented successfully resulted in methods which are trouble free. The intuitive user interface that uses React sets new standards of user experience in this form of investment thus allowing any person irrespective of her/his background will be able to manage or create projects easily.

11. IMPACT AND FUTURE ENHANCEMENTS

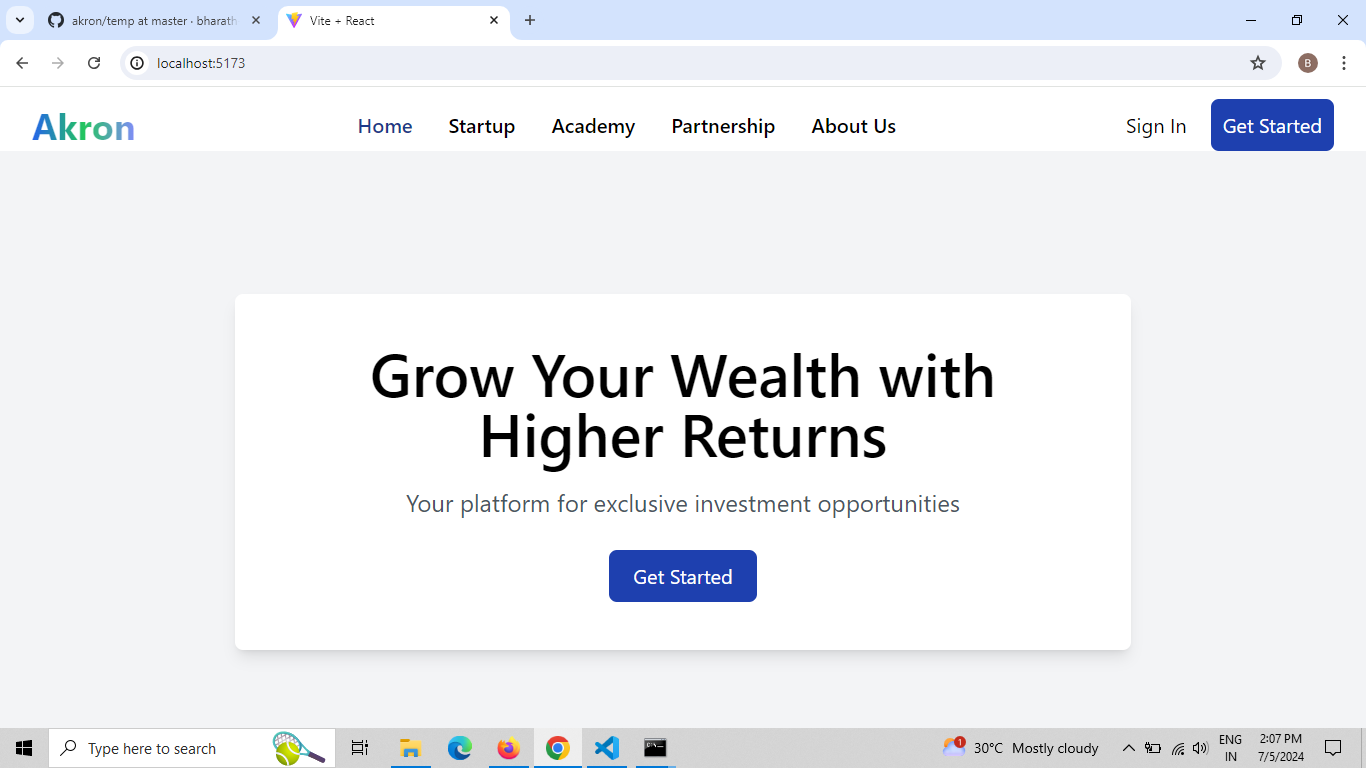
Impact and Future Enhancements

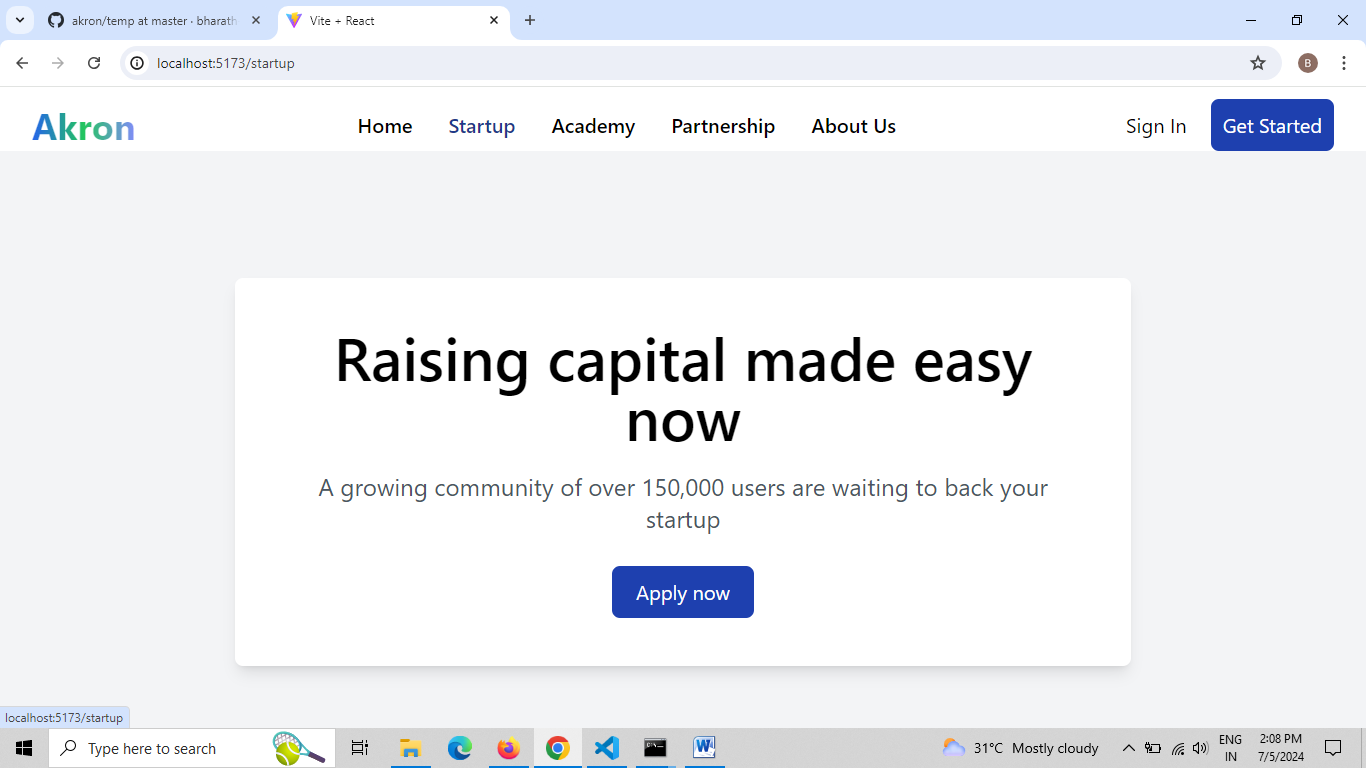
Akorn has the potential to significantly improve Investment platforms and efficiency. As the platform matures, future enhancements can further strengthen its impact:

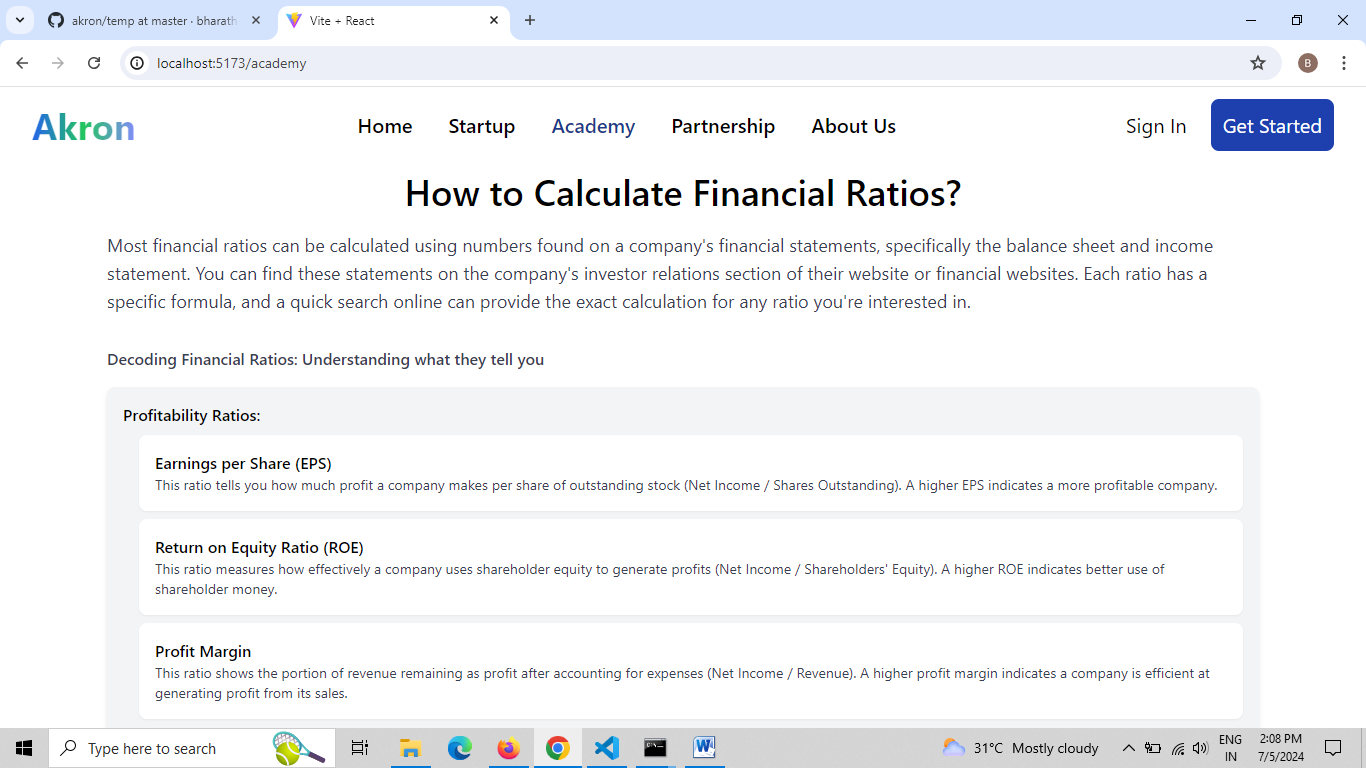
* **Integration with NSE and BSE Data Sources:** Direct integration with NSE and BSE databases can automate project data population, reducing manual input and improving data accuracy.
* **Advanced Analytics and Reporting:** Implementing data visualization tools and machine learning algorithms can provide deeper insights into project trends and risks.
* **Multilingual Support:** Expanding language support can cater to a wider audience and ensure inclusivity for citizens across India's diverse linguistic landscape.
* **Mobile App Development:** Developing a mobile application can increase accessibility and allow users to actively track and Invest.

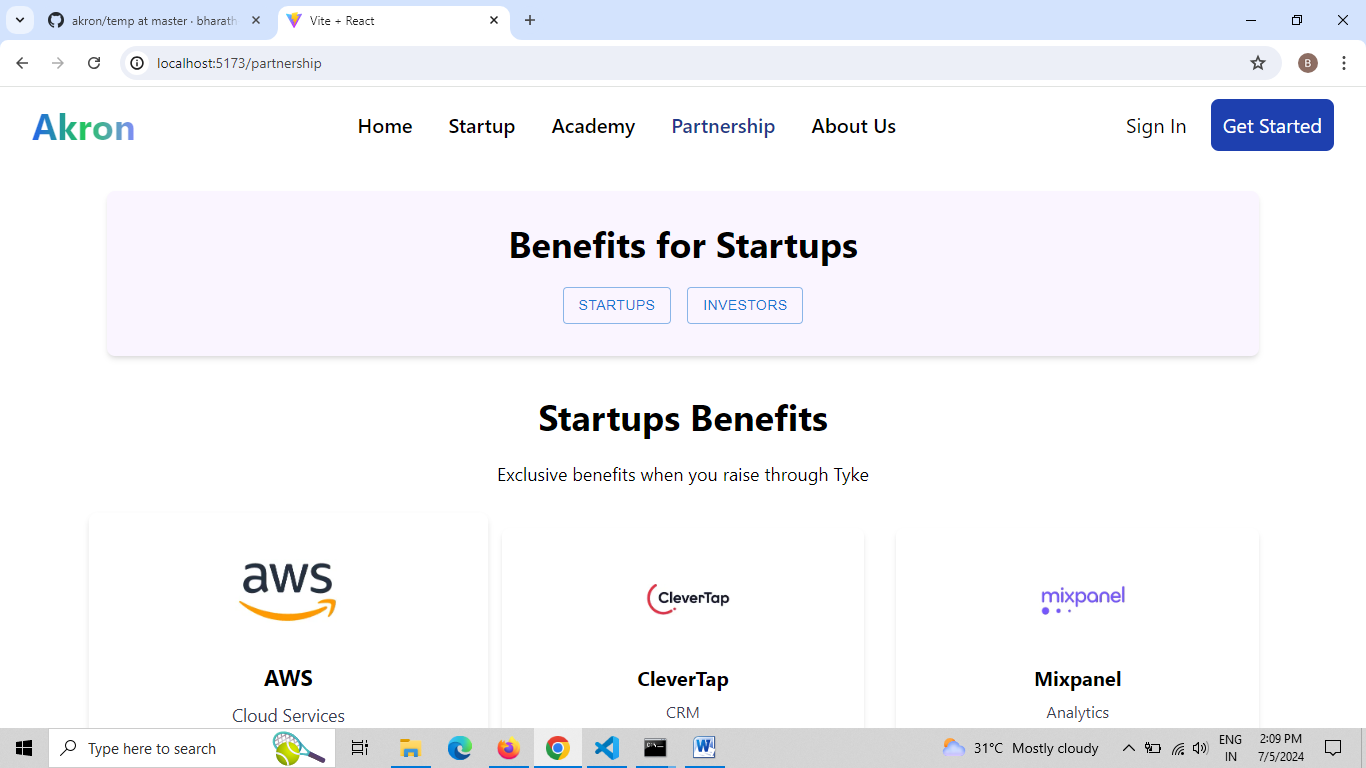
12. SCREENSHOTS

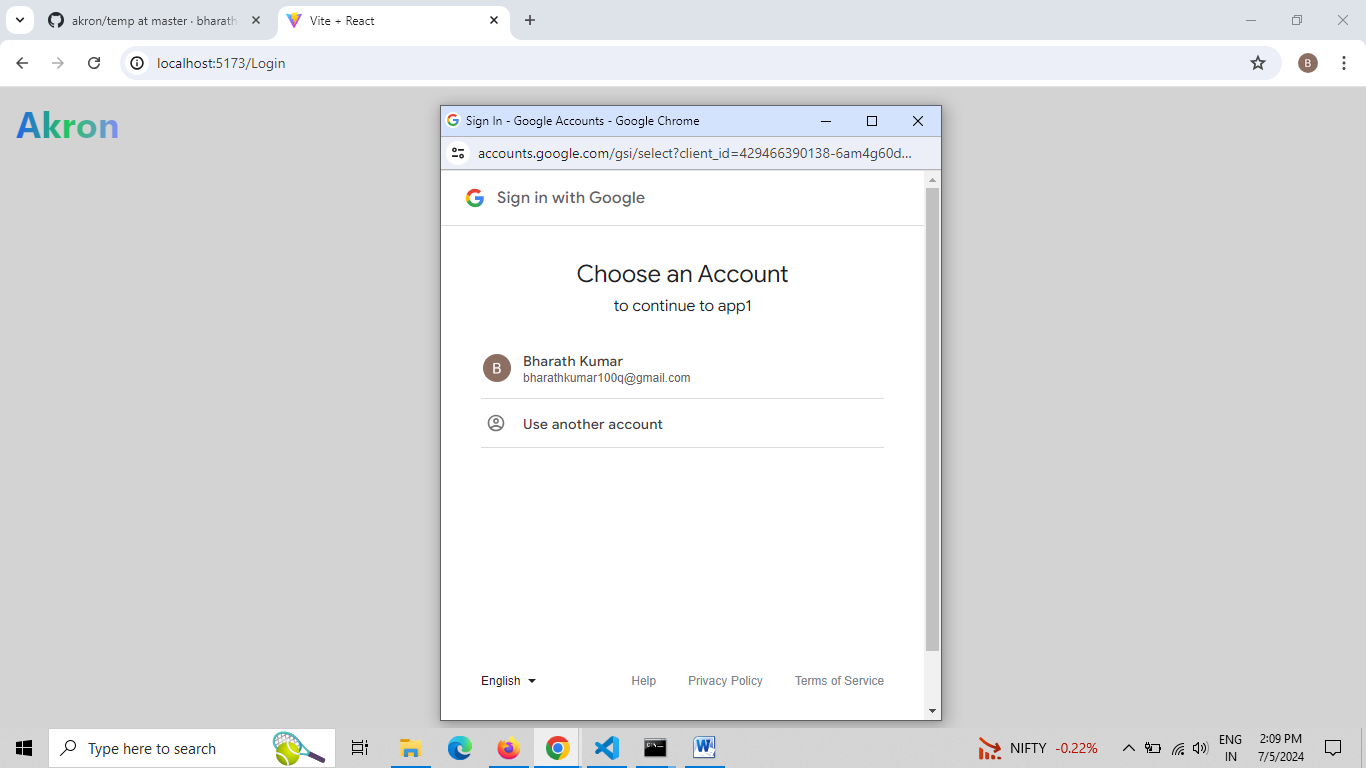
Screenshots

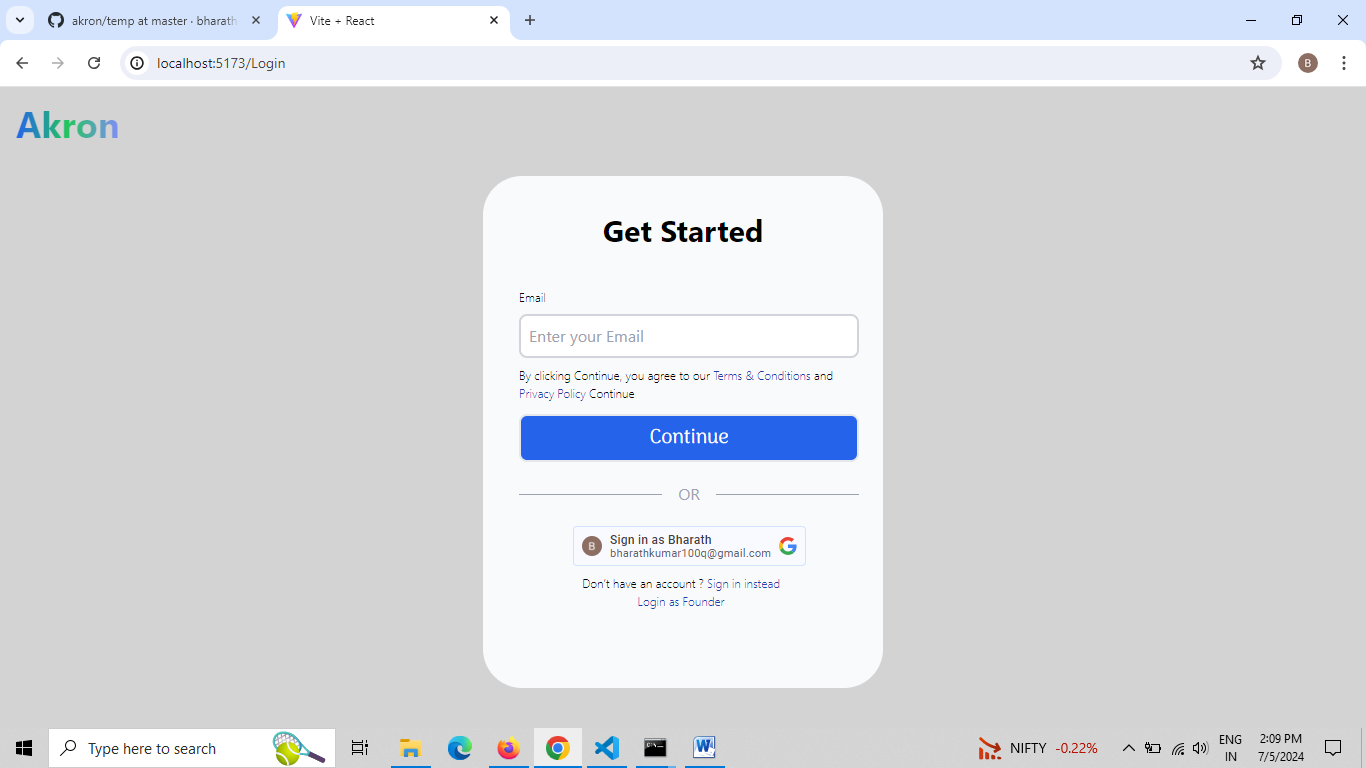


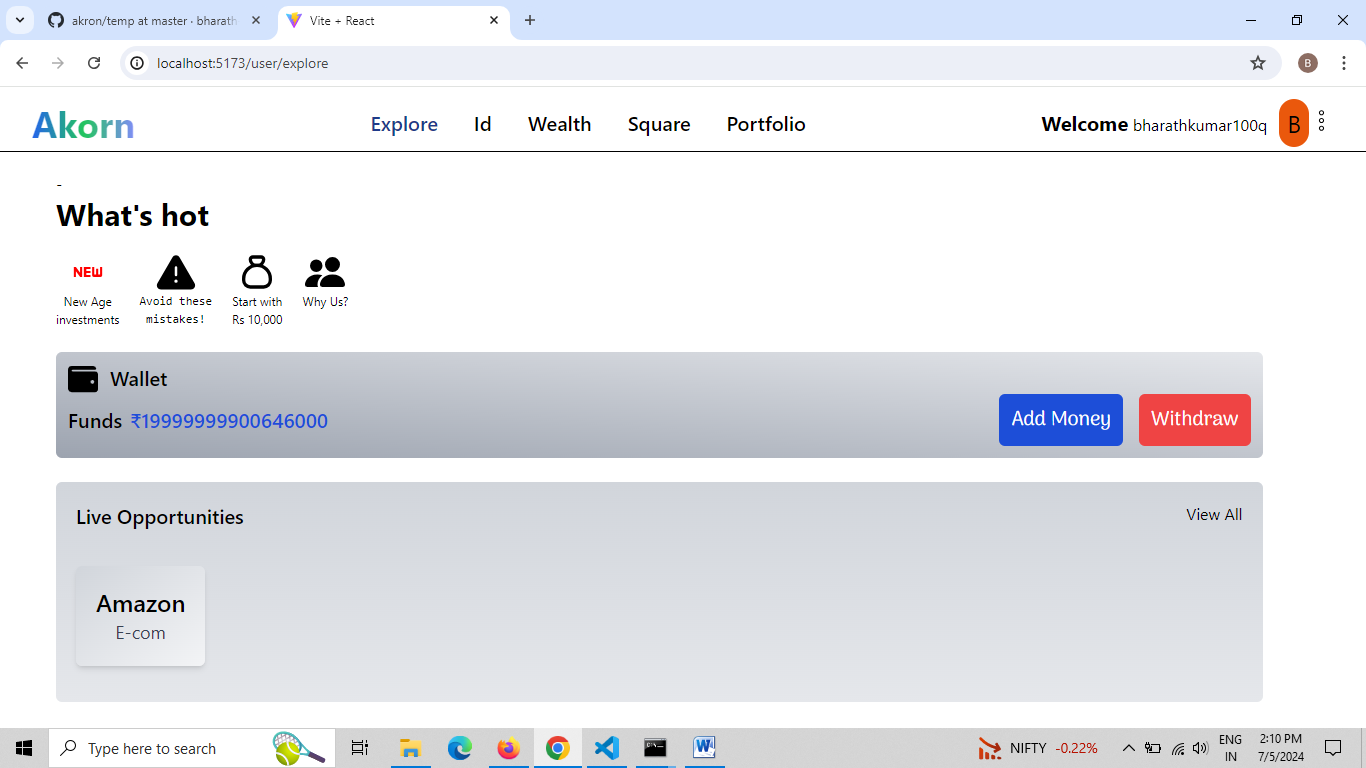


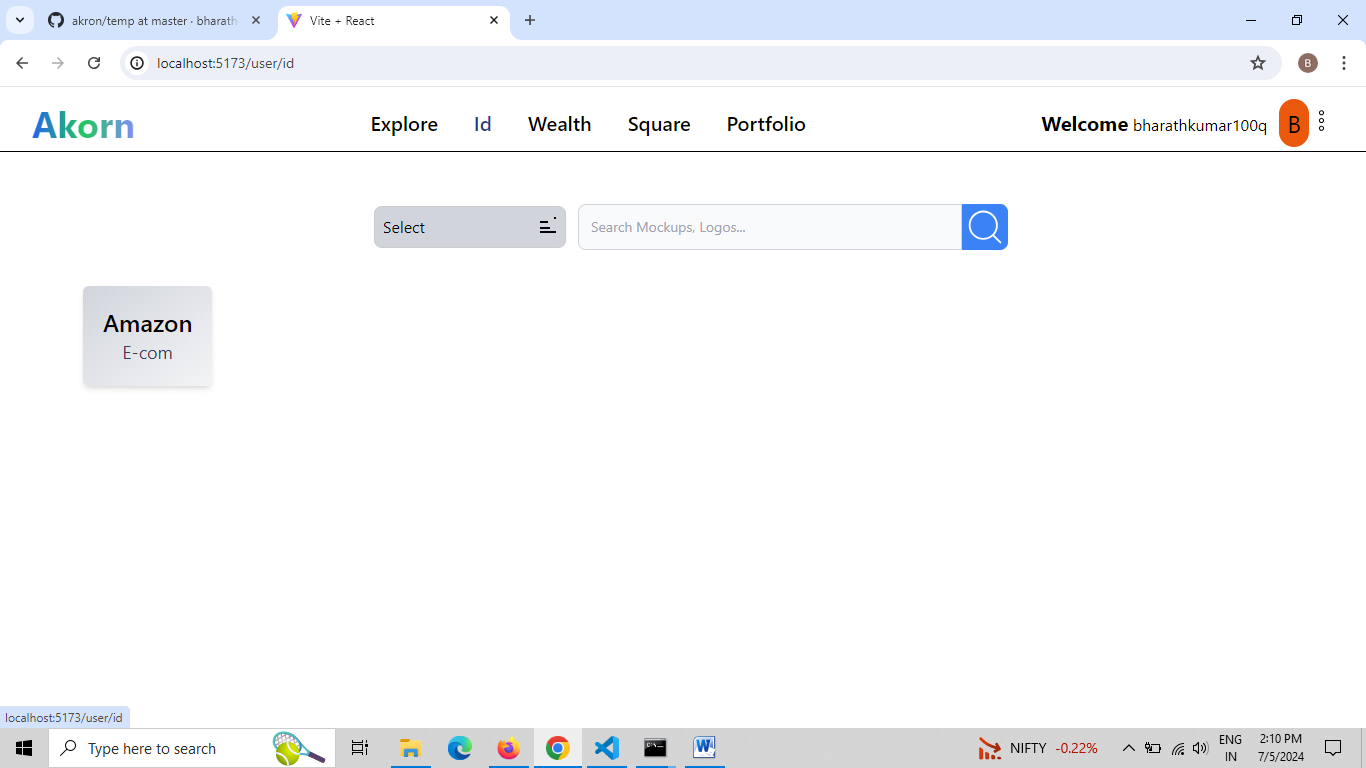


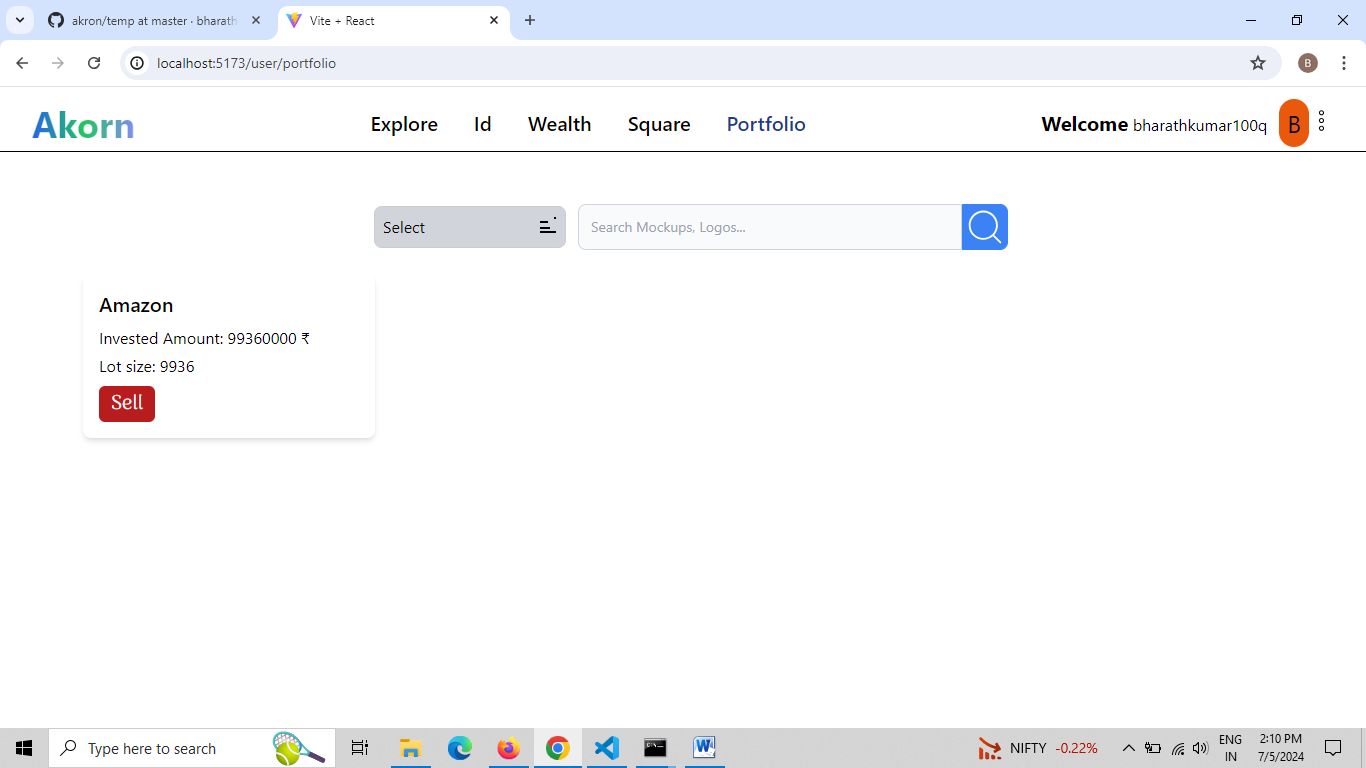












13. REFERENCES

References

* "React - A JavaScript library for building user interfaces" by [react.dev](https://reactjs.org/) , The official React documentation is a valuable resource for learning about this popular JavaScript library for building user interfaces.
* "Recharts - React Chart Library" by Recharts [https://recharts.org/en-US/](https://recharts.org/en-US/%20) , The Recharts documentation explains how to use this library to create interactive charts and graphs within React applications.