

Akorn

PROJECT REPORT

Submitted by
BHARATH KUMAR M (Reg no: MJAWMCA012)

In Partial Fulfillment of requirements For the award of the Degree of
MASTER OF COMPUTER APPLICATION (MCA)



CENTRE FOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
University Of Calicut MANJERI JULY 2023

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 theproject with their constructive criticisms.

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.

Date:
Place:

Signature:
Name. : **BHARATH KUMAR M**
Reg. No.: MJAWMCA012

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.

Signature
Mr. Rinshad . K
Assistance Professor

Signature
Ms. Shalini NP
Associate Coordinator

Place:
Date:

PROJECT EVALUATION REPORT OF THE EXAMINERS

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

Examiners:

- 1.
- 2.

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 CONTENT

INTRODUCTION.....	8
PROBLEM DEFINITION AND METHODOLOGY.....	9
1. Problem Definition	9
2. Objectives	9
3. Motivation	9
4. Methodology	9
5. Scope	9
Feasibility Study.....	10
1. Technical Feasibility	10
2. Economic Feasibility	10
3. Operational Feasibility	10
4. Legal Feasibility:	10
Requirement Analysis and Specification	11
1. Requirement Analysis	11
2. Existing System	11
3. Proposed System	11
4. Requirement Specification	11
5. Feasibility Study.....	12
Modularity Criteria and User Interface Layout	13
Header	13
Hero Section	13
Main Content Area	13
Project Details Page.....	13
Footer.....	13
Design Principles.....	13
Users of the System.....	13
1. Creators (Entrepreneurs or Visionaries):	13
2. Backers and Users (Investors or Supporters):	13
3. Administrators (Platform Managers):.....	13
Architecture Diagrams	14
a. DFD	14
b. Database Design	17
IMPLEMENTATION	21
1. Brief Description about the Tools/Scripts for Implementation	21
2. Module Hierarchy.....	21

3. Coding	21
4. Problems Encountered.....	22
5. System Design.....	22
1. Agile Manifesto Principles:	22
2. Agile Practices and Akorn Implementation:	22
3. Agile Implementation Benefits for Akorn:	23
4. Challenges and Mitigation:	23
TESTING.....	24
1. Test Plans	24
2. Unit Testing.....	24
3. Integration Testing.....	24
4. System Testing	24
CONCLUSION	25
BIBLIOGRAPHY	26

LIST OF FIGURES

Figure 1 : DFD Level 0.....	14
Figure 2 : DFD Level 2.....	15
Figure 3 : DFD Level 3.....	15
Figure 4 : DFD Level 4.....	16

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.

PROBLEM DEFINITION AND METHODOLOGY

1. 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.

2. Objectives

The primary objectives of the feasibility study for the Akorn project are:

- Assessing the technical feasibility of developing the Akorn platform, including infrastructure requirements, scalability, and integration capabilities with API's.
- Conducting an economic analysis to determine the financial viability of the project, including initial setup costs, revenue streams, and sustainability measures.
- Evaluating operational feasibility by examining user experience, administrative processes, and platform management strategies.

3. Motivation

The motivation behind launching Akorn lies in revolutionizing the investment landscape and to find investors as soon as possible

Akorn aims to unlock new possibilities for both creators and backers, facilitating the realization of innovative ideas through collective support and collaboration.

4. Methodology

The feasibility study will employ the following methodology:

- **Technical Feasibility:** Assessing the capability of integrating API's, infrastructure and developing a scalable platform with optimal performance and security.
- **Economic Feasibility:** Conducting a cost-benefit analysis to determine financial viability, revenue generation potential, and sustainability through transaction fees and premium features.
- **Operational Feasibility:** Evaluating user experience design, administrative processes, and backend systems to ensure efficient platform operation and management.
- **Legal Feasibility:** Ensuring compliance with regulatory requirements related to data protection, financial regulations, and intellectual property rights across jurisdictions.

5. Scope

The feasibility study will focus on:

- Analyzing the technical requirements and capabilities necessary for developing and maintaining the Akorn platform.
- Evaluating operational feasibility by examining user interface design, project management tools, and community engagement features.
- Addressing legal considerations to ensure compliance with relevant laws and regulations governing investment platforms.

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 management.
- **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.

Requirement Analysis and Specification

1. Requirement Analysis

The requirement analysis for the Akorn project involves assessing the needs and expectations for an 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.

2. Existing System

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

3. Proposed System

The proposed Akorn system aims to revolutionize investment by leveraging the MERN stack's capabilities:

- **MongoDB:** Provides a scalable and flexible NoSQL database solution for storing diverse data types related to projects, users, transactions, and comments.
- **Express.js:** Offers a robust backend framework for building RESTful APIs to manage project creation, user authentication, transaction handling, and communication between the client and server.
- **React:** Facilitates dynamic and interactive user interfaces with reusable components, enhancing user experience and engagement.
- **Node.js:** Powers the backend server environment, enabling efficient handling of concurrent requests, real-time updates, and integration with external APIs.

4. Requirement Specification

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.

c. Environmental Details (Hardware & Software Requirements)

- **Hardware:** Standard servers or cloud-based infrastructure to host MongoDB, Node.js server, and frontend React application.
- **Software:** Operating systems compatible with Node.js (e.g., Linux, Windows), MongoDB database management system, and development tools like Visual Studio Code for coding.

5. Feasibility Study

a. Technical Feasibility

Assessing the technical feasibility involves evaluating the capability of the MERN stack to meet project requirements, including platform scalability, integration with external APIs and deployment on chosen infrastructure (cloud or dedicated servers).

b. Economical Feasibility

Conducting a cost-benefit analysis to determine initial setup costs (development, infrastructure), ongoing operational expenses (maintenance, server costs), revenue generation through transaction fees, and profitability projections based on market demand and growth potential.

c. Operational Feasibility

Evaluating the operational feasibility by analyzing user experience design, administrative processes (account management, project submission review), and backend system management (database administration, server monitoring) to ensure efficient platform operation and management.

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

- Contact Information: Links to contact support, FAQs, and customer service information.
- 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:

1. 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.

2. 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.

3. 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.

Architecture Diagrams

a. DFD

Level 0

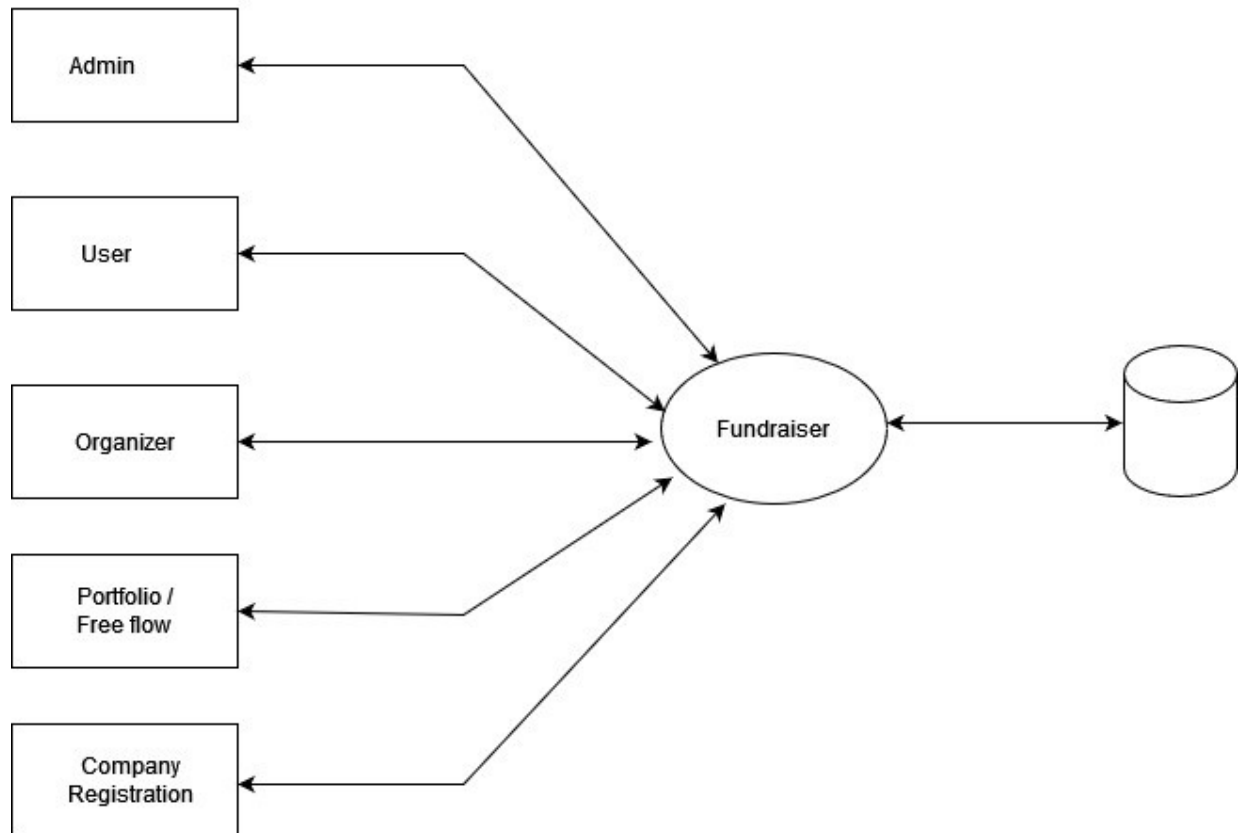
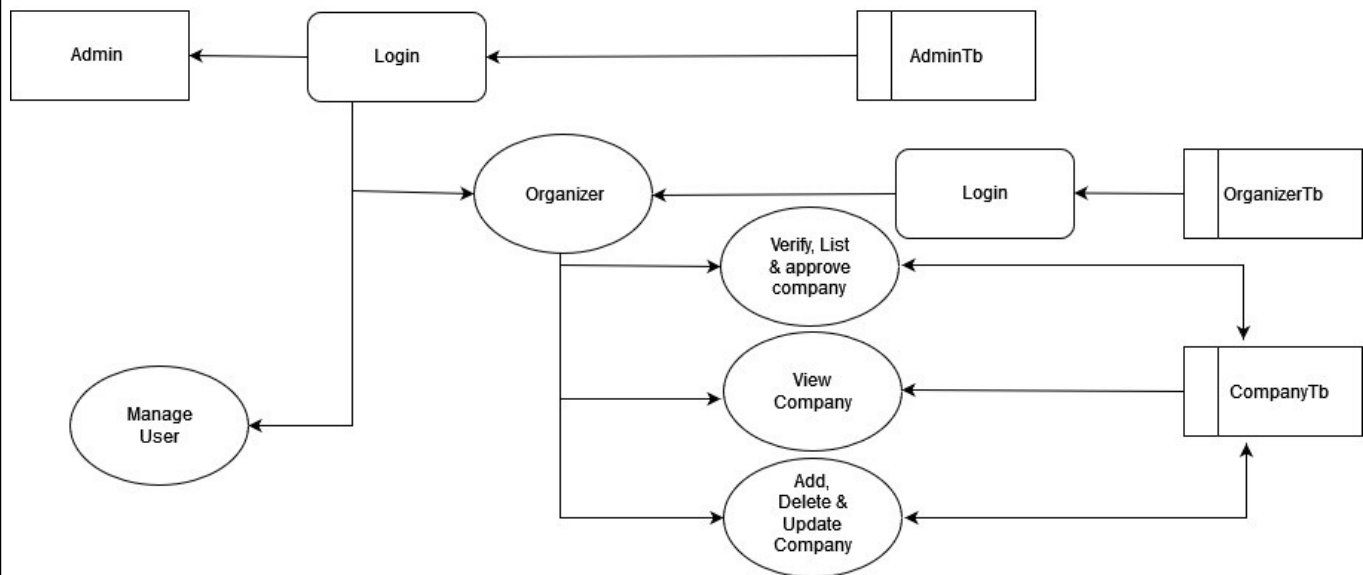
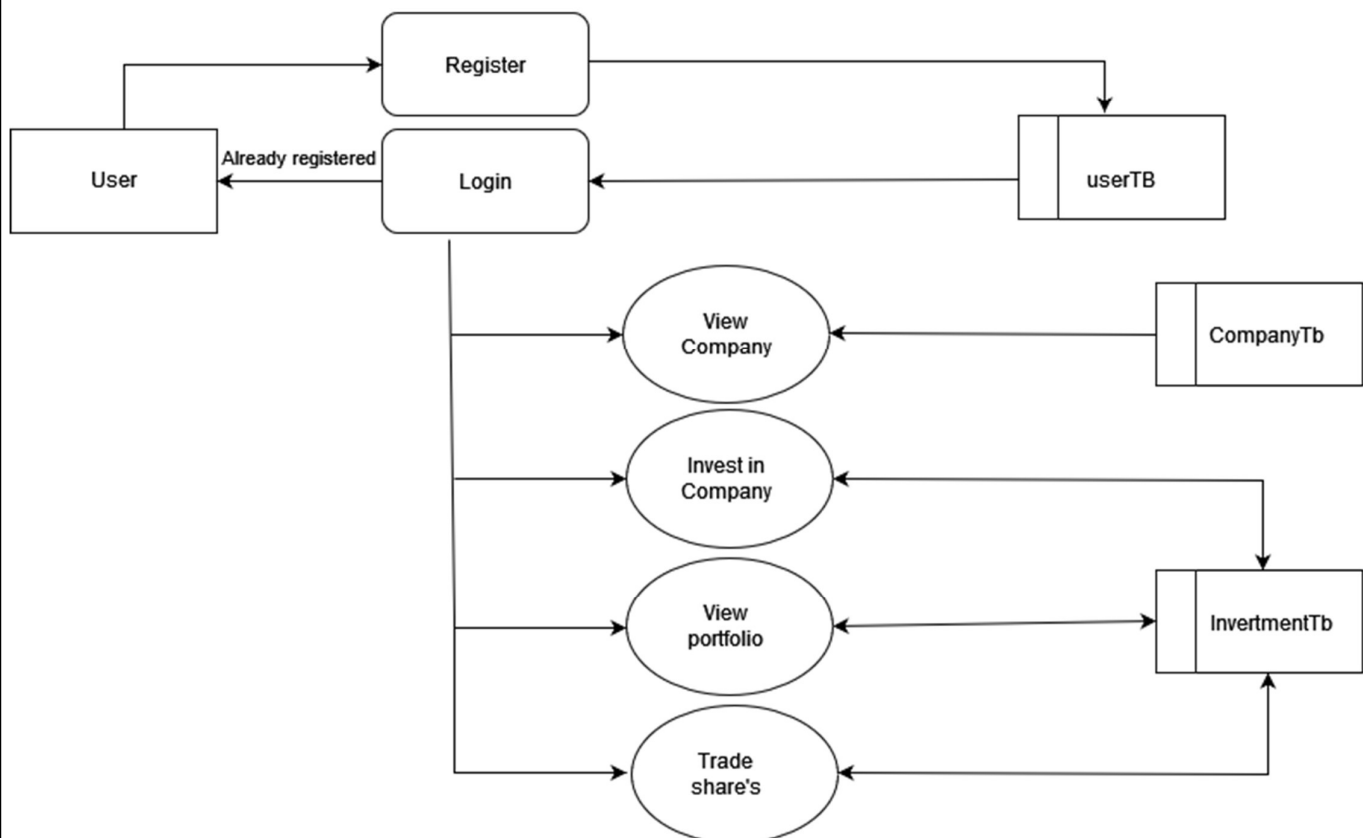


Figure 1 : DFD Level 0

Level 1



Level 2



Level 3

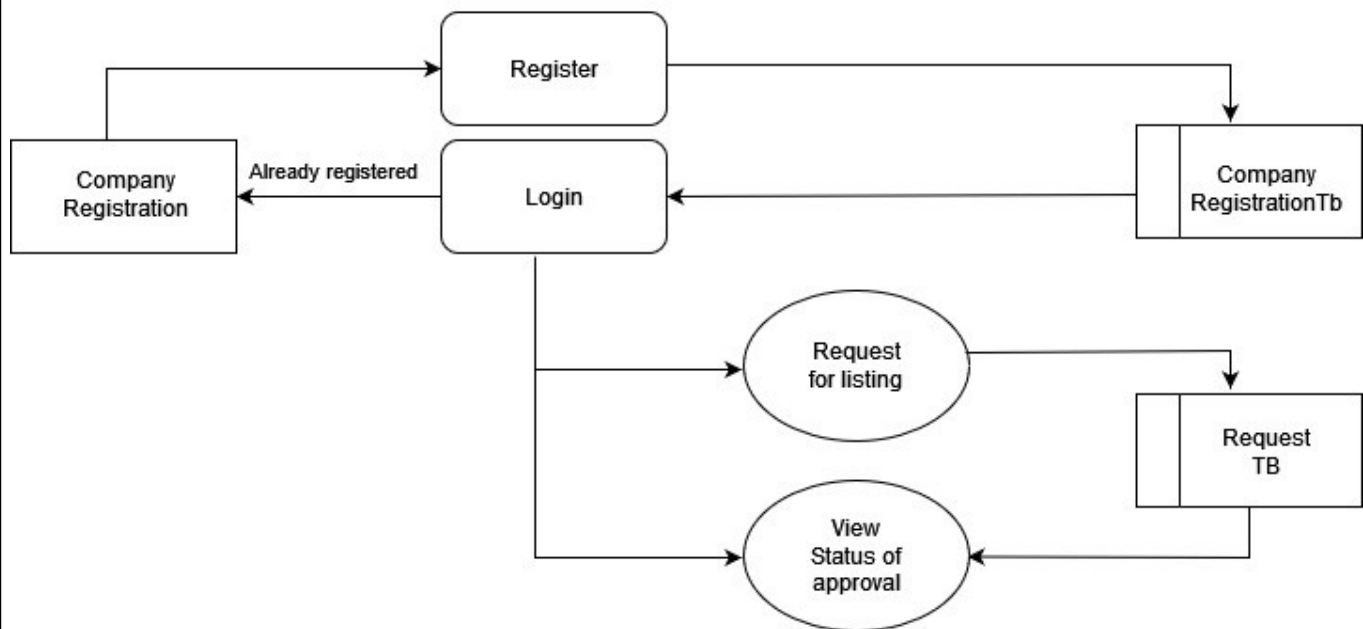


Figure 4 : DFD Level 4

b. Database Design

MongoDB is a popular NoSQL database known for its flexibility, scalability, and performance. It diverges from traditional relational databases by using a document-oriented data model, where data is stored in flexible, JSON-like documents instead of tables and rows. Here are some key points about MongoDB:

1. Document-Oriented: MongoDB stores data in JSON-like documents called BSON (Binary JSON), which allows for nested structures and flexible schemas. This makes it easy to represent hierarchical relationships and complex data types.
2. Schema-less: Unlike traditional relational databases, MongoDB does not require a predefined schema. Fields can vary across documents within a collection, which simplifies data management and allows for agile development.
3. Scalability: MongoDB is designed to scale horizontally across multiple servers, making it suitable for large-scale applications and high-volume data storage. It supports automatic sharding, which distributes data across different machines for improved performance.
4. High Performance: MongoDB's architecture is optimized for both read and write operations. It uses internal memory mapping for fast data access and supports indexes to improve query performance.
5. Querying: MongoDB supports rich query expressions and a flexible aggregation framework, allowing for complex queries and data aggregation operations.
6. Replication and High Availability: MongoDB provides built-in replication with automatic failover, ensuring data redundancy and high availability. This is critical for maintaining uptime and data durability.
7. Community and Ecosystem: MongoDB has a large and active community of developers, offering extensive documentation, tutorials, and support resources. It integrates well with many programming languages and frameworks.
8. Use Cases: MongoDB is widely used in a variety of applications, including content management systems, real-time analytics, e-commerce platforms, mobile applications, and IoT (Internet of Things) solutions.

lotsData

Column Name	
companyId (FK)	References companyregistration._id (Likely)
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

Column Name	Data Type
name	VARCHAR(255)
email	VARCHAR(255)
linkedin_founder	VARCHAR(255)
linkedin_company	VARCHAR(255)
sector	VARCHAR(255)
market_cap	VARCHAR(255)
address	TEXT
pan_number	VARCHAR(255)
website	VARCHAR(255)
previous_fundraising_rounds_discription	TEXT
product_discription	TEXT
traction	TEXT
revenue_and_making	TEXT
team_size	INT
community_fund_raising_reason	TEXT
where_you_learn_about_us	TEXT
existing_commitments	TEXT
pitch	TEXT
financials	TEXT
amount_expected_to_raise	DECIMAL
excel	VARCHAR(255)
status	BOOLEAN
lots	INT
lotsOriginal	INT

excelData

Column Name	
company_id (FK)	References companyregistration._id (Likely)
quarters	Array of strings representing quarters
incomeExpenses	Array of objects with metric (string) and values (array of numbers)

Userkyc

Column Name	
name	User's name
email	User's email (unique)
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
whatsapp	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')
createdAt	Timestamp of KYC document creation
updatedAt	Timestamp of KYC document update

investmentData

Column Name	
companyId (FK)	References companyregistration._id (Likely)
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

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

CompanyWithdrawals

Column Name	
companyId (FK)	References companyregistration._id (Likely)
amount	Withdrawal amount
email	User's email (redundant, consider removing)
status	Withdrawal request status (default: "null")
createdAt	Timestamp of withdrawal request creation
updatedAt	Timestamp of withdrawal request update

IMPLEMENTATION

1. Brief Description about the Tools/Scripts for Implementation

For developing the Akorn platform on the MERN stack, the following tools and scripts were utilized:

- **Tools:**
 - Visual Studio Code (VSCode): Integrated Development Environment (IDE) for writing and debugging code.
 - Node.js: JavaScript runtime for server-side execution.
 - Express.js: Backend web application framework for Node.js.
 - React: Frontend JavaScript library for building user interfaces.
 - MongoDB: NoSQL database for storing project data, user information, and transaction details.
 - Mongoose: Object Data Modeling (ODM) library for MongoDB and Node.js.
 - Axios: HTTP client for making API requests from the frontend to the backend.
 - JWT (JSON Web Tokens): Used for authentication and authorization.
- **Scripts:**
 - npm scripts for running development and production servers.

2. Module Hierarchy

The Akorn platform is structured into several modules to ensure modularity and maintainability:

- **Backend Modules:**
 - User Authentication and Authorization
 - Project Management
 - Database Management (MongoDB with Mongoose)
 - API Routes (Express.js)
- **Frontend Modules:**
 - User Authentication (Login, Registration)
 - Project Creation and Management
 - Backer Interaction and Engagement
 - Responsive UI Components (React)

3. Coding

Key components of the Akorn platform implementation include:

- **User Authentication:**
 - Implemented using JWT for secure token-based authentication.
 - Protected routes in Express.js to ensure authorized access to specific endpoints.
- **Project Management:**
 - CRUD operations for projects, including creation, updating, deletion, and retrieval.
 - Integration with MongoDB to store project details, images, videos, and funding information.
- **Frontend Development:**
 - React components for project creation forms, project listings, and backer interactions.
 - Redux for managing application state and handling asynchronous actions.
 - Axios for making API requests to the backend and updating UI components dynamically.

4. Problems Encountered

During implementation, several challenges were encountered:

- **Complexity of MongoDB Schema Design:**
 - Addressed by carefully designing MongoDB schemas to accommodate nested data structures for projects, comments, and user profiles efficiently.
- **Deployment and Environment Configuration:**
 - Configuring environments (development, staging, production) and managing environment variables securely using dotenv and deployment scripts.
- **Handling Asynchronous Operations:**
 - Managed asynchronous operations using async/await in Node.js to ensure proper flow control and error handling, especially during database queries and API requests.

5. System Design

Agile Model for Akorn Platform

1. Agile Manifesto Principles:

- Individuals and interactions over processes and tools: Emphasize collaboration between developers, stakeholders, and users to ensure continuous feedback and alignment with project goals.
- Working software over comprehensive documentation: Prioritize delivering functional increments of the Akorn platform, focusing on user stories and iterative development.
- Customer collaboration over contract negotiation: Engage stakeholders and end-users throughout the development process to validate features and ensure they meet expectations.
- Responding to change over following a plan: Adapt the project scope and features based on continuous feedback, market trends, and emerging technologies.

2. Agile Practices and Akorn Implementation:

a. Scrum Framework:

- Roles:
 - Product Owner: Represents stakeholders and prioritizes features based on business value.
 - Scrum Master: Facilitates the team, removes impediments, and ensures adherence to Agile practices.
 - Development Team: Cross-functional team members responsible for implementing features.
- Artifacts:
 - Product Backlog: Prioritized list of features (user stories) based on stakeholder feedback and project goals.
 - Sprint Backlog: Features selected from the product backlog for implementation during a sprint.
 - Increment: Functional software developed within a sprint, potentially releasable.
- Events:
 - Sprint Planning: Define sprint goals and select user stories for implementation.
 - Daily Stand-ups: Daily meetings to synchronize activities, discuss progress, and identify any blockers.
 - Sprint Review: Demonstration of completed features to stakeholders for feedback.
 - Sprint Retrospective: Reflect on sprint outcomes, discuss what went well, what could be improved, and plan adjustments for future sprints.

b. Iterative Development:

- Implement features of the Akorn platform in short iterations (sprints), typically 1-4 weeks long, to deliver incremental value and gather feedback early.

c. Continuous Integration and Deployment (CI/CD):

- Automate testing and deployment processes to ensure new features are integrated smoothly into the existing platform.
- Conduct regular testing (unit tests, integration tests) to maintain code quality and ensure functionality meets requirements.

d. User-Centric Design:

- Prioritize user stories that enhance user experience (UX), such as intuitive navigation, responsive design, and personalized project recommendations.

e. Cross-Functional Teams:

- Foster collaboration between developers, designers, testers, and stakeholders to ensure a holistic approach to feature development and deployment.

3. Agile Implementation Benefits for Akorn:

- Flexibility: Ability to respond to changing market demands and user feedback.
- Transparency: Regular updates and demos foster transparency with stakeholders and users.
- Quality: Continuous testing and integration improve software quality and reliability.
- Customer Satisfaction: Early and frequent delivery of valuable features enhances user satisfaction and engagement.

4. Challenges and Mitigation:

- Scope Creep: Prioritize features based on business value and maintain a stable product backlog.
- Team Coordination: Daily stand-ups and sprint planning sessions ensure alignment and address any coordination issues promptly.
- Technical Debt: Regular refactoring and addressing technical debt during sprints to maintain code quality.

The Agile model enhances the development of the Akorn platform by emphasizing collaboration, iterative development, and responsiveness to user needs. By adopting Agile practices, Akorn can deliver a robust and user-centric platform that revolutionizes the experience.

This Agile model aligns with Akorn's goals of fostering innovation, enhancing user engagement, and ensuring transparency in project funding, making it well-suited for iterative and adaptive development.

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.

CONCLUSION

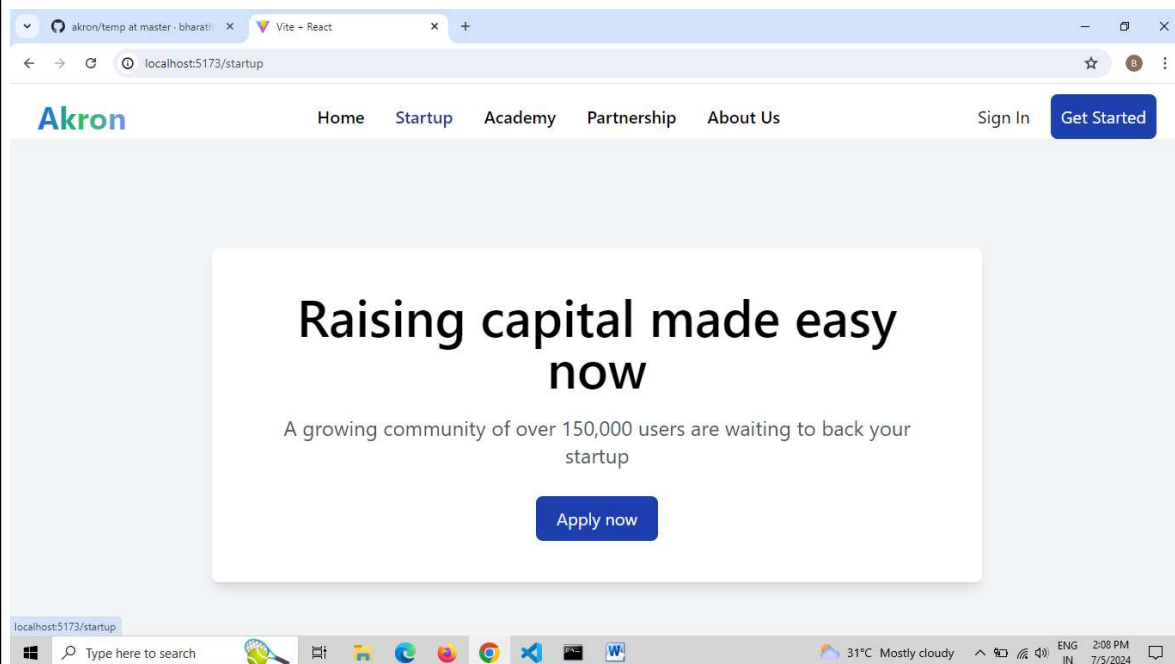
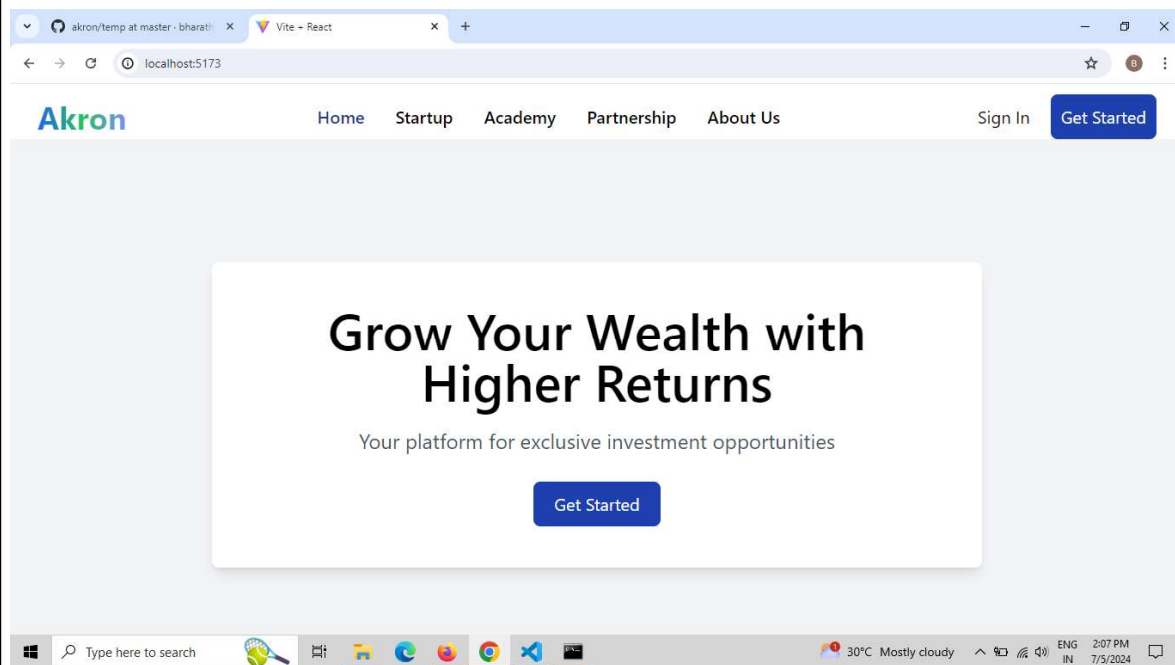
The Akorn platform emerges not only as a technological innovation but also as a game changer when it comes to investment platform. Akorn is built using the MERN stack – MongoDB, Express.js, React, and Node.js and aims to change how creators interact with their backers in the investment space. In all 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.

BIBLIOGRAPHY

- Patidar, S. (2020). **MERN Quick Start Guide: Build web applications with MongoDB, Express.js, React, and Node**. Birmingham, UK: Packt Publishing.
- Subramanian, V. (2019). **Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node**. Berkeley, CA: Apress.
- Rauch, R., & Tolley, C. (2017). **Fullstack React: The Complete Guide to ReactJS and Friends**. Self-published.
- Wilson, E., & Larson, C. (2018). **MongoDB: The Definitive Guide: Powerful and Scalable Data Storage**. Sebastopol, CA: O'Reilly Media.
- Hettiarachchi, S. (2019). **React Design Patterns and Best Practices: Build easy to scale modular applications using the most powerful components and design patterns**. Birmingham, UK: Packt Publishing.
- Banker, S. (2019). **MongoDB Applied Design Patterns: Practical Use Cases with the Leading NoSQL Database**. Berkeley, CA: Apress.



The screenshot shows a web browser window with the Akron website. The navigation bar includes Home, Startup, Academy, Partnership, and About Us, along with Sign In and a Get Started button. The main heading is "How to Calculate Financial Ratios?". Below it, a paragraph explains that most financial ratios are calculated using numbers from a company's financial statements (balance sheet and income statement). A sub-section titled "Decoding Financial Ratios: Understanding what they tell you" lists three profitability ratios:

- Earnings per Share (EPS)**: This ratio tells you how much profit a company makes per share of outstanding stock (Net Income / Shares Outstanding). A higher EPS indicates a more profitable company.
- Return on Equity Ratio (ROE)**: This ratio measures how effectively a company uses shareholder equity to generate profits (Net Income / Shareholders' Equity). A higher ROE indicates better use of shareholder money.
- Profit Margin**: This ratio shows the portion of revenue remaining as profit after accounting for expenses (Net Income / Revenue). A higher profit margin indicates a company is efficient at generating profit from its sales.

The Windows taskbar at the bottom shows the date as 7/5/2024 and the time as 2:08 PM.

The screenshot shows the Akron Partnership page. The navigation bar is the same as the previous page. The main heading is "Benefits for Startups", with buttons for STARTUPS and INVESTORS. Below this is a section titled "Startup Benefits" with the subtitle "Exclusive benefits when you raise through Tyke". Three partner logos are displayed in a grid:

- AWS**: Cloud Services
- CleverTap**: CRM
- Mixpanel**: Analytics

The Windows taskbar at the bottom shows the date as 7/5/2024 and the time as 2:09 PM.

