

Strategic Project WorkFlow System

to Submitted in Partial Fulfilment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Atul Chaurasiya (Roll No.-2104220100036) Kuldeep Giri (Roll No.-2104220100058) Abhishek Kumar(Roll No.-2104220100002) Ashish Rathore (Roll No.-2104220100031)

Under the supervision of Mrs. Mohinee Singh

(Assistant Professor)



BANSAL INSTITUTE OF ENGINEERING AND TECHNOLOGY LUCKNOW

Affiliated to

DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

2024-25

DECLARATION

I hereby declare that the project entitled “Strategic Project Workflow System” submitted for the B. Tech. (Computer Science and Engineering) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship, or any other similar titles.

Name of Group Members with Signature

Atul Chaurasiya (2104220100036)

Kuldeep Giri (2104220100058)

Abhishek Kumar(2104220100002)

Ashish Rathore(2104220100031)

CERTIFICATE

This is to certify that the project titled “Strategic Project Workflow System” is the bonafide work carried out by Atul Chaurasiya (2104220100036), Kuldeep Giri (2104220100058), Abhishek Kumar (2104220100002), and Ashish Rathore (2104220100031), students of B.Tech. (Computer Science and Engineering) at Bansal Institute of Engineering & Technology (College Code-422), Lucknow, affiliated to Dr. APJ Abdul Kalam Technical University, Lucknow, during the academic year 2024-25, in fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship, or any other similar title.

Signature of the Guide

Mrs. Mohinee Singh

Assistant Professor

ABSTRACT

The Strategic Project Workflow System is an intuitive and efficient project management tool designed to streamline task coordination, team collaboration, and workflow tracking within organizations. This web-based platform enables users to create, assign, and monitor tasks in real time, ensuring transparency, accountability, and improved productivity across teams.

Built using Spring Boot, React.js, and Tailwind CSS, the system offers a responsive and user-centric interface, making it easy for users to manage complex project workflows. With features like task status updates, role-based access, progress tracking, and notifications, the system enhances communication and provides better visibility into project timelines and dependencies.

The primary objective of this project is to facilitate structured project management for technical and non-technical teams, promoting organization-wide efficiency and smarter decision-making.

Keywords: project management, workflow system, collaboration, task tracking, productivity, Spring Boot, React.js, Tailwind CSS, team coordination, progress monitoring, real-time updates,

ACKNOWLEDGEMENT

First of all, we are grateful to Almighty, the most beneficent and merciful, who provided us with confidence and determination to accomplish this work.

We are immensely thankful and express our deep sense of gratitude and indebtedness to our Project Guide **Mrs. Mohinee Singh, Assistant Professor and Mr. Shashank Gaur, Assistant Professor, Department of Computer Science & Engineering**, our Project Coordinator. They suggested a problem of vital interest and, through their valuable guidance and advice, helped bring this project to fruition. Their continuous monitoring and time management were an inspiring force for us to complete the work.

Among the rest, we would like to thank all our colleagues and friends—especially our team members—who supported and helped us in the completion of this project.

We are also grateful to the authors and scholars whose work we have referred to as guiding stones throughout the development of this system.

Lastly, we would like to thank our **father and mother**—no words can express our heartfelt gratitude toward them.

Name of Group Members with Signature

Atul Chaurasiya (2104220100036)

Kuldeep Giri(2104220100058)

Abhishek Kumar(2104220100002)

Ashish Rathore(2104220100031)

TABLE OF CONTENTS

Topic Page No

Title i

[Declaration ii](#)

[Certificate iii](#)

[Abstract iv](#)

[Acknowledgement v](#)

Table OfContent vi

[List Of Figures ix](#)

[Abbreviations xi](#)

CHAPTER-1 Introduction 1-7

- - 1. [Introduction 1](#)
 - 2. [Purpose 1](#)
 - 3. [About the Project 1](#)
 - 4. Objective 2
 - 5. [Overview the Project 3](#)
 - 6. [About the Technology 4](#)
 - 1. HTML 4
 - 2. CSS 4
 - 3. JavaScript 4
 - 4. Bootstrap 5
 - 5. Query 5
 - 6. Java 5
 - 7. JSP 6
 - 8. MYSQL 6
 - 9. Servlet 7
 - 10. AWS 7

[CHAPTER-2 Literature Survey 8-9](#)

- - 1. [Existing System 8](#)
 - 1. Disadvantage of the Current 8
 - 2. [Proposed System 8](#)

[CHAPTER-3 Feasibility Analysis 10-11](#)

- - 1. [Introduction 10](#)
 - 1. Technical Feasibility 10
 - 2. Operational Feasibility 11
 - 3. Economic Feasibility 11

[CHAPTER-4 Requirement Analysis 12-13](#)

- - 1. [Introduction 12](#)
 - 1. Hardware Requirements 12
 - 2. Software Requirements 12
 - 3. Language Used 12

CHAPTER-5 Methodology & System Design

14-32

- - 1. Waterfall Model 15
 - 2. Singleton Design Pattern 16
 - 3. Entity-Relationship Model 17
 - 4. Flowchart Diagram (Home) 18
 - 5. Data Flow Diagram (Login) 19
 - 6. Data Flow Diagram (Signup) 19
 - 7. How to Validate OTP 20
 - 8. Use Case Diagram 21
 - 9. Component Design 23
 - 10. Security Design 24
 - 11. Performance Design 26
 - 12. Integration Design 28
 - 13. System Design 29

CHAPTER-6 Implementation 33-55

- - 1. Module Description 33
 - 1. Home Page 33
 - 2. Sign in Page 34
 - 3. Welcome Page 37
 - 2. OTP Validation 40
 - 3. Book Listing 42
 - 4. Searching 44
 - 5. Buying 44
 - 6. Selling 45
 - 7. Feedback Mechanism 46
 - 8. User Authentication & Validation 46
 - 9. Session Management 47
 - 10. Error Handling and Logging 47
 - 11. Performance Optimization 48
 - 12. Data Validation and Sanitization 48
 - 13. API Integration 48
 - 14. Profile Account 49

LIMITATIONS 56

RECOMMENDATIONS 57

FUTURE SCOPE 58

CONCLUSION 59

REFERENCES 60

RESUME

LIST OF FIGURES

S. No	Figure No	Figure Name	Page No
1	1.1	Logo	2
2	1.2	JavaScript Engine	5
3	1.3	JSP Architecture	6
4	1.4	Logical Architecture MySQL	6
5	1.5	Servlet Working	7
6	1.6	AWS	7
7	3.1	Feasibility Diagram	10
8	4.1	Software Requirement Analysis	13
9	5.1	Waterfall Model	16
10	5.2	Singleton Design Pattern	17
11	5.3	Database Management Tables	17
12	5.4	Data Flow Diagram (Home)	18
13	5.5	Data Flow Diagram (Login)	19
14	5.6	Data Flow Diagram (Signup)	19
15	5.7	OTP Authentication Flow	20
16	5.8	Use Case Diagram	21
17	6.1	Home Page	34
18	6.1.1	Home Page-Code	34
19	6.2.1	Sign up Page	35
20	6.2.1.1	Sign up Page-Code	35
21	6.2.2	Sign in Page	36
22	6.2.2.1	Sign in Page-Code	36
23	6.3	Welcome Page	37
24	6.4	Select Book Page	37

25	6.4.1	Select Book Page-Code	38
26	6.5.1	Search Book filter1 Page	38
27	6.5.2	Search Book filter2 Page	39
28	6.6	User Cart Page	39
29	6.6.1	User Cart Page	40
30	6.7	OTP Validation Code	41
31	6.7.1	OTP Validation Forget Password	41
32	6.7.2	OTP Validation Create Account Password	42
33	6.8	Book Listing form	43
34	6.8.1	Book Listing Form-Code	43
35	6.9	Book Searching	44
36	6.10	Book Buying	44
37	6.10.1	Book Buying-Code	45
38	6.11	Book Selling	45
39	6.11.1	Book Selling-Code	46
40	6.12	Session Maintain Code	47
41	6.13	Error Handling Code	48
42	6.14	Activation API	49
43	6.15	My Account Page	49
44	6.15.1	My Account Page-Code	50

ABBREVIATIONS

UI	User Interface
UX	User Experience
DB	Database
API	Application Programming Interface
OTP	One-Time Password
SQL	Structured Query Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
JVM	Java Virtual Machine
MVC	Model-View-Controller
AWS	Amazon Web Services
JSP	Java Server Pages
IDE	Integrated Development Environment
CRUD	Create, Read, Update, Delete
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
CDN	Content Delivery Network
JSON	JavaScript Object Notation
CLI	Command Line Interface
DNS	Domain Name System
SSL	Secure Sockets Layer

Chapter 1

Introduction

1.1 Introduction

The Strategic Project Workflow System is developed to streamline project and task management operations within any organization. The platform focuses on core project management functionalities such as task assignment, progress tracking, team collaboration, and workflow automation.

The concept of automation lies at the heart of this system—automation meaning the design of processes that function independently in response to specific events or triggers. From autopilot systems in aircraft to smart homes, automation has proven effective in simplifying complex workflows. In the same vein, this project management tool enables project teams to manage their entire project lifecycle more efficiently.

Our objective is to provide a digital solution that assists teams—whether technical or non-technical—in managing projects systematically and transparently. Users can create projects, break them into tasks, assign those tasks to members, and monitor status through a centralized dashboard. The system supports real-time updates, notifications, and status changes to help everyone stay aligned and on track.

1.2 Purpose

The purpose of building the Strategic Project Workflow System is to provide organizations, teams, and individuals with a centralized, user-friendly platform to plan, manage, and execute their projects effectively. This platform eliminates the dependency on scattered tools and manual tracking systems by offering a unified interface for task management, communication, and progress visualization.

The system allows users to:

- Create and assign tasks with deadlines.
- Track progress with status indicators.
- Collaborate via comments or notes.
- Get real-time updates and notifications.
- View overall progress through dashboards.

This enhances overall team productivity, ensures better time management, and reduces the risk of delays due to miscommunication or disorganization.

1.3 About the Project

The Strategic Project Workflow System is a web-based application developed using Spring Boot (backend), React.js (frontend), and Tailwind CSS (styling). It acts as a powerful tool for managing workflows across various departments or team projects. Each user has role-based access, allowing administrators, managers, and team members to interact with the system based on their responsibilities.

The system simplifies the entire process of project monitoring by allowing project managers to define goals, distribute tasks among teams, set priorities, and visualize the timeline. Each task can be updated with notes, deadlines, and progress stages, keeping the entire team aligned.

This platform is ideal for teams looking to organize their work efficiently, boost transparency, and enhance accountability—all through a seamless and responsive interface.

-

1. Objectives

The objectives of the Strategic Project Workflow System revolve around enhancing task organization, improving team collaboration, streamlining project tracking, and increasing overall operational efficiency. This section outlines the key goals behind the development of the system and how each objective supports the larger vision of effective project management.

1. Enhancing Workflow Management

Rationale:

Traditional project tracking using spreadsheets or informal tools often leads to miscommunication, task overlap, and missed deadlines. The Strategic Project Workflow System aims to centralize and automate project tracking to reduce such inefficiencies.

Strategies: The system provides structured modules for task creation, assignment, status updates, and progress visualization. This allows team members and managers to stay aligned on responsibilities and deadlines through a single, cohesive platform.

Simplifying Project and Task Management Processes

Rationale:

Traditional approaches to managing projects often involve scattered tools like emails, spreadsheets, or manual tracking, which leads to inefficiencies, miscommunication, and delays. The Strategic Project Workflow System aims to replace these fragmented methods by offering a centralized, structured, and intuitive platform to manage all aspects of a project in one place.

Strategies:

To achieve this, the system incorporates the following key features:

- Advanced Task Management: Users can create tasks with detailed descriptions, deadlines, priorities, and assignees, ensuring clear and organized delegation.
- User Dashboards: Personalized views for each user display pending tasks, deadlines, and project summaries, helping individuals stay focused.
- Real-Time Updates: The system provides instant status updates and notifications whenever a task is modified, reducing the need for follow-ups.
- Activity Logs: Automatic logging of every user action increases accountability and allows tracking of progress at every stage.
- Intuitive UI/UX: Built with React.js and Tailwind CSS, the interface ensures ease of use, reducing the learning curve and encouraging adoption.

These strategies work together to minimize confusion, boost efficiency, and enable teams to concentrate on achieving their project goals rather than struggling with administrative overhead.

Enhancing User Experience

Rationale:

The success of any project management tool lies in its user experience (UX). A seamless, intuitive, and visually appealing platform ensures that users can easily navigate and utilize the system, thus promoting greater adoption and sustained use. The Strategic Project Workflow System prioritizes a user-friendly design to enhance team collaboration and project tracking.

Strategies:

This objective focuses on the following key aspects:

- User-Centered Design: The system is designed to focus on user needs, with streamlined workflows that align with user goals.
- Intuitive Navigation: The interface is built with React.js and Tailwind CSS, ensuring that users can quickly navigate through tasks, projects, and team communications.
- Responsive Layouts: The platform is fully responsive, providing an optimal viewing experience across devices, whether desktop, tablet, or mobile.
- Interactive Features: Features such as drag-and-drop task management, live notifications, and real-time collaboration help create an engaging experience for users.

Additionally, feedback loops are incorporated, where users can rate features and suggest improvements, ensuring continuous enhancement of the system's usability.

Fostering Collaboration Among Users

Rationale:

Collaboration and transparency are essential in successful project management. The Strategic Project Workflow System aims to bridge communication gaps between team members and stakeholders by fostering a collaborative environment.

Strategies:

The system implements the following features to encourage collaboration:

- Shared Dashboards and Task Views: Team members can share and track task progress in real-time, ensuring that everyone is aligned and on the same page.
- Real-Time Updates and Notifications: Users are notified of changes in task status, comments, or new assignments, ensuring transparent and instant communication.
- Discussion Threads and Notes: Teams can discuss specific tasks or ideas using comment threads, ensuring that feedback is accessible and organized.
- Collaboration Tools: The system provides tools for file sharing, team announcements, and integrated chat functionality to encourage ongoing collaboration.

These features help create an open, collaborative space where team members can contribute ideas, track each other's progress, and ensure successful project execution.

1.1 Overview of the Project

The Strategic Project Workflow System is a comprehensive web application designed to facilitate the management of projects and tasks within a team. Developed using Spring Boot for the backend and React.js with Tailwind CSS for the frontend, the system offers a responsive, scalable solution to streamline project management processes.

- Frontend(React.js&TailwindCSS):

The frontend offers a dynamic and user-friendly interface where users can interact with various project management features, such as task assignment, status updates, and progress tracking. React.js ensures a responsive experience, while Tailwind CSS allows for rapid, customized design implementation.

- Backend(Spring-Boot):

The backend uses Spring Boot to handle user authentication, data processing, and server-side logic. It connects seamlessly with a relational database (e.g., MySQL or PostgreSQL) to manage project and task data.

- Database(SQL):

The system relies on a robust relational database to store user data, project information, task assignments, and collaboration details. SQL queries handle data retrieval, updates, and task history logging.

By leveraging these technologies, the Strategic Project Workflow System offers a comprehensive and efficient solution for project management, task delegation, team collaboration, and performance tracking

- - 1. About The Technology
 - 1. **HTML**

Hypertext Markup Language (HTML) is the standard markup language for creating web pages that are displayed in a web browser. HTML defines the structure of a web page by using elements enclosed within tags, like `<html>`, `<body>`, and `<div>`. These elements are typically paired with an opening and closing tag (e.g., `<div>` and `</div>`). Some elements, like ``, are self-closing and do not require a closing tag.

HTML serves as the foundation for all web pages and is essential for organizing and structuring content. In the **Strategic Project Workflow System**, HTML is used to create the static structure of the web pages, ensuring that users can view and interact with the essential components like task lists, project boards, user profiles, and real-time updates.

- - - 1. **CSS**

Cascading Style Sheets (CSS) is a style sheet language used to control the presentation and layout of a document written in HTML. CSS allows web developers to separate content (written in HTML) from design elements such as color schemes, fonts, layouts, and spacing. This separation enhances accessibility, makes the website easier to maintain, and reduces redundancy in the markup.

In the **Strategic Project Workflow System**, CSS (combined with **Tailwind CSS**) is utilized for designing the user interface, ensuring that the platform is visually appealing, responsive, and easy to navigate. Tailwind CSS, a utility-first framework, provides a flexible approach to creating custom designs while ensuring that the system adapts seamlessly across different devices.

- - - 1. **JavaScript**

JavaScript (JS) is a dynamic, high-level programming language that is primarily used to create interactive effects within web browsers. Originally designed for client-side scripting, JavaScript enables real-time user interactions and dynamic content updates. It is a prototype-based language that supports first-class functions and dynamic typing, making it ideal for building interactive web applications.

In the **Strategic Project Workflow System**, JavaScript (along with **React.js**) is responsible for creating a dynamic and interactive user interface. JavaScript allows for real-time updates, seamless navigation, and interactive features like task assignment, status updates, and instant notifications. React.js enhances the user experience by efficiently rendering changes to the UI without needing to reload the page, providing a smooth and responsive experience.

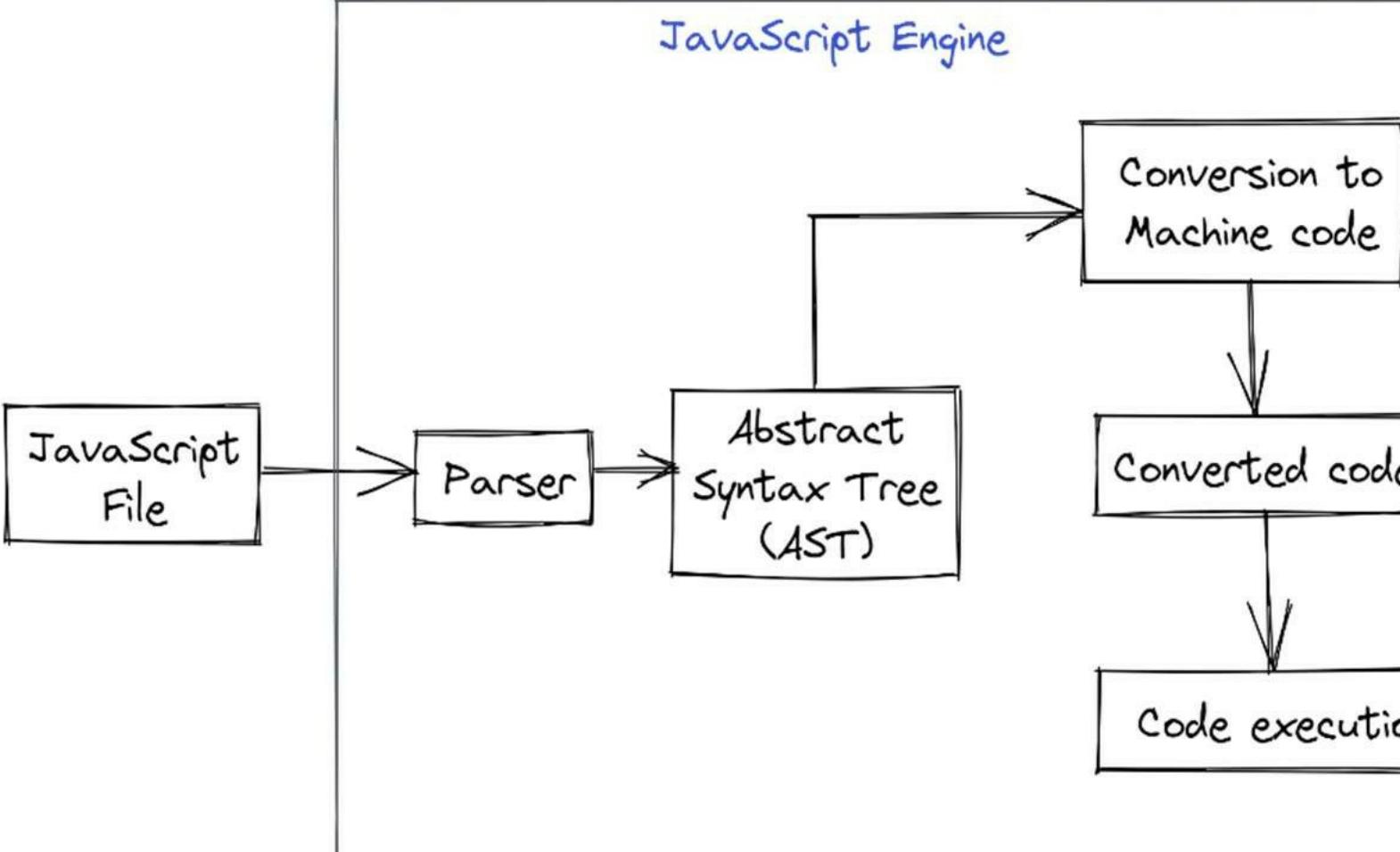


Fig 1.2 JavaScript Engine

- - - 1. **Bootstrap**

Bootstrap is a popular free and open-source front-end web framework used for designing responsive and mobile-first websites and web applications. Developed by Twitter, Bootstrap includes HTML and CSS-based design templates for common UI components such as typography, buttons, forms, tables, and navigation bars. Additionally, it offers optional JavaScript plugins to enhance interactivity.

Bootstrap promotes rapid and consistent development through its modular structure and predefined classes. It is built on a responsive grid system and provides numerous customization options through configuration variables. Developers can include the entire framework or use specific components based on project needs. In the context of the **Strategic Project Workflow System**, Bootstrap helps in building a consistent and responsive layout, ensuring compatibility across various devices and screen sizes.

1. JQuery

JQuery is a lightweight, fast, and feature-rich JavaScript library designed to simplify HTML document traversal and manipulation, event handling, animation, and Ajax interactions. As an open-source tool, JQuery is widely used across web applications and is known for its cross-browser compatibility.

Its concise and easy-to-use syntax allows developers to perform complex tasks with fewer lines of code. JQuery also supports plugin development, making it extensible and customizable. Within the **Strategic Project Workflow System**, JQuery is utilized to enhance interactivity, streamline DOM manipulation, and manage asynchronous server interactions smoothly.

1. Java

Java is a versatile, object-oriented programming language renowned for its platform independence, security features, and scalability. It is widely used for building enterprise-level applications, including web-based systems.

In the **Strategic Project Workflow System**, Java serves as the core backend technology. It is used for implementing business logic, processing user requests, handling sessions, and interacting with the database. The use of **JSP (JavaServer Pages)** and **Servlets** facilitates dynamic content generation and request handling, ensuring robust server-side functionality.

1. MYSQL

MySQL is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for managing and querying data. It is known for its speed, reliability, and ease of use, making it a preferred choice for web applications.

In the **Strategic Project Workflow System**, MySQL is responsible for storing and managing essential data, including user profiles, project tasks, status updates, and collaboration history. It ensures data integrity and supports efficient data retrieval and manipulation operations, enabling the system to deliver a responsive and seamless experience to users.

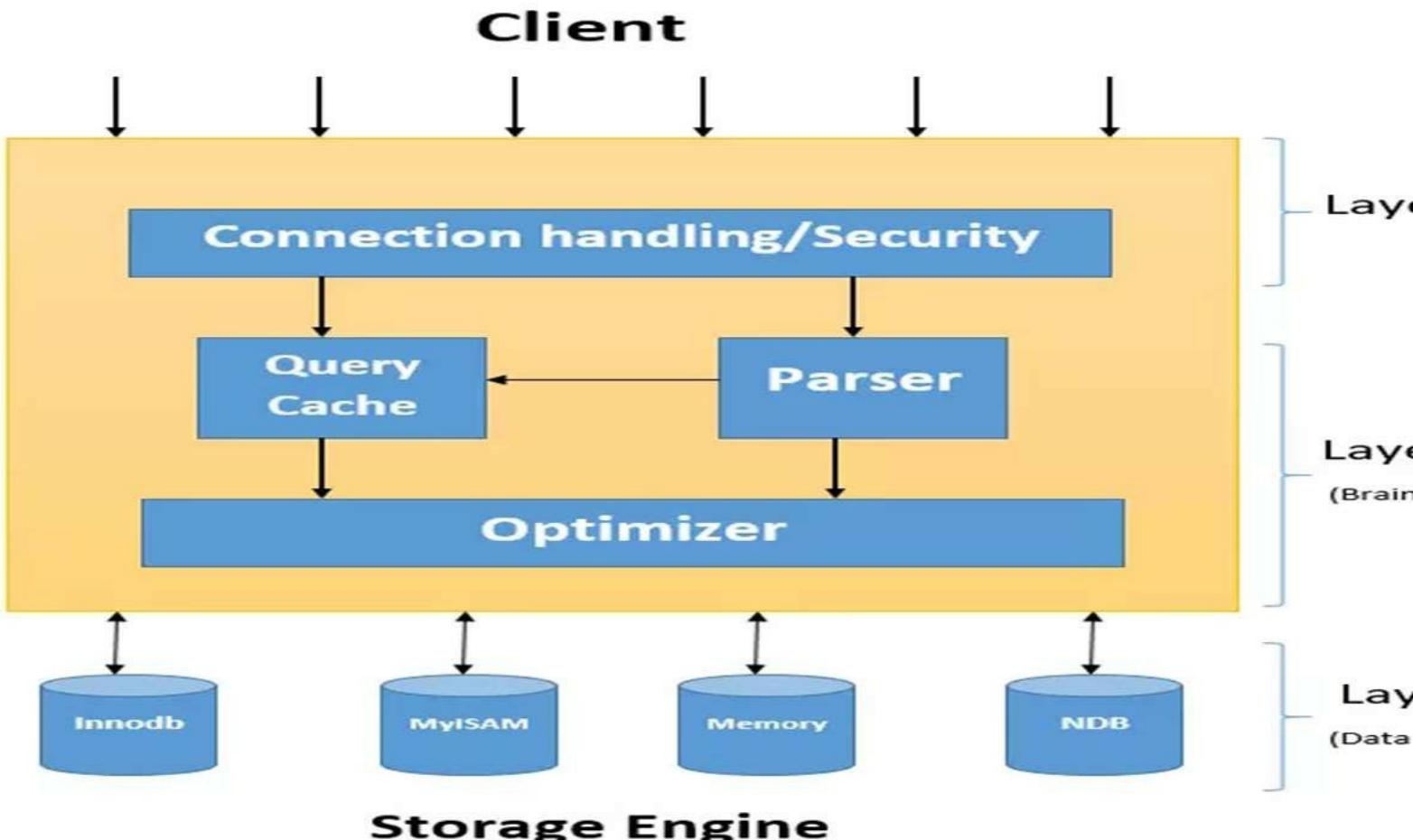


Fig 1.4 Logical Architecture MySQL

1. AWS

Amazon Web Services (AWS) is a leading cloud computing platform that offers a vast array of services, including computing power, storage, databases, and networking. It enables developers and organizations to deploy, manage, and scale applications efficiently and cost-effectively.

In the context of the **Strategic Project Workflow System**, AWS provides a robust infrastructure for hosting and managing the application. Services such as **EC2 (Elastic Compute Cloud)** can be used to host the web application, ensuring high availability and accessibility over the internet. Additionally, **AWS RDS (Relational Database Service)** can be integrated to manage the MySQL database, offering automated backups, scalability, security, and reduced administrative overhead.

By leveraging AWS, the system benefits from improved performance, reliability, and security—critical for supporting multiple users, maintaining uptime, and ensuring data protection in a collaborative environment.

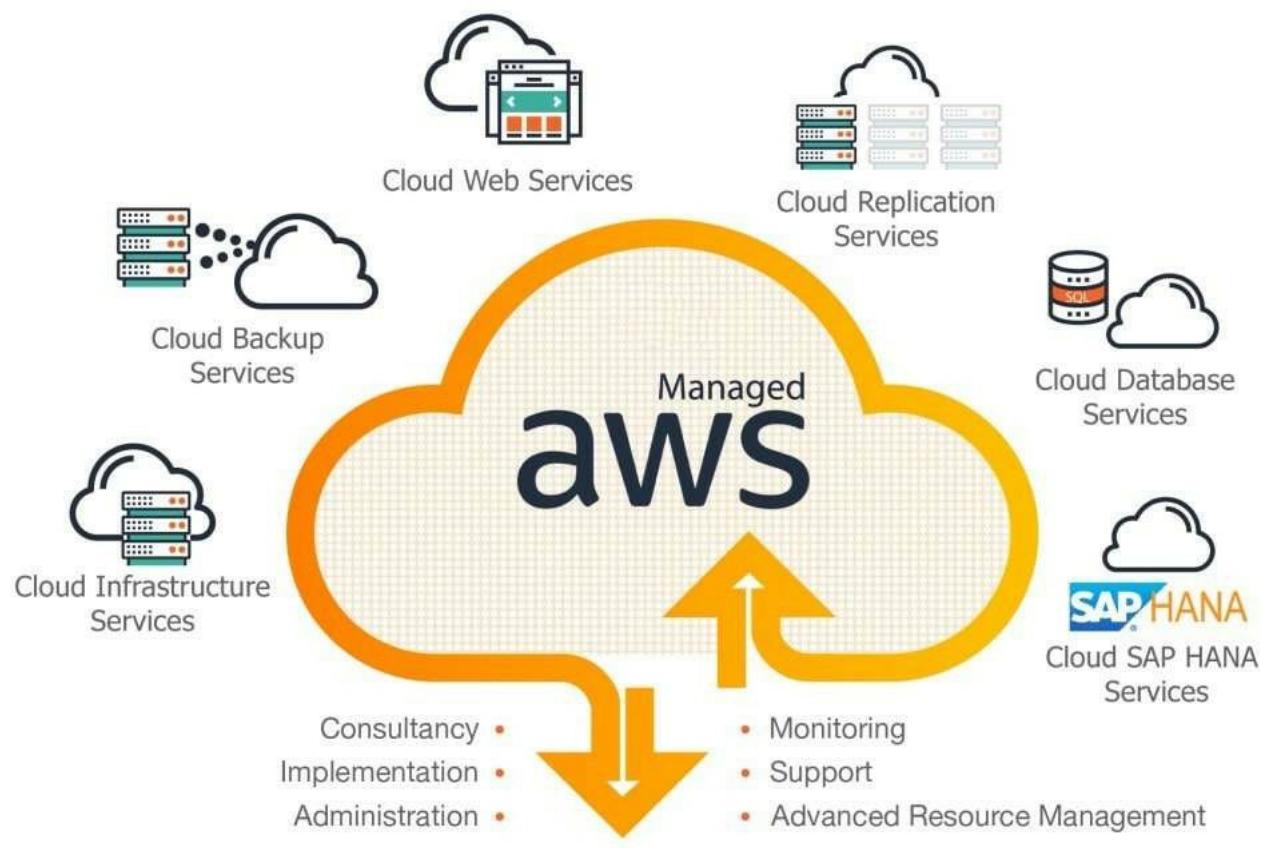


Fig 1.6 AWS

Chapter 2 Literature Survey

2.1 Existing System

The current system for buying and selling books is entirely manual. Users typically rely on offline methods such as notice boards, physical meetups, or word-of-mouth to manage book transactions. This process lacks efficiency, structure, and communication between users and administrators. It is difficult to track historical data, and managing orders or transactions becomes cumbersome.

In the existing setup, the process of ordering books is completely dependent on coordination between the seller and the customer, with no involvement or oversight by the site manager. Additionally, there is no structured mechanism for users to raise enquiries or interact with the admin. While users can inquire about books, the process is not streamlined or user-friendly.

2.1.1 Disadvantages of the Current System

- No direct communication between users and the administrator.
- Frequent errors and corrections delay transactions.
- Server upload issues are common.
- The system is time-consuming and inefficient.
- Lack of support for modifying orders or listings.
- High risk of losing order details or data.
- No proper enquiry or feedback mechanism.
- Limited or no filtering/searching options for users.
- Reviewing available books is tedious and inefficient.

Proposed System

The **proposed system** is a comprehensive online platform designed to simplify and digitize the process of buying and selling textbooks. It allows users to advertise books for sale by uploading images, descriptions, and pricing details. The system supports a built-in enquiry mechanism, chat messaging (similar to email), and user-to-user as well as user-to-admin communication.

- The administrator has enhanced capabilities including:
- Authenticating users (admin, registered users, guest users)
- Removing inappropriate content or users
- Adding and managing book listings
- Communicating with users for moderation or support

Study Emporium represents a shift from manual inefficiency to an organized, secure, and accessible textbook marketplace—meeting the needs of students and educational institutions.

1. **Textbook Trading:** The traditional textbook market has been marked by high costs and limited purchasing avenues for students. According to Johnson et al. (2018), online textbook trading platforms help address these issues through peer-to-peer transactions and increased price transparency. Study Emporium builds on this concept by offering a wide range of textbooks from different sellers, empowering students with affordability, variety, and accessibility.
2. **Educational Technology:** Educational Technology (EdTech) is increasingly influencing modern learning environments. Mishra and Yadav (2020) highlight how EdTech tools enhance learning through personalization, interactivity, and collaboration. **Study Emporium** integrates these elements by adopting user-centric design, interactive browsing experiences, and collaborative tools—making textbook buying and selling more engaging and effective.
3. **User Experience and Satisfaction:** User experience (UX) plays a critical role in the success of any digital platform. Research by Kim and Forsythe (2018) indicates that a well-designed user interface, smooth navigation, and trustworthy transactions contribute significantly to user satisfaction and loyalty. **Study Emporium** reflects these best practices by offering an intuitive interface, responsive layout, and secure payment mechanisms.
4. **Impact of Online Marketplaces on Traditional Retail:** Online marketplaces have reshaped consumer behavior and challenged traditional retail models. Chatterjee et al. (2017) discuss the disruptive influence of e-commerce on brick-and-mortar stores, emphasizing the need for digital adaptation. **Study Emporium** not only caters to student users but also provides an avenue for physical bookstores to reach a wider online audience, helping them remain relevant in a digital-first environment.

Introduction

The feasibility study evaluates the practicality and viability of implementing the **Worldwide Flight Tourism Project**, focusing on the integration of new modules and the refinement of existing functionalities. The goal is to assess whether the project is technically, operationally, and economically achievable within the available resources and time constraints.

It is important to recognize that **any system can appear feasible** under the assumption of **unlimited resources and infinite time**. However, in real-world scenarios, feasibility must be carefully analyzed under practical constraints. This section outlines the three critical aspects of feasibility:

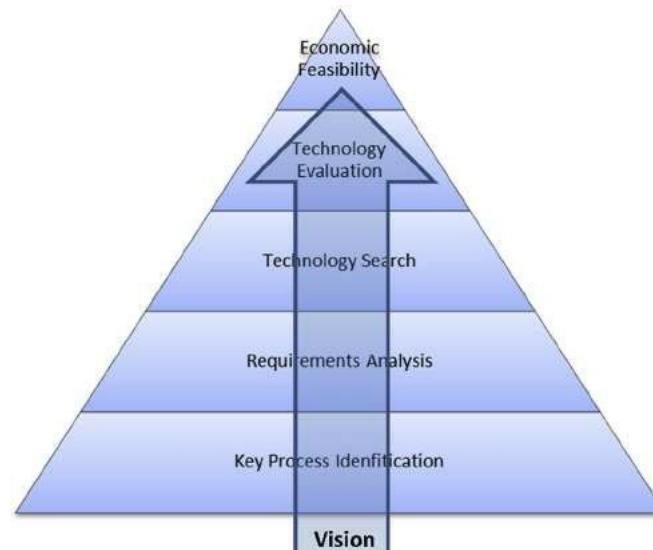


Fig 3.1 Feasibility Diagram

1. Technical Feasibility

Technical feasibility assesses whether the current technology and infrastructure can support the development and integration of the proposed modules. This includes:

- - - Evaluating compatibility with existing hardware and software systems.
 - Determining the availability of skilled technical personnel.
 - Ensuring scalability and reliability of the system.
 - Identifying the tools and platforms required for development.

Conclusion: The project is technically feasible as modern tools, cloud platforms, and development frameworks are readily available to implement and enhance the system effectively.

1. Operational Feasibility

Operational feasibility evaluates how well the proposed system will function in the operational environment, and whether users will accept and effectively use the system. This includes:

- Assessing user friendliness of the system interface.
- Determining the impact on current workflow and processes.
- Evaluating staff training requirements.
- Ensuring system reliability during real-time operations.

1. Economic Feasibility

Economic feasibility analyzes whether the benefits of the system outweigh the costs involved in its development and maintenance. This includes:

- Estimating development, deployment, and maintenance costs.
- Calculating expected returns through improved efficiency and user engagement.
- Analyzing cost-saving measures like automation and digital transformation.

Chapter 4 Requirement Analysis

Introduction

To ensure the successful development and smooth execution of the project, a comprehensive understanding of all requirements is essential. This includes hardware, software, and programming technologies. Requirement analysis plays a crucial role, as it lays the foundation for implementation by identifying the necessary tools and resources.

1. Hardware Specifications

Processor : Intel Core i3 or AMD Ryzen 3 (or equivalent)

with a clock speed of at least 1.8 GHz Processor Speed : 1 GHz CPU

RAM : 4 GB

Hard disk : 500 GB

Graphics : Integrated Graphics Card

- - - 1. Software Specifications

Operating System : Windows 8.1/10/11

Languages : HTML, Java, CSS, JavaScript, jQuery, JSP,

Servlet.

Back End : MySQL

IDE : Eclipse, Sublime Text Editor

Browser : Opera, Mozilla Firefox, Google Chrome

- - - 1. Languages Used
 - HTML (Hypertext Markup Language): For creating the structure and content of web pages.
 - CSS (Cascading Style Sheets): For styling and formatting the appearance of web pages.
 - JavaScript: For adding interactivity and dynamic features to web pages.
 - jQuery: A JavaScript library used for simplifying tasks like DOM manipulation and event handling.
 - Java: For backend programming, handling server-side logic, and implementing business logic.
 - JSP (Java Server Pages): For generating dynamic web pages with Java.
 - Servlets: Java classes used to handle requests from web clients, process data, and generate responses.
 - MySQL: A relational database management system (RDBMS) used for storing and managing data related to users, books, transactions, etc.

Software Requirement Analysis

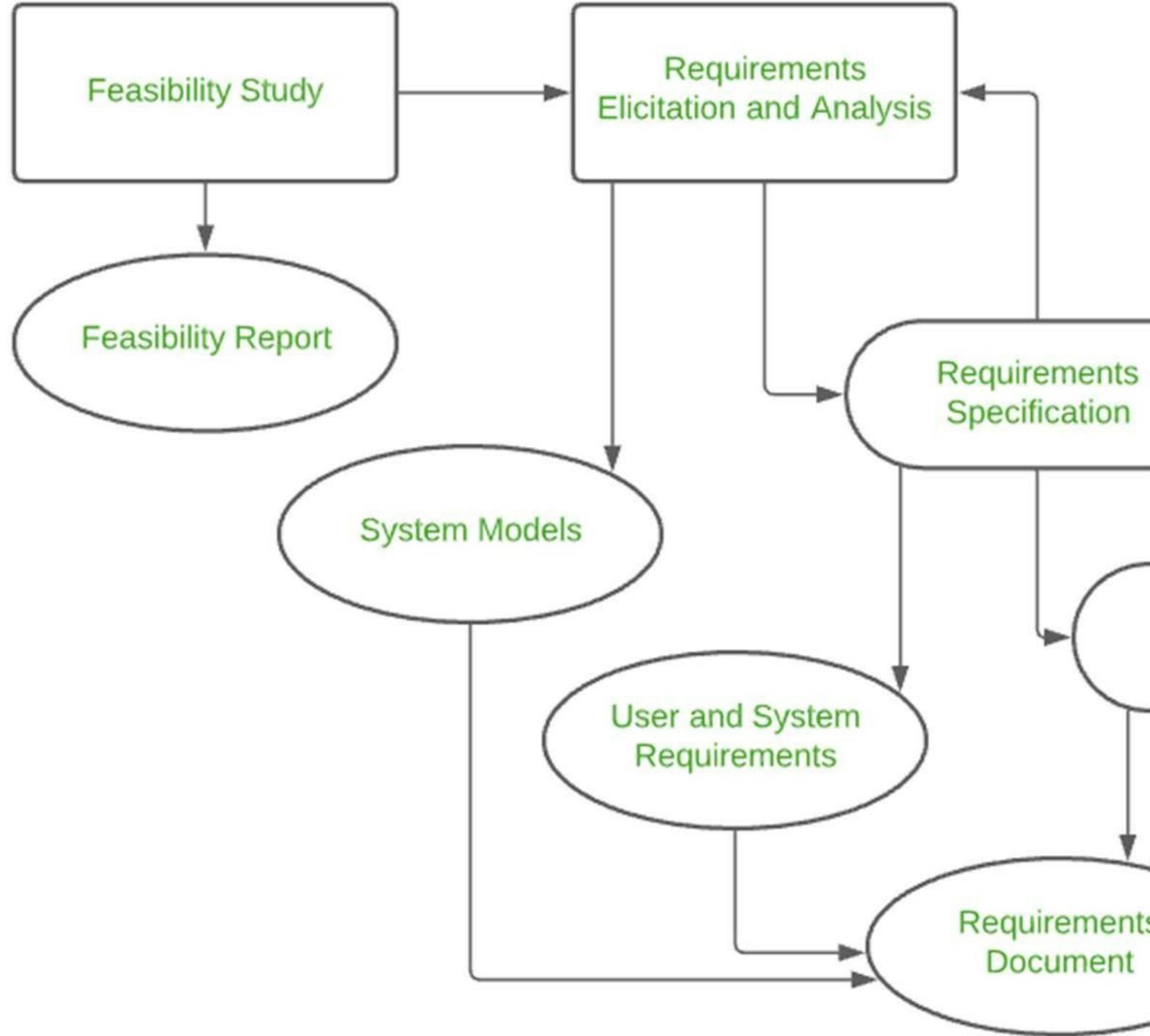


Fig 4.1 Software Requirement Analysis

Chapter 5 Methodology & System Design

The **Strategic Project Workflow System (SPWS)** is designed to manage, streamline, and optimize project workflows in a structured, organized manner. The system aims to facilitate efficient task management, resource allocation, team collaboration, and progress tracking in a project environment. The architecture of the SPWS is built to be modular, scalable, and secure, ensuring that it can accommodate diverse project workflows, user roles, and enterprise-level requirements. Modularity is one of the defining characteristics of Study Emporium's architecture. This approach divides the system into distinct, interchangeable components, each responsible for a specific functionality. By structuring the system in this way, it offers several key benefits:

Architecture Overview:

The system is divided into three primary layers:

1. **Frontend Layer** – The user-facing interface where project managers, team members, and other stakeholders interact with the system.
2. **Backend Layer** – The core logic and business rules that manage data processing, project flow, and resource management.
3. **Database Layer** – The data storage component that securely stores project, task, user, and other related information.

Each of these layers works cohesively to ensure the smooth functioning of the system, with a focus on **modularity** and **separation of concerns**.

1. Frontend Layer:

The frontend layer of the Strategic Project Workflow System is designed to provide an intuitive, user-friendly interface for all system users. It is responsible for presenting real-time data, managing user inputs, and providing necessary controls for task management, collaboration, and project tracking.

Responsibilities:

- **Project Dashboard:** Displays an overview of ongoing projects, deadlines, milestones, and resource allocation.
- **Task Management Interface:** Allows users to create, assign, update, and track tasks within a project.
- **Team Collaboration Tools:** Facilitates communication among project team members, including real-time updates, chat, and notifications.
- **Reports & Analytics:** Offers project progress insights, team performance metrics, and milestone completion status.

Technologies Used:

- **React.js**: For building dynamic, single-page applications with responsive interfaces.
- **HTML/CSS**: For structuring and styling web pages.
- **JavaScript (ES6+)**: For adding interactivity and dynamic content.
- **Axios/FETCH**: For API calls to interact with the backend server.

2. Backend Layer:

The backend layer is the heart of the system, containing the business logic and serving as the communication bridge between the frontend and the database. It handles the processing of user requests, task assignments, resource management, and tracking project progress.

Responsibilities:

- **Task and Project Management**: Manages the creation, update, and deletion of tasks and projects.
- **User Management**: Handles user authentication, authorization, and role-based access control (admin, project manager, team member).
- **Resource Allocation**: Ensures that tasks are assigned to the appropriate team members and that resources are distributed efficiently.
- **Progress Tracking**: Monitors and updates the status of projects and tasks, sending notifications about due dates and project milestones.

Technologies Used:

- Spring Boot (Java): A powerful backend framework used for implementing business logic, creating RESTful APIs, and handling HTTP requests.
- JWT (JSON Web Tokens): For secure user authentication and session management.
- Spring Security: For role-based access control and authorization.
- REST APIs: For communication between frontend and backend.
- Apache Kafka: For handling real-time project updates and notifications.

3. Database Layer:

The **database layer** stores all the persistent data for the system, including project details, task information, user data, and team communications. The design of the database is optimized for performance, security, and scalability, ensuring quick retrieval and secure storage of data.

Responsibilities:

- Data Storage: Stores information about users, projects, tasks, timelines, and other necessary entities.
- Data Integrity: Ensures that data is consistent and accurate, maintaining relationships between entities like users, tasks, and resources.
- Efficient Querying: Supports fast, efficient queries for data retrieval, especially for large datasets in enterprise-level projects.

Technologies Used:

- MySQL: A robust relational database management system (RDBMS) used for storing and managing structured data.
- SQL Queries: For querying and managing the database.
- ORM (Hibernate): For object-relational mapping to streamline data interaction in the backend.

Key Principles of the Architecture

1. Modularity

- The system follows a modular approach where each part of the application (frontend, backend, and database) can be independently modified or updated without affecting the entire platform.
- Modules like task management, user management, project analytics, and collaboration tools can be enhanced or replaced as needed.

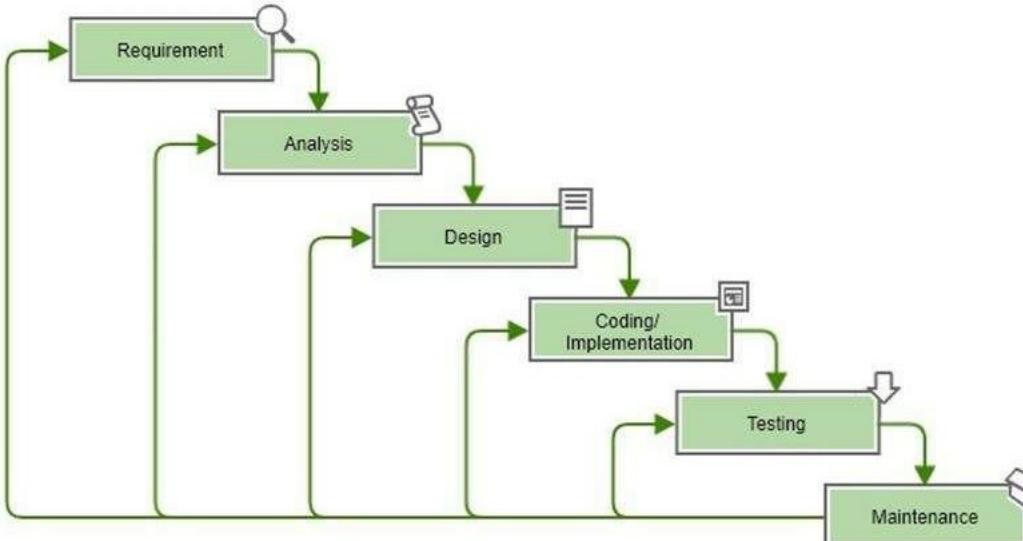


Fig 5.1: Waterfall Model

The Singleton Design Pattern is a crucial architectural pattern employed in the Strategic Project Workflow System to ensure that specific components, particularly those dealing with global resources like database connections and service managers, are instantiated only once throughout the system's lifecycle. This pattern offers several benefits in terms of efficiency, resource management, and consistency, making it highly suitable for managing critical resources in a large-scale system like StrategicProjectWorkflowSystem.

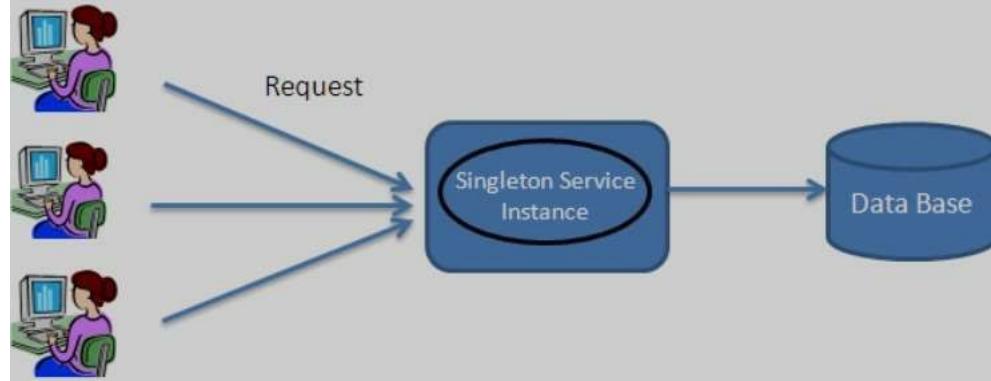


Fig 5.2: Singleton Design Pattern

- 1. Entity-Relationship Model:
- The **Strategic Project Workflow System** relies on an organized and efficient database design to manage essential data, such as user information, project tasks, milestones, and feedback. The system utilizes MySQL, which offers excellent reliability and scalability, ensuring smooth management of large datasets and complex relationships between entities. Registration

Normalization:

Normalization techniques are applied to ensure data integrity and eliminate redundancy in the database schema. By breaking down data into smaller, related tables and reducing data duplication, normalization helps optimize storage and

minimize update anomalies. The schema is organized into multiple normalized forms, adhering to the principles of atomicity, consistency, isolation, and durability(ACID).

Indexing and Optimization:

Indexing strategies are employed to enhance query performance and speed up data retrieval operations. Indexes are created on frequently queried columns, such as Book titles or User IDs, to facilitate efficient data access. Additionally, optimization techniques such as query optimization and database tuning are implemented to fine-tune the database performance and improve overall system responsiveness.

Data Security:

Data security measures are implemented to safeguard sensitive user information stored in the database. This includes encryption techniques to protect data at rest and in transit, access control mechanisms to restrict unauthorized access to database resources, and data masking techniques to anonymize personally identifiable information (PII). Regular security audits and vulnerability assessments are conducted to identify and mitigate potential security threats.

Scalability and Flexibility:

The database design is engineered to support the scalability and flexibility requirements of Study Emporium. Scalability features such as horizontal partitioning and vertical scaling are implemented to accommodate growing data volumes and increasing user traffic. The schema is designed to be flexible, allowing for easy adaptation to evolving business needs and future enhancements.

- 1. Flowchart Diagram (Home):

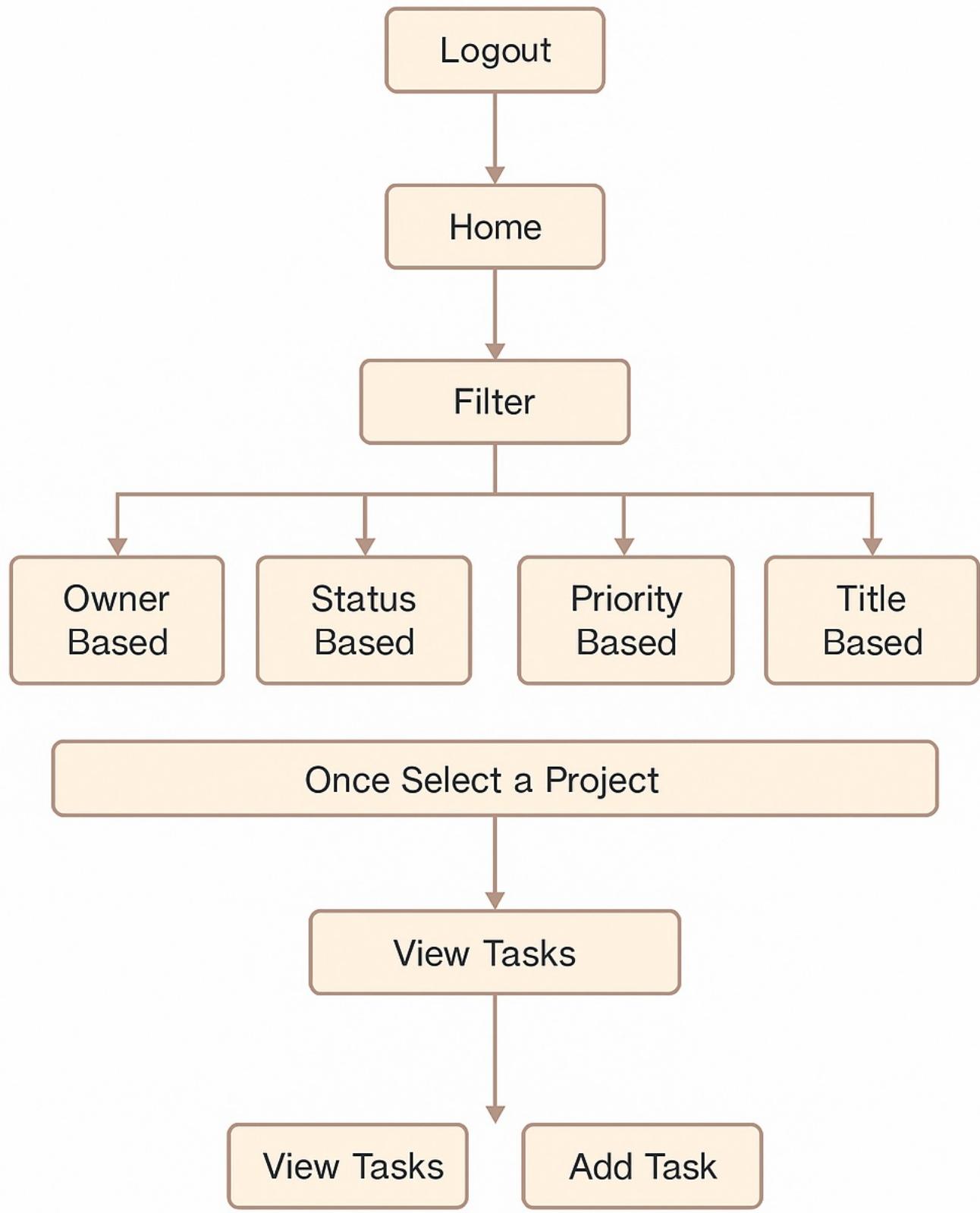


Fig 5.4: Data Flow Diagram (Home)

- 1. Data Flow Diagram (Login):

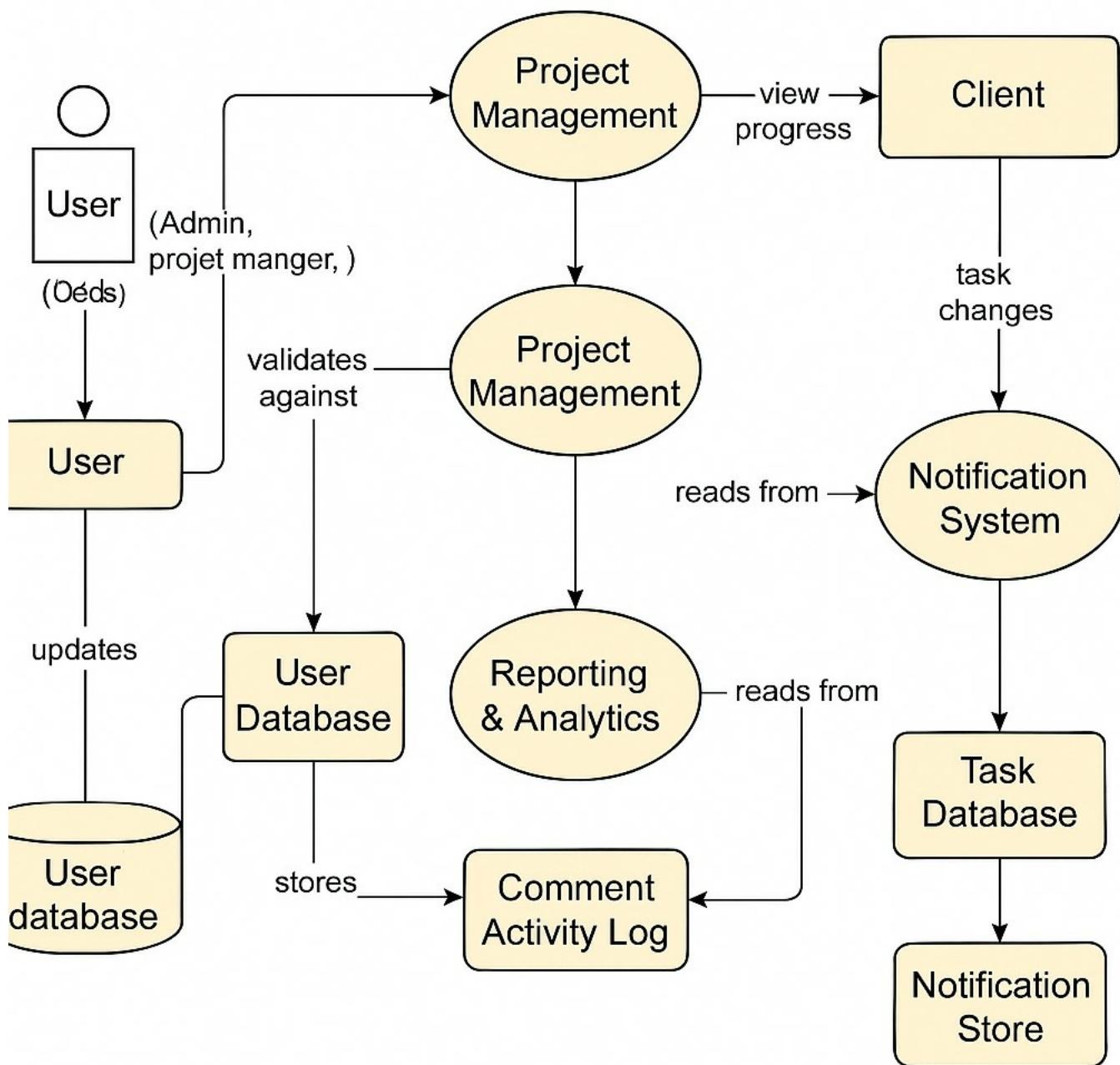


Fig 5.5: Data Flow Diagram (Login)

In the data flow diagram of Study Emporium, user interactions and data flow are visually represented to illustrate the system's functionality. It showcases how data moves between various components within the system, including users, the database, and external entities such as payment gateways. The diagram delineates processes such as user registration, browsing products, adding items to the cart, and completing purchases. By mapping out these data flows, the diagram provides a comprehensive overview of the system's operations, aiding in understanding and analysis of Study Emporium's functionality and data handling processes.

- 1. Data Flow Diagram(Signup):

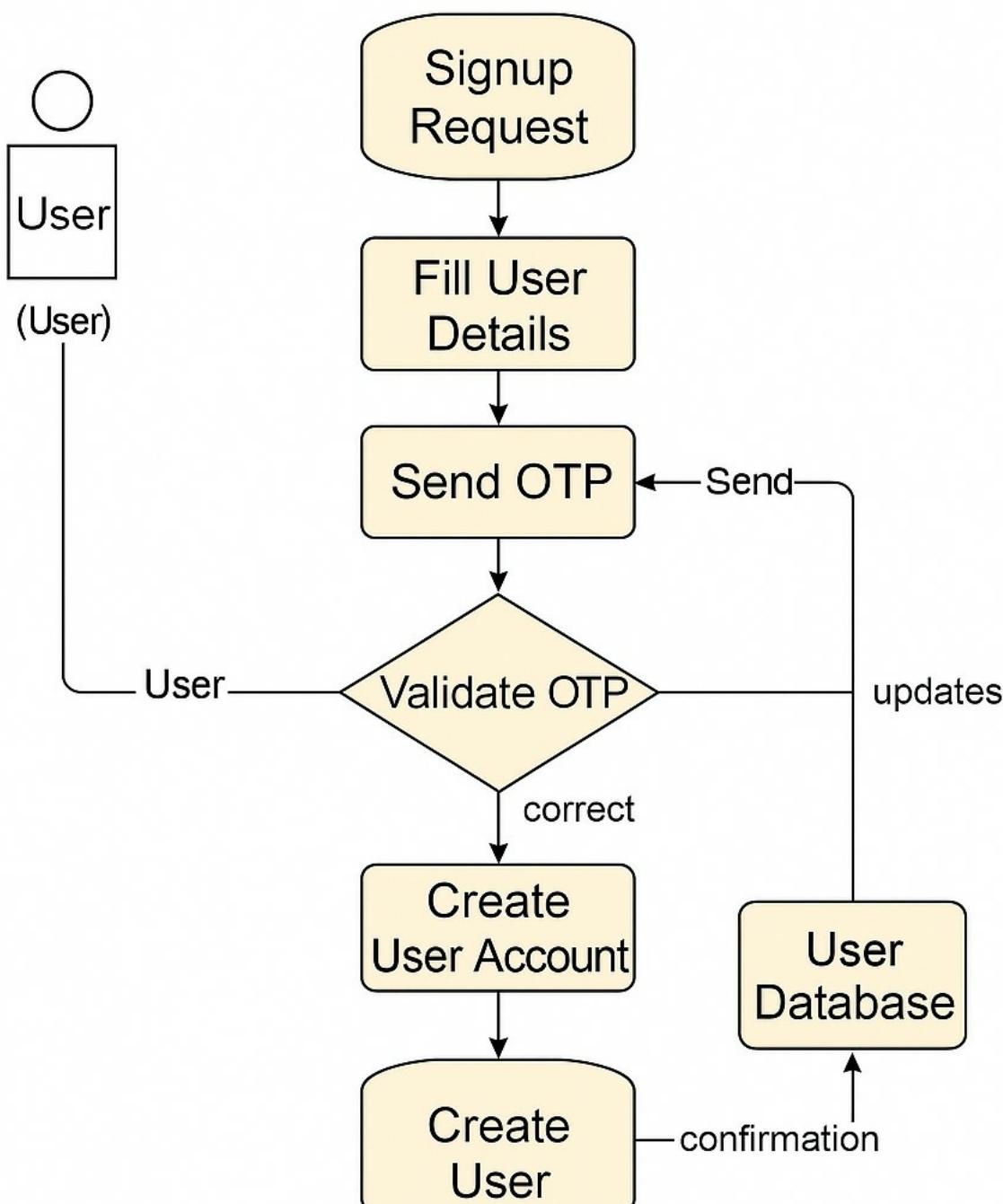


Fig 5.6: Data Flow Diagram (Signup)

How to Validate OTP(One-Time-Password) for Signup Process:

To validate OTP (One-Time Password) through email for signup, you can follow these steps:

-
- 1. Generate OTP: When a user signs up, generate a random OTP and associate it with the user's email address. This OTP will be sent to the user via email for validation.
 2. Send OTP via Email: Use an email service provider (such as SendGrid, Mailgun, or Amazon SES) to send the OTP to the user's email address. Include instructions for the user to enter the OTP on the signup form.
 3. User Enters OTP: Prompt the user to enter the OTP they received in their email on the signup form.
 4. Validate OTP: On form submission, compare the OTP entered by the user with the OTP generated and stored in your database. If they match, proceed with the signup process. Otherwise, display an error message indicating that the OTP is invalid.
 5. Expiry and Resending OTP: Set an expiry time for the OTP (e.g., 5 minutes) to ensure security. If the OTP expires before the user enters it, prompt them to request a new OTP. Provide an option to resend the OTP via email.
 - Call `generateOTP()` to generate a random OTP.
 - Call `sendOTP(email, otp)` to send the OTP to the specified email address.

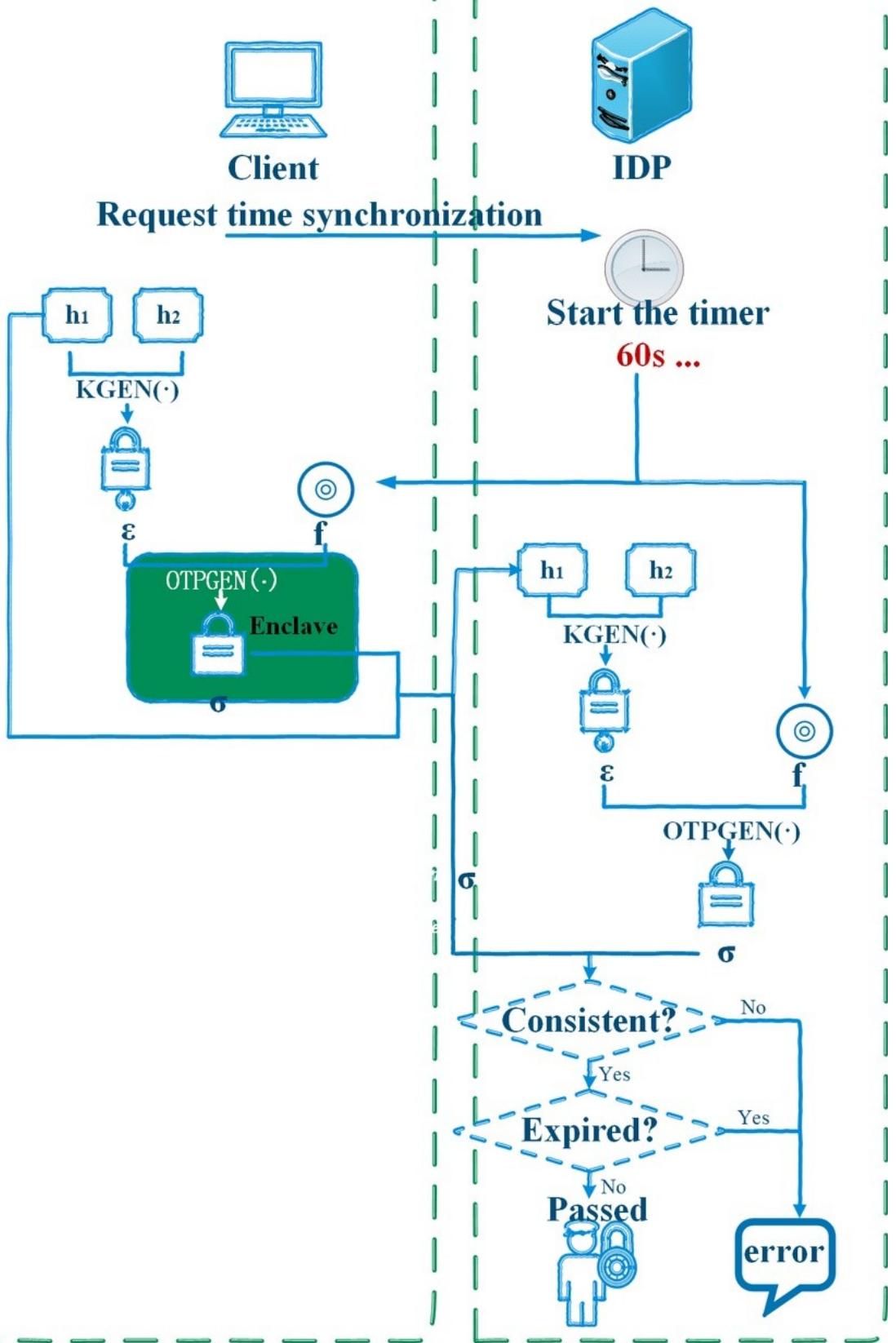


Fig 5.7: OTP Authentication flow

Use Case Diagram:

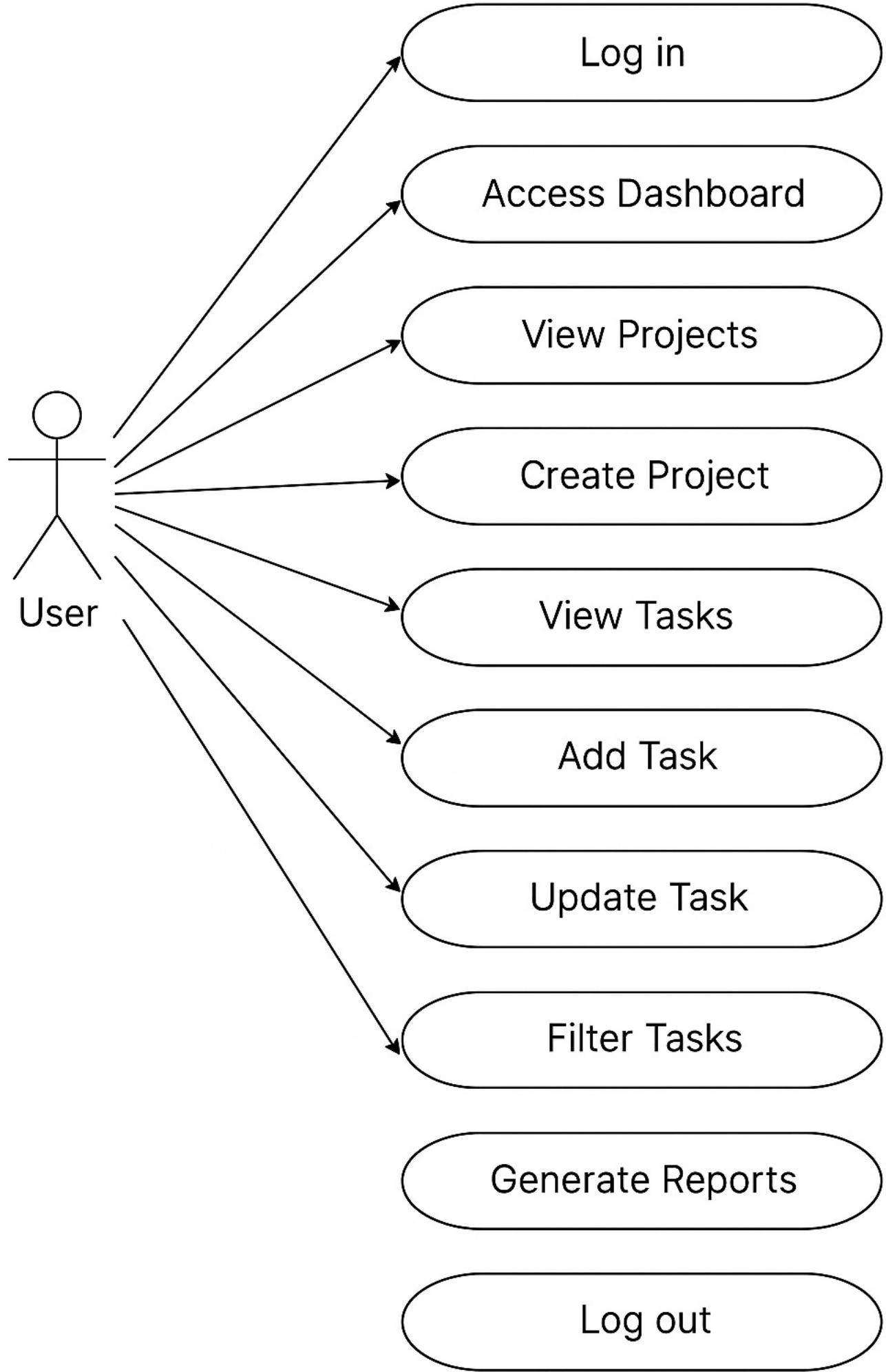


Fig 5.8: Use Case Diagram

A Use Case Diagram is a visual representation of the functional requirements of a system, showcasing the interactions between users (actors) and the system itself. For the **Strategic Project Workflow System**, the use case diagram helps illustrate how various users—such as team members, project managers, and administrators—interact with the platform. This diagram outlines the primary use cases and the associated actors involved in managing and tracking project workflows effectively.

The Use Case Diagram for the **Strategic Project Workflow System** provides a clear visual representation of the interactions between users and the system. It highlights the primary functionalities and the actors involved, helping to ensure that all user needs and system requirements are effectively addressed in the platform's design and development. This structured approach aids in identifying and defining the necessary components and their interactions, facilitating a more comprehensive and user-centric project management system.

- - 1. Log in
 - Allows authenticated users to access the system using their credentials. User Login.
 - 1. Access Dashboard
 - Provides users with an overview of current projects, tasks, deadlines, and status summaries in a centralized view.
 - 1. View Projects
 - Enables users to see the list of all active and archived projects they are involved in.
 - 1. Create Project
 - Allows users (typically project managers) to create new projects by entering essential information like project title, description, team members, and timelines. Administer User Accounts
 - 1. View Tasks
 - Lets users view detailed information about the tasks assigned to them or others within a project

6. Add Tasks

- Enables users to create new tasks within a project, define descriptions, assign members, and set deadlines.

7. Update Task

- Allows users to update the status, comments, progress, or other details of an existing task.

8. Filter Tasks

- Lets users filter tasks based on priority, status, due date, or assignee to manage work more efficiently.

9. Log out

- Safely ends the user session and redirects them to the login or landing page.

Component Design:

Component design is a critical phase in the development lifecycle of the Strategic Project Workflow System, where the system is broken down into smaller, manageable units or components. Each component is responsible for handling a specific functionality such as user management, project creation, task assignment, status tracking, or report generation.

Key Concepts in Component Design

- Modularity: Dividing the system into distinct components, each responsible for a specific part of the functionality.
- Cohesion and Coupling: Ensuring that components are highly cohesive (internally focused on a single task) and loosely coupled (independent of each other).
- Reusability: Designing components that can be reused across different parts of the system or in different projects.
- Encapsulation: Hiding the internal details of the components and exposing only what is necessary.

Steps in Component Design

- Identify Components: Based on the system requirements and architecture, identify the key components that need to be developed.
- Define Interfaces: Specify the interfaces through which components will interact with each other.
- Detailed Design: For each component, provide detailed design specifications including data structures, algorithms, and control logic.
- Interaction Design: Define how the components will interact with each other to achieve the desired functionality.

Tools and Techniques

Unified Modelling Language (UML): Use UML diagrams like class diagrams, sequence diagrams, and component diagrams to visualize component design.

Design Patterns: Apply standard design patterns to solve common design problems and improve component robustness.

Best Practices

- Single Responsibility Principle: Ensure each component has a single responsibility and does it well.
- Open/Closed Principle: Design components that are open for extension but closed for modification.
- Component Reusability: Design components that can be easily reused in different parts of the system or in future projects.

Challenges in Component Design

- Integration Complexity: Ensuring that all components work together seamlessly can be challenging.
- Balancing Cohesion and Coupling: Finding the right balance between highly cohesive and loosely coupled components.
- Scalability Issues: Designing components that can scale effectively as the system grows.

Component design is a foundational aspect of system development that ensures the system is modular, maintainable, and scalable. By focusing on creating well-defined, reusable, and loosely coupled components, developers can build robust and flexible systems.

Security Design

Security design is an essential aspect of system design that focuses on protecting the system and its data from unauthorized access, breaches, and other security threats. It involves implementing measures to ensure confidentiality, integrity, and availability of the system.

Key Concepts in Security Design

- Confidentiality: Ensuring that sensitive information is accessible only to those authorized to access it.
- Integrity: Ensuring that the information is accurate and has not been tampered with.
- Availability: Ensuring that the system and its data are available to authorized users when needed.
- Authentication and Authorization: Verifying the identity of users and ensuring they have the necessary permissions to perform actions.

Steps in Security Design

- Threat Modelling: Identify potential threats and vulnerabilities in the system.
- Security Requirements: Define security requirements based on the identified threats.
- Security Controls: Implement security controls such as encryption, firewalls, and intrusion detection systems.
- Access Control: Implement mechanisms for user authentication and authorization.
- Data Protection: Ensure that sensitive data is protected both in transit and at rest.

Tools and Techniques

- Encryption: Use encryption techniques to protect data.
- Firewalls: Implement firewalls to protect the system from unauthorized access.
- Intrusion Detection Systems: Use intrusion detection systems to monitor and respond to potential security threats.
- Security Testing: Conduct security testing to identify and address vulnerabilities.

Best Practices

- Least Privilege: Ensure that users and processes have the minimum level of access necessary to perform their functions.
- Regular Updates: Keep the system and its components up to date with the latest security patches.
- Secure Coding Practices: Follow secure coding practices to prevent vulnerabilities such as SQL injection and cross-site scripting.
- Incident Response Plan: Develop and implement an incident response plan to address security breaches.

Challenges in Security Design

- Evolving Threat Landscape: Keeping up with the constantly evolving threat landscape.
- Balancing Security and Usability: Ensuring that security measures do not hinder the usability of the system.
- Complexity: Managing the complexity of implementing and maintaining security controls.

Security design is a crucial aspect of system design that ensures the system is protected from various security threats. By implementing robust security measures and following best practices, developers can build secure systems that protect sensitive data and maintain user trust.

Performance Design:

Performance design focuses on ensuring that the system meets its performance requirements, such as response time, throughput, and resource utilization. It involves optimizing various aspects of the system to ensure it performs efficiently under different conditions.

Key Concepts in Performance Design

- Response Time: The time taken by the system to respond to a request.
- Throughput: The number of requests the system can handle in a given period.
- Scalability: The system's ability to handle increased load by adding resources.
- Resource Utilization: The efficient use of system resources such as CPU, memory, and storage.

Steps in Performance Design

- Identify Performance Requirements: Define the performance requirements based on user expectations and system goals.
- Performance Modelling: Create models to predict the system's performance under different conditions.
- Optimization Techniques: Apply optimization techniques to improve performance.
- Load Testing: Conduct load testing to evaluate the system's performance under expected and peak loads.
- Performance Monitoring: Continuously monitor the system's performance to identify and address performance issues.

Tools and Techniques

- Profiling Tools: Use profiling tools to identify performance bottlenecks.
- Caching: Implement caching strategies to reduce the load on the system.
- Load Balancing: Use load balancing to distribute the load across multiple servers.
- Database Optimization: Optimize database queries and indexing to improve performance.

Best Practices

- Efficient Algorithms: Use efficient algorithms and data structures to improve performance.
- Minimize Network Latency: Reduce network latency by optimizing data transfer.
- Scalable Architecture: Design a scalable architecture that can handle increased load.
- Performance Budgeting: Allocate performance budgets for different components to ensure they meet performance requirements.

Challenges in Performance Design

- Identifying Bottlenecks: Identifying and addressing performance bottlenecks can be challenging.
- Resource Constraints: Ensuring optimal performance within resource constraints.
- Dynamic Workloads: Handling dynamic workloads that vary over time.

Performance design is essential to ensure the system meets its performance requirements and provides a good user experience. By following best practices and using appropriate tools and techniques, developers can build systems that perform efficiently under various conditions.

Integration Design

Integration design focuses on ensuring that different components and subsystems of the system work together seamlessly. It involves defining the interfaces and protocols for communication between components and ensuring that they are integrated effectively.

Key Concepts in Integration Design

- Interoperability: Ensuring that different components can work together and exchange information.
- Data Exchange: Defining the format and protocols for data exchange between components.
- Middleware: Using middleware to facilitate communication between components.
- API Design: Designing APIs for components to interact with each other.

Steps in Integration Design

- Define Integration Requirements: Identify the integration requirements based on system architecture and component design.

- Design Interfaces: Define the interfaces and protocols for communication between components.
- Middleware Selection: Select appropriate middleware to facilitate communication.
- Data Mapping: Define data mappings and transformations for data exchange.
- Integration Testing: Conduct integration testing to ensure components work together seamlessly.

Tools and Techniques

- API Gateways: Use API gateways to manage and secure API interactions.
- Message Brokers: Use message brokers to facilitate asynchronous communication between components.
- Service Oriented Architecture (SOA): Implement SOA to enable integration of services.
- Enterprise Service Bus (ESB): Use ESB to manage communication between different services.

Best Practices

- Standard Protocols: Use standard protocols for communication to ensure interoperability.
- Modular Design: Design components to be modular and independent. Error Handling: Implement robust error handling mechanisms for communication failures.
- Documentation: Provide comprehensive documentation for APIs and interfaces.

Challenges in Integration Design

- Compatibility Issues: Ensuring compatibility between different components and subsystems.
- Complexity: Managing the complexity of integrating multiple components.
- Security Concerns: Ensuring secure communication between components.

Integration design is crucial to ensure that different components of the system work together seamlessly. By following best practices and using appropriate tools and techniques, developers can build systems that are interoperable, scalable, and efficient.

System Design

System testing design is a critical phase in the software development lifecycle that ensures the entire system meets its requirements and performs as expected. This phase involves a comprehensive and systematic approach to identify defects and ensure that the integrated components of the system function correctly. Effective system testing design helps in delivering a high-quality product by uncovering issues before the system goes live.

Key Concepts in System Testing Design

- Test Planning: The process of defining the scope, approach, resources, and schedule of intended testing activities. It provides a clear roadmap of what needs to be tested and how testing will be carried out.
- Test Cases: Detailed specifications of inputs, execution conditions, testing procedures, and expected results that are used to validate the functionality of the system.
- Test Automation: The use of specialized software tools to control the execution of tests and the comparison of actual outcomes with predicted outcomes. Automation helps in increasing the efficiency and coverage of testing.
- Regression Testing: A type of testing that ensures that new code changes do not adversely affect the existing functionalities of the system.
- Performance Testing: Testing conducted to evaluate the system's performance under expected and peak load conditions. Security Testing: The process of identifying vulnerabilities in the system to ensure that data and resources are protected from potential threats.

Steps in System Testing Design

- Requirement Analysis: Understanding the functional and non-functional requirements of the system to define the scope of testing.
- Test Strategy Development: Outlining the testing approach, tools, and techniques to be used, along with defining the test levels and types.
- Test Plan Creation: Documenting the test objectives, resources, schedule, deliverables, and risks involved in the testing process.
- Test Case Development: Creating detailed test cases based on the requirements and design documents. Each test case should include test data, steps for execution, and expected results.
- Environment Setup: Preparing the hardware, software, and network configurations required to execute the tests. This may involve setting up test servers, databases, and application instances.
- Test Execution: Running the test cases and logging the outcomes. This step involves manual testing, automated testing, and the use of testing tools.
- Defect Tracking and Management: Identifying, recording, and tracking defects using defect management tools. Ensuring that all defects are addressed and resolved.
- Regression Testing: Re-running test cases to verify that fixes do not introduce new defects into the system.
- Performance and Security Testing: Conducting specialized tests to ensure the system meets performance benchmarks and is secure against threats.
- Test Reporting and Analysis: Documenting the test results, analysing defects, and preparing test summary reports. This helps in evaluating the quality of the system and making informed decisions about its readiness for deployment.

Tools and Techniques

- Test Management Tools: Tools like JIRA, TestRail, and Quality Centre for planning, tracking, and managing test activities.
- Automation Tools: Selenium, QTP, and JUnit for automating functional and regression testing.
- Performance Testing Tools: JMeter, LoadRunner, and Gatling for conducting performance and load tests.
- Security Testing Tools: OWASP ZAP, Burp Suite, and Nessus for identifying security vulnerabilities.

Best Practices

- Early and Continuous Testing: Begin testing early in the development cycle and continue it throughout the lifecycle to catch defects early.
- Test-Driven Development (TDD): Develop tests before writing the actual code to ensure that the system meets the specified requirements.
- Automated Regression Testing: Automate regression tests to quickly identify issues after code changes.
- Comprehensive Test Coverage: Ensure that all functional and non-functional requirements are covered by test cases.
- Risk-Based Testing: Prioritize testing efforts based on the risk and impact of potential defects.
- Clear Documentation: Maintain clear and comprehensive documentation of test plans, test cases, and test results to facilitate communication and traceability.

Challenges in System Testing Design

- Complexity of Integration: Ensuring that all components work together seamlessly can be challenging, especially in large and complex systems.
- Resource Constraints: Limited resources (time, budget, and manpower) can impact the thoroughness and extent of testing.
- Changing Requirements: Adapting to changing requirements during the testing phase can complicate the testing process.
- Defect Management: Efficiently managing and resolving a large number of defects can be demanding.
- Performance and Security: Ensuring that the system meets performance benchmarks and is secure against a wide range of threats requires specialized skills and tools.

System testing design is a vital component of the software development process that ensures the quality and reliability of the system. By systematically planning and executing tests, identifying and addressing defects, and continuously improving the testing process, organizations can deliver robust and high-quality software products. Adopting best practices and leveraging appropriate tools can significantly enhance the effectiveness of system testing and contribute to the overall success of the project.

By focusing on these aspects of system testing design, developers and testers can ensure that the system not only meets its functional requirements but also performs reliably and securely in real-world conditions.

Chapter 6 Implementation

This chapter sets the stage for understanding the **Strategic Project Workflow System**, an online platform designed to streamline the management of projects, tasks, and team collaboration within organizations. It emphasizes the significance of the system in addressing common challenges in project tracking, task assignment, and workflow transparency.

Introduction

Strategic Project Workflow System emerges as a pivotal solution in the modern landscape of project and task management, addressing the core challenges associated with collaboration, task tracking, and workflow transparency. In an era driven by digital transformation and remote work dynamics, the need for a streamlined, centralized platform for managing team activities and project lifecycles has become increasingly critical. Recognizing this necessity,

Modules Description:

Modules are the structural partitions of the Strategic Project Workflow System, designed to simplify and organize the development process. By breaking down the system into well-defined modules, each handling a specific functionality, the implementation, debugging, and maintenance become more efficient and manageable.

On a broad perspective the functionalities in the project are constrained depending on whether or not the user is a registered member with the website. Further, the user should also be logged in to enjoy the complete range of the features available.

The different pages of the system are:

- Home page
- Sign in page
- Welcome page
- Project Selection Page
- Task Management Page
- Report Generation Page
 - 1. Home Page

Serves as the landing page, providing an overview of the platform and navigation to key sections like login or dashboard.

Strategic Project WorkFlow System				
WORKSPACE		ALL MEMBERS		
Create Project		ID	Full Name	E-mail
Projects	Members	M-75168	JANE DOE	janedoe@e-mail.com
Boards	Workflow Management App	M-25729	JOHN DOE	johndoe@e-mail.com

Fig 6.1 Home Page

```

import { useSelector } from "react-redux";
import HomepageRow from "../components/HomepageRow";

function Homepage() {
  const projects = useSelector((store) => store.projectSlice.projects);

  const activeUser = useSelector((store) => store.userSlice.activeUser);

  const userProjects = projects.filter((project) => project.members.find((member) => member.id === activeUser?.id));

  return (
    <section className="">
      <h1 className="text-xl font-semibold mb-2.5">PROJECTS</h1>
      <div className="overflow-x-auto">
        <table className="table-auto w-full text-center">
          <thead className="border-b border-slate-500">
            <tr>
              <th className="px-6 py-3 text-stone-800 uppercase">ID</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Title</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Due To</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Priority</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Status</th>
            </tr>
          </thead>
          <tbody>
            {userProjects.map((project) => (
              <tr key={project.id}>
                <td>{project.id}</td>
                <td>{project.title}</td>
                <td>{project.dueTo}</td>
                <td>{project.priority}</td>
                <td>{project.status}</td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    </section>
  );
}

```

Fig 6.1.1 Home Page-Code

- - 1. Sign in page

Allows registered users to log into the system securely to access their personalized workspace and features.

the password typed. It re-verifies the password by re-password method validation used in the page.

sign in'."/>

SIGN UP

FULL NAME
your fullname

E-MAIL ADDRESS
emailaddress@email.com

PASSWORD

CONFIRM PASSWORD

SIGN UP

already have an account? [sign in](#)

Fig 6.2.1 Sign up Page

```

import { useDispatch, useSelector } from "react-redux";
import { Link, useNavigate } from "react-router-dom";

import { registerUser } from "../features/user";
import { useState } from "react";
import { idGenerator } from "../utils/idGenerator";

function SignUp() {
  const dispatch = useDispatch();
  const users = useSelector((store) => store.userSlice.users);
  const navigate = useNavigate();

  const [fullName, setFullName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState("");

  const [fullNameErr, setFullNameErr] = useState("");
  const [emailErr, setEmailErr] = useState("");
  const [passwordErr, setPasswordErr] = useState("");
  const [confirmPasswordErr, setConfirmPasswordErr] = useState("");

  function handleRegister(e) {
    e.preventDefault();
  }
}

```

Fig 6.2.1.1 Sign up Page-Code

For registered user:

If user is already registered then they will be able to give enquire or upload any book by just Sign in on home page. It also allows the user to deal or interact with the admin.

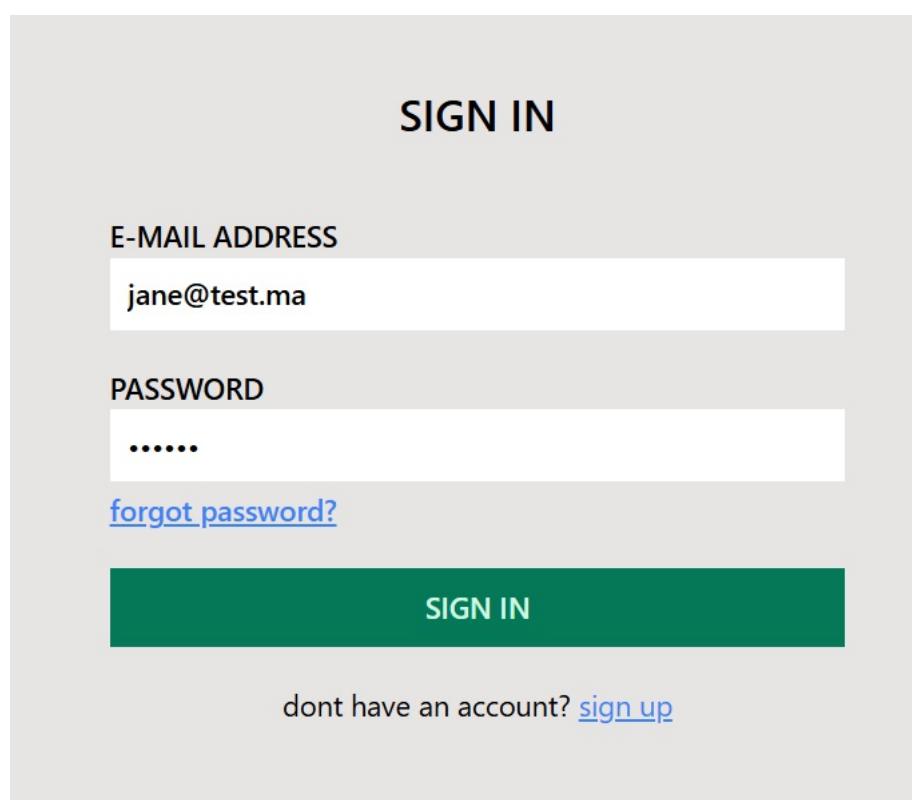


Fig 6.2.2 Sign in Page

```
unction Login() {
  const users = useSelector((store) => store.userSlice.users);
  const dispatch = useDispatch();
  const navigate = useNavigate();
  const [login, setLogin] = useState("jane@test.ma");
  const [pwd, setPwd] = useState("246810");
  const [user, setUser] = useState(null);

  const [loginError, setLoginError] = useState(false);
  const [pwdError, setPwdError] = useState(false);
  const [error, setError] = useState(false);

  function handleLogin(e) {
    setLoginError(false);
    setPwdError(false);
    setError(false);
    e.preventDefault();
    if (!login) setLoginError(true);
    if (!pwd) setPwdError(true);
    if (!(login && pwd)) return:
```

Fig 6.2.2.1 Sign in Page

- - 1. Welcome page

After successfully signing in, the user is redirected to the **Welcome Page**, which acts as the central dashboard for project and task management. From here, users can view ongoing projects, create new ones, and manage their existing tasks. The left-side navigation panel provides quick access to essential features

Strategic Project Workflow System – Simplifying Team Collaboration & Task Management

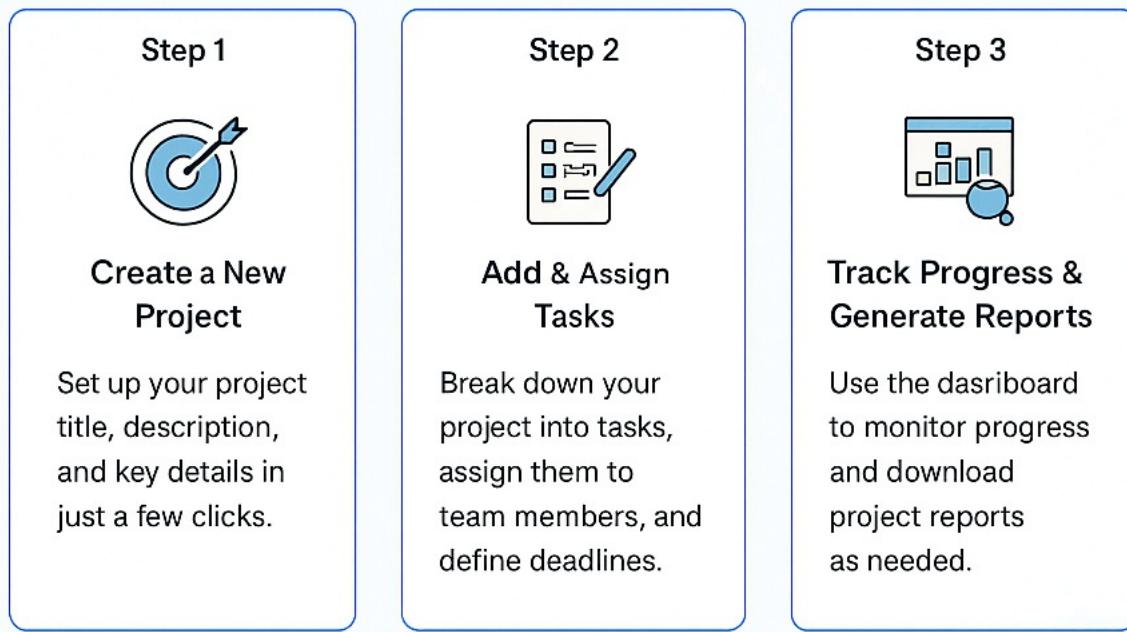
Are you a developer, manager, or part of an organization struggling with project tracking and task delegation? Do you often lose track of who's working on what or find it difficult to organize your team's workflow effectively? If yes, then *Strategic Project Workflow System* is designed just for you.

Strategic Project Workflow System is a modern, intuitive platform built using React.js and Spring Boot: tailored to help teams and individuals plan, track, and manage projects with clarity and control. Whether you're handling a solo project or managing a team of collaborators, our system gives you the tools to stay productive and aligned.

*Managing projects shouldn't be chaotic. Experience structured productivity with Strategic Project Workflow System.**

How to Start Managing Projects on Strategic Project Workflow System

You can streamline your task management in just 3 simple steps:



[Start Managing Projects](#)

Fig 6.3 Welcome Page

- - 1. Select Project page

Strategic Project WorkFlow System				
WORKSPACE	PROJECTS			
	ID	TITLE	CLIENT	REQUEST
Create Project	p-92549	Workflow Management App	John Doe	24-01-2024
Projects	p-54283	E-Presentation	Etablissment X	24-01-2024
Members				
BOARDS				
Workflow Management App				

Fig 6.4 Select Project Page

```
1 package filterbooks;
2
3 import jakarta.servlet.ServletException;
4
5 /**
6  * Servlet implementation class removefilter_college
7  */
8 @WebServlet("/removefilter_college")
9 public class removefilter_college extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     /**
13      * @see HttpServlet#HttpServlet()
14     */
15     public removefilter_college() {
16         super();
17         // TODO Auto-generated constructor stub
18     }
19
20     /**
21      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
22     */
23     protected void doGet(HttpServletRequest request, HttpServletResponse response)
24         throws ServletException, IOException {
25         // TODO Auto-generated method stub
26         HttpSession session = request.getSession();
27         session.removeAttribute("College"); // invalidate the session
28         session.removeAttribute("College_BookType");
29         session.removeAttribute("College_status");
30
31         response.sendRedirect("Home1.jsp");
32     }
33
34     /**
35      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
36     */
37     protected void doPost(HttpServletRequest request, HttpServletResponse response)
38         throws ServletException, IOException {
39         // TODO Auto-generated method stub
40         doGet(request, response);
41     }
42
43     /**
44      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
45     */
46     protected void doPost(HttpServletRequest request, HttpServletResponse response)
47         throws ServletException, IOException {
48         // TODO Auto-generated method stub
49         doGet(request, response);
50     }
51 }
52 }
```

Fig 6.4.1 Select Project Page-Code

1. Select Project

For the unregistered user, one can search for the projects by entering the project name, description, or assigned team in the search bar. By simply clicking on the search icon or pressing enter, the user can view the actual results in the form of a list of relevant projects.

Project Details: E-Presentation

PROJECT NAME

Workflow Management App

CLIENT NAME

John Doe

COMPANY NAME

Holding

DUE DATE

20-05-2024

STATUS

TODO

PRIORITY

URGENT

BUDGET

PROJECT DESCRIPTION

LOREM IPSUM.....

Fig 6.5.1 Select project

Workflow Management App

TODO

Build app layout

Add chatroom feature

+ New Item

IN PROGRESS

Creating file manipulation logic

+ New Item

DONE

+ New Item

Fig 6.5.2 Add task ,Report and Done

ADD NEW MEMBER

E-mail Address

mailaddress@e-mail.com

Send Invitation

Cancel

Fig 6.6 Adding member

```
function NewMemberModal({ setCreateNew }) {
  const [emailAddress, setEmailAddress] = useState("");

  return (
    <div className="absolute modal">
      <form onSubmit={(e) => e.preventDefault()} className="bg-stone-50 py-5 px-10 font-semibold m-a w-96">
        <h2 className="mb-5 uppercase text-center">Add New Member</h2>
        <div className="mb-4 space-y-2">
          <label htmlFor="">E-mail Address</label> <br />
          <input
            type="text"
            value={emailAddress}
            onChange={(e) => setEmailAddress(e.target.value)}
            placeholder="mailaddress@e-mail.com"
            className="py-2 px-2.5 outline-none border border-stone-300 w-full rounded-md bg-stone-50"
          />
        </div>
        <div className="space-y-2">
          <button className="bg-emerald-600 text-emerald-50 text-center py-2 w-full">Send Invitation</button>
          <button onClick={() => setCreateNew(false)} className="bg-red-500 text-red-50 text-center w-full">Cancel</button>
        </div>
      </form>
    </div>
  );
}
```

Fig 6.6.1 Add Member

Implementation:

In the implementation details section, we dive deep into the technical aspects of building the **Strategic Project Workflow System** using React.js and Spring Boot. This includes the implementation of key features such as task management, project tracking, user roles, real-time updates, notifications, and project analytics. Code snippets and explanations are provided to illustrate the implementation process, along with insights into best practices and optimization techniques.

Task and Project Management:

Details: The Strategic Project Workflow System integrates robust task and project management features to help teams stay organized and on track. This includes creating, assigning, and tracking tasks within projects. The system allows project managers and team members to create new tasks, assign them to relevant team members, and track progress through various stages.

PROJECT INFOS

Project Name*

Due Date*

dd-mm-yyyy

Budget*

Project Description*

CLIENT INFOS

Company*

Client Name*

Address*

Number Phone*

E-mail*

Additional Notes

SAVE

Fig 6.6.2 Task and project Management

```
import { useState } from "react";
import { useDispatch, useSelector } from "react-redux";
import { useParams } from "react-router";

import KanbanCol from "./KanbanCol";
import { addList } from "../features/projects";
import { idGenerator } from "../utils/idGenerator";

function ProjectBoard() {
  const { id } = useParams();
  const projects = useSelector((store) => store.projectSlice.projects);
  const project = projects.find((item) => item.id.toLowerCase() == id.toLowerCase());

  const [createList, setCreateList] = useState(false);
  const [ newList, setNewList ] = useState("");
  const dispatch = useDispatch();

  function handleCreateList(e) {
    e.preventDefault();

    if (!newList.trim()) return;

    const listID = idGenerator("L", 6);
    dispatch(addList({ projectID: id, list: { id: listID, label: newList, tasks: [] } }));
  }
}
```

Fig 6.7: Task and Project Management

Task Management :

Details: In the Strategic Project Workflow System, task management is a key feature designed to improve team collaboration and project execution. The To-Do list feature allows users to keep track, set priorities, and monitor progress.

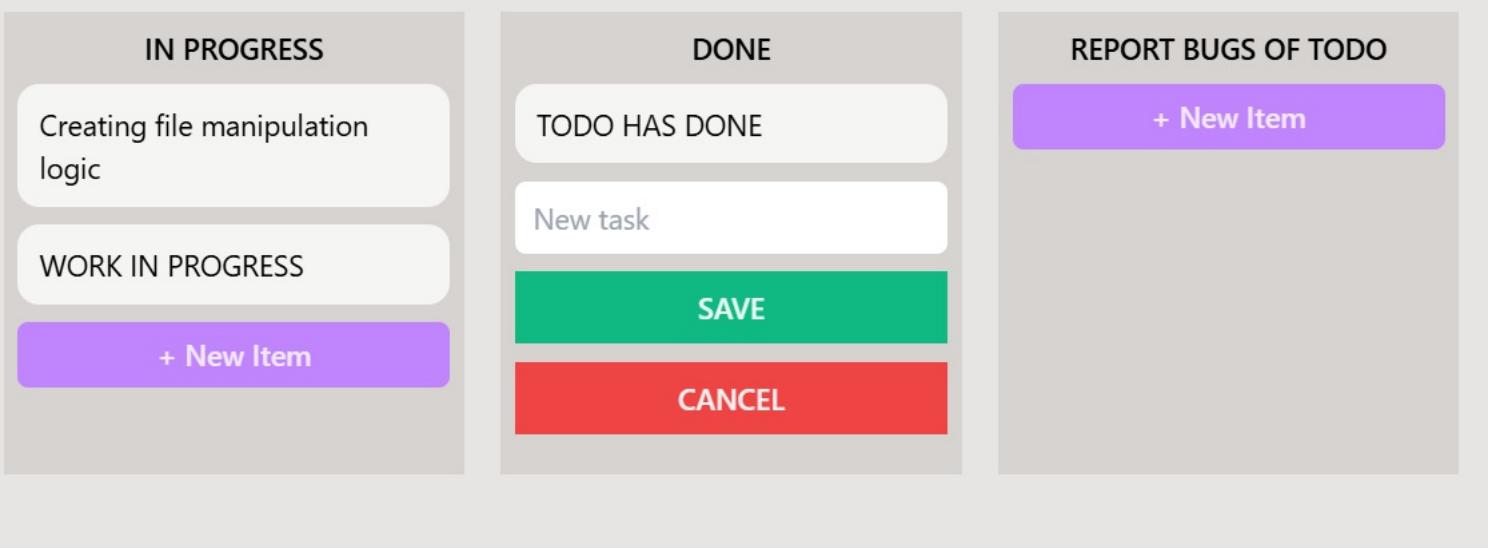


Fig 6.7.1: Task Management

Id Generator Code:

```
function idGenerator(prefix = "ID", length = 4) {
  if (length < 4) throw new Error("ID length must be equal or greater than 4");
  const lng = 10 ** length - 1;
  return `${String(prefix).toUpperCase()}-${String(parseInt(Math.random() * lng)).padStart(length, "0")}`;
}

export { idGenerator };
```

Fig 6.7.2: Id generator code

- 1. Creating Project:

```
import { useSelector } from "react-redux";
import ProjectRow from "./ProjectRow";

function Projects() {
  const projects = useSelector((store) => store.projectSlice.projects);
  console.log(projects);
  return (
    <section className="">
      <h1 className="text-xl font-semibold mb-2.5">PROJECTS</h1>
      <div className="overflow-x-auto">
        <table className="table-auto w-full text-center">
          <thead className="border-b border-slate-500">
            <tr>
              <th className="px-6 py-3 text-stone-800 uppercase">ID</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Title</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Client</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Requested At</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Due To</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Priority</th>
              <th className="px-6 py-3 text-stone-800 uppercase">Status</th>
            </tr>
          </thead>
          <tbody>
```

Fig 6.7.3 Project Creation Code

```

import { useState } from "react";
import { useDispatch, useSelector } from "react-redux";
import { useParams } from "react-router";

import KanbanCol from "./KanbanCol";
import { addList } from "../features/projects";
import { idGenerator } from "../utils/idGenerator";

function ProjectBoard() {
  const { id } = useParams();
  const projects = useSelector((store) => store.projectSlice.projects);
  const project = projects.find((item) => item.id.toLowerCase() == id.toLowerCase());

  const [createList, setCreateList] = useState(false);
  const [ newList, setNewList ] = useState("");
  const dispatch = useDispatch();

  function handleCreateList(e) {
    e.preventDefault();

    if (!newList.trim()) return;

    const listID = idGenerator("L", 6);
    dispatch(addList({ projectID: id, list: { id: listID, label: newList, tasks: [] } }));
    setNewList("");
  }
}

```

Fig 6.8: Project Board

```

import { useState } from "react";

function NewMemberModal({ setCreateNew }) {
  const [emailAddress, setEmailAddress] = useState("");

  return (
    <div className="absolute modal">
      <form onSubmit={(e) => e.preventDefault()} className="bg-stone-50 py-5 px-10 font-semibold m-a w-96">
        <h2 className="mb-5 uppercase text-center">Add New Member</h2>
        <div className="mb-4 space-y-2">
          <label htmlFor="">E-mail Address</label> <br />
          <input
            type="text"
            value={emailAddress}
            onChange={(e) => setEmailAddress(e.target.value)}
            placeholder="mailaddress@e-mail.com"
            className="py-2 px-2.5 outline-none border border-stone-300 w-full rounded-md bg-stone-50"
          />
        </div>
      </form>
    </div>
  );
}

```

Fig 6.8.1: New Member Model

- 1. Searching:

Details: The searching feature enables users to search for specific textbooks based on criteria such as title, author, edition, or ISBN. Study Emporium employs advanced search algorithms to retrieve relevant results quickly and accurately. Users can further refine their search results using filters and sorting options.

Summary: Searching enhances the user experience on Study Emporium by enabling users to find the textbooks they need quickly and easily. By providing robust search functionality, the platform empowers users to navigate the marketplace efficiently and locate desired books with minimal effort.

- 1. Task Assignment and Collaboration:

The task assignment feature allows project managers to allocate tasks to individual team members. Each task includes specific details such as title, description, priority, deadlines, and status. Assigned users can update task status in real-time and leave comments or updates for seamless communication.

Task assignment and collaboration tools streamline team communication and project execution.

```

function HomepageRow({ project }: { project: any; }) {
  const activeUser = useSelector((store) => store.userSlice.activeUser);
  return (
    <tr className="border-b-2 border-stone-300">
      <td className="px-6 py-3 text-stone-800 font-semibold">{project.id}</td>
      <td className="px-6 py-3 text-stone-700 font-semibold cursor-pointer">
        {activeUser.admin ? (
          <Link to={`/projects/${project.id}`}>{project.name}</Link>
        ) : (
          <Link to={`/projects/${project.id}/board`}>{project.name}</Link>
        )}
      </td>
      <td className="px-6 py-3 text-stone-700 font-semibold">{project.dueDate}</td>
      <td className="px-6 py-3 uppercase">
        <span className="inline-block bg-red-400 px-2.5 py-1.5 rounded-full">{project.priority}</span>
      </td>
      <td className="px-6 py-3 text-stone-700 font-semibold">
        <Link className="bg-green-400 px-2.5 py-1.5 bg-opacity-50 rounded-full" to={`/projects/${project.id}/board`}>
          {project.status}
        </Link>
      </td>
    </tr>
  );
}

export default HomepageRow;

```

Fig 6.10.1: Home Page Code

```

function MemberRow({ member }) {
  return (
    <tr className="font-semibold text-stone-700 border-b-2 border-stone-300">
      <td className="py-4">{member.id}</td>
      <td className="py-4 uppercase">{member.fullName}</td>
      <td className="py-4">{member.emailAddress}</td>
      <td className="py-4">On {member.enrolled}</td>
      <td className="py-4 space-x-2">
        <button className="bg-blue-400 text-blue-50 p-2">{member.admin ? "Drop Admin" : "Make Admin"}</button>
        <button className="bg-red-400 text-red-50 p-2">Leave X</button>
      </td>
    </tr>
  );
}

export default MemberRow;

```

Fig 6.11: Member Row Code

```

function NewClient() {
  return (
    <>
    <h1 className="font-semibold uppercase text-xl mb-2">CLIENT INFOS</h1>
    <form className="space-y-5">
      <div>
        <label className="mb-2 inline-block">Company</label>
        <input className="py-1 px-2.5 w-full" />
      </div>
      <div>
        <label className="mb-2 inline-block">Client Name</label>
        <input className="py-1 px-2.5 w-full" />
      </div>
      <div>
        <label className="mb-2 inline-block">Address</label>
        <input className="py-1 px-2.5 w-full" />
      </div>
      <div className="space-y-5">
        <div>
          <label className="mb-2 inline-block">Number Phone</label>
          <input type="date" className="py-1 px-2.5 w-full" />
        </div>
        <div>
          <label className="mb-2 inline-block">E-mail</label>
          <input type="mail" className="py-1 px-2.5 w-full" />
        </div>
      </div>
    </form>
  )
}

```

Fig 6.11.1: Adding New Client

- 1. Feedback Mechanisms:

Details: Feedback mechanisms allow users to provide feedback and ratings on their experiences with Study Emporium, including buying, selling, and overall user experience. Feedback is collected and displayed publicly to help users make informed decisions and improve the platform's reputation.

Summary: Feedback mechanisms promote transparency and trust on Study Emporium by enabling users to share their experiences and opinions. By soliciting and displaying feedback, the platform fosters accountability and encourages continuous improvement in user satisfaction and platform performance.

- 1. User Authentication and Authorization:

Details: User authentication mechanisms, such as username/password authentication or OAuth-based login, are implemented to verify user identities during signup and login processes. Upon successful authentication, users are granted access to specific features and functionalities based on their assigned roles and permissions.

Summary: User authentication and authorization mechanisms ensure that only authenticated users with appropriate permissions can access the platform's resources and functionalities. By verifying user identities and controlling access levels, Study Emporium maintains security and protects user data from unauthorized access.

- 1. Session Management:

In the Strategic Project Workflow System, session management is implemented to ensure secure and continuous user interactions across the application. Upon successful login, the system issues a JWT (JSON Web Token), which is securely stored on the client side (typically in local storage or HTTP-only cookies). This token is attached to subsequent API requests, allowing the backend to authenticate and authorize the user without requiring repeated logins.

Summary: Session management ensures a seamless and consistent user experience by preserving user sessions and maintaining stateful interactions with the platform. By managing user sessions effectively, Study Emporium enhances usability and convenience for users while maintaining security and privacy.

```
package com.zosh.config;

import jakarta.servlet.http.HttpServletRequest;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.DefaultSecurityFilterChain;
import org.springframework.security.web.authentication.www.BasicAuthenticationFilter;
import org.springframework.web.cors.CorsConfiguration;
import org.springframework.web.cors.CorsConfigurationSource;

import java.util.Arrays;
import java.util.Collection;
import java.util.Collections;

@Configuration
@EnableWebSecurity
public class AppConfig {
```

Fig 6.12: Session Maintain Code

- 1. Error Handling and Logging:

Details: Robust error handling mechanisms are implemented to detect, handle, and recover from errors or exceptions that may occur during the operation of the platform. Additionally, logging mechanisms are employed to record system events, errors, and user interactions for troubleshooting and analysis purposes.

Summary: Error handling and logging mechanisms enhance the reliability and maintainability of Study Emporium by identifying and addressing errors or issues in a timely manner. By logging system events and user interactions, the platform gains valuable insights into its performance and usage patterns, enabling continuous improvement and optimization.

```

try {
    Class.forName("com.mysql.jdbc.Driver"); // load the driver
    Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/studyemp", "root",
                                                "password123A$");
    if (con.isClosed()) {
        out.print("not connected.....");
    } else {
        out.print("connected.....");
    }
    String q = "select * from registration where gmail =? and password = ?";
    PreparedStatement pst = con.prepareStatement(q);
    pst.setString(1, Email);
    pst.setString(2, Password);

    ResultSet rs = pst.executeQuery();
    if (rs.next()) {
        session.setAttribute("name", rs.getString("first_name"));
        session.setAttribute("email", rs.getString("gmail"));
        dispatcher = request.getRequestDispatcher("Home.jsp");
    } else {
        request.setAttribute("status", "error");
        dispatcher = request.getRequestDispatcher("index.jsp");
    }
    dispatcher.forward(request, response);
} catch (Exception e) {
    e.printStackTrace();
}
}

        dispatcher.forward(r
try {
    Cla: } catch (Exception e) {
    Con:     e.printStackTrace();
}

```

Fig 6.13: Error Handling Code

- 1. Performance Optimization:

Details: Performance optimization techniques, such as caching, lazy loading, and asynchronous processing, are implemented to improve the speed and responsiveness of Study Emporium. Profiling tools and performance monitoring are used to identify and address performance bottlenecks in the system.

Summary: Performance optimization enhances the efficiency and scalability of Study Emporium, ensuring optimal performance even under high loads or increased user demand. By optimizing resource utilization and minimizing latency, the platform delivers a seamless and responsive user experience, leading to increased user satisfaction and engagement.

- 1. Data Validation and Sanitization:

Details: Data validation and sanitization techniques are employed to ensure the integrity and security of user input data. Validation libraries or frameworks are used to enforce data validation rules and sanitize input data before processing or storing it in the database, mitigating common vulnerabilities such as SQL injection or cross-site scripting (XSS) attacks.

Summary: Data validation and sanitization mechanisms protect Study Emporium from security threats and vulnerabilities by validating and sanitizing user input data before processing it. By enforcing data integrity and sanitization standards, the platform maintains the integrity of its data and safeguards against malicious attacks or exploits.

- 1. API Integration:

Details: Study Emporium integrates with external APIs or services to enhance its functionality and interoperability. Integration with payment gateways, shipping providers, or third-party services enables additional features such as online payments or order fulfillment, enhancing the overall user experience.

Summary: API integration expands the capabilities and utility of Study Emporium by leveraging external services and resources. By integrating with external APIs, the platform gains access to additional features and functionalities, enabling users to perform tasks more efficiently and effectively.



activation-1.1.jar
javamail-1.6.0.jar
mysql-connector-j-8.0.32.jar

Fig 6.14: Activation API

HTTP Request :

```

private CorsConfigurationSource corsConfigurationSource(){
    return new CorsConfigurationSource() {
        @Override
        public CorsConfiguration getCorsConfiguration(HttpServletRequest request) {
            CorsConfiguration cfg=new CorsConfiguration();
            cfg.setAllowedOrigins(Arrays.asList(
                "http://localhost:3000",
                "http://localhost:5173",
                "http://localhost:4200"
            ));

            cfg.setAllowedMethods(Collections.singletonList("*"));
            cfg.setAllowCredentials(true);
            cfg.setAllowedHeaders(Collections.singletonList("*"));
            cfg.setExposedHeaders(Arrays.asList("Authorization"));
            cfg.setMaxAge(3600L);

            return cfg;
        }
    };
}

```

Fig 6.15:handling request

```

@Configuration
@EnableWebSecurity
public class AppConfig {
    @Bean
    DefaultSecurityFilterChain securityWebFilterChain(HttpSecurity http) throws Exception {
        http.sessionManagement(Management -> Management.sessionCreationPolicy(SessionCreationPolicy
            .authorizeHttpRequests(Authorize -> Authorize
                .requestMatchers( ...patterns: "/api/**").authenticated()
                .anyRequest().permitAll()))
            .addFilterBefore(new JwtTokenValidator(), BasicAuthenticationFilter.class)
            .csrf(csrf -> csrf.disable())
            .cors(cors -> cors.configurationSource(corsConfigurationSource())));
    }

    return http.build();
}

```

Fig 6.15.1: token validation

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/mav-
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.4.4</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.zosh</groupId>
  <artifactId>StarategicProjectWorkFlowSystem</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>Starategic Project WorkFlow System</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection>
      <url>https://github.com/zosh/StarategicProjectWorkFlowSystem.git</url>
      <developerName>zosh</developerName>
      <email>zosh@zosh.com</email>
    </connection>
    <branch>main</branch>
    <tag>v0.0.1-SNAPSHOT</tag>
  </scm>

```

Fig 6.15.2 dependency

Some Main Code Functionalities of Study Emporium

1. User Authentication and Management

Purpose: To provide secure access to the platform and manage user sessions effectively.

Implementation:

- - User Registration: Collect user details such as name, email, and password. Verify the email using OTP as described above.
 - User Login: Authenticate users using their email and password.
 - Session Management: Maintain user sessions using Java Servlets and HTTP sessions to keep track of logged-in users.
 - Technologies Used: Java Servlets, JSP, MySQL Database

Process:

User Registration:

- - User fills out the registration form.
 - System sends an OTP to the user's email.
 - User enters the OTP, which is then verified.
 - User details are stored in the database upon successful verification.

User Login:

- - User enters email and password.
 - System checks credentials against the database.
 - A session is created for the user upon successful authentication.

Best Practices:

- - Hash passwords using a strong algorithm like bcrypt before storing them in the database.
 - Implement multi-factor authentication for additional security.
 - Use HTTPS to secure data transmission between the client and server.

1. OTP Validation

Purpose: To enhance security during user registration and password recovery, ensuring that only legitimate users can access the platform.

Implementation:

•

- Signup OTP: When a user registers on the platform, an OTP (One-Time Password) is generated and sent to the user's email address to verify their identity.
- Password Recovery OTP: If a user forgets their password, an OTP is sent to their registered email address to authenticate the password reset request.
- Technologies Used: JavaMail API, Java Servlets, MySQL Database

Process:

- - Generate OTP: A random OTP is generated using a secure random number generator.
 - Send OTP: The OTP is sent to the user's email using the JavaMail API.
 - Verify OTP: The user enters the OTP on the platform, which is then verified against the stored OTP in the database.
 - Registration Completion: If the OTP is valid, the user registration is completed. For password recovery, the user is allowed to set a new password.

Best Practices:

- - Use a secure random number generator for OTP generation to enhance security.
 - Ensure OTP expiration after a certain period to prevent misuse.
 - Encrypt OTPs stored in the database to protect sensitive data.

1. Book Listing and Management

Purpose: To allow users to list their books for sale with detailed information and manage these listings effectively.

Implementation:

- - Form Submission: Users can submit details about the books they want to sell, including title, author, edition, condition, and price.
 - Database Integration: Book details are stored in a MySQL database.
 - Validation: Ensure all required fields are filled out and the data is valid.
 - Technologies Used: Java Servlets, JSP, MySQL Database

Process:

- - Form Submission: User fills out a form with book details.
 - System validates the input data.
 - Valid data is stored in the database.
 - Book Management: Users can update or delete their listings.
 - System provides an interface for managing listed books.

Best Practices:

- - Validate all input data to prevent SQL injection and other security threats.
 - Provide a user-friendly interface for managing book listings.
 - Use transactions to ensure data integrity during updates and deletions.

1. Project Searching and Browsing

Purpose: To enable users to find books easily through search and browsing features.

Implementation:

- - Search Functionality: Users can search for books by title, author, edition, and condition.
 - Advanced Filters: Users can refine search results based on criteria such as proximity to colleges, price range, and book condition.
 - Database Queries: Optimize queries to ensure fast and accurate search results.
 - Technologies Used: Java Servlets, JSP, MySQL Database, AJAX

Process:

- - Search Functionality: User enters search criteria.
 - System performs a database query to find matching books.
 - Results are displayed to the user.
 - Advanced Filters: User applies filters to narrow down search results.
 - System updates the search query to include filter criteria.
 - Filtered results are displayed to the user.

Best Practices:

- - Use indexed columns in the database to speed up search queries.
 - Implement pagination to manage large result sets.
 - Provide user-friendly filters to enhance the search experience.

1. Selecting Projects

Purpose: To facilitate the purchase of listed books.

Implementation:

- - Book Details Page: Display detailed information about the book, including the seller's contact information.
 - Contact Seller: Users can reach out to the seller directly through the provided contact information.
 - Transaction Management: Maintain a record of completed transactions in the database.
 - Technologies Used: Java Servlets, JSP, MySQL Database

Process:

- **Project Details Page:**
 1. User views the details of a selected book.
 2. System displays the book information and seller contact details.
- **Client Contact:**
 1. User contacts the seller to negotiate the purchase.
 2. System logs the interaction for future reference.
- **Transaction Management:**
 1. Record transaction details in the database.
 2. Update the status of the book as sold.

Best Practices:

- Ensure secure communication between buyers and sellers.
- Provide a feedback mechanism for buyers to rate their experience.
- Keep transaction records for auditing and resolving disputes.

1. Selecting Projects

Purpose: To allow users to list their books for sale.

Implementation:

- Sell Section: Users can list books by providing detailed information.
- Book Status: Update the status of the book once it is sold.
- Notification System: Notify users when their book is sold or when there is an inquiry.
- Technologies Used: Java Servlets, JSP, MySQL Database

Process:

- **Sell Section:**
 1. User fills out a form to list a book for sale.
 2. System validates the input data and stores it in the database.
- Project Status:
 1. System updates the status of the book to sold once a transaction is completed.
- Notification System:
 1. Send notifications to the seller when their book is sold or there is an inquiry.
 2. Notify buyers of new listings that match their search criteria.

Security and Performance Optimization

Ensuring the platform is secure and performs well was a top priority:

Data Protection:

- Sensitive information is encrypted, and regular security audits are conducted to identify and fix vulnerabilities.
- Performance Tuning: Database queries are optimized to ensure quick response times, even under heavy load.
- Scalability: The platform is designed to scale, allowing it to handle an increasing number of users and transactions as it grows.

User Interface and Experience:

The user interface (UI) and user experience (UX) were carefully designed to be intuitive and engaging:

- **Responsive Design:** The platform is fully responsive, ensuring it works well on both desktop and mobile devices.
- **User-Friendly Navigation:** A clean and straightforward navigation structure helps users find what they need quickly.
- **Visual Appeal:** Attractive design elements make the platform visually appealing, enhancing user engagement.

LIMITATIONS

- Limited Availability:** Study Emporium may face challenges in ensuring comprehensive coverage of textbooks, as it relies on the participation of bookstores and individual sellers to list their inventory.
- Geographic Constraints:** The effectiveness of Study Emporium may be limited by geographical factors, as certain regions or institutions may have lower adoption rates or limited access to the platform.
- Quality Control:** Ensuring the quality and accuracy of listed textbooks can be challenging, as Study Emporium relies on user-generated content and may encounter issues with incorrect or misleading listings.
- Technical Challenges:** Maintaining and updating the platform requires ongoing investment in technology infrastructure and development resources, which may pose challenges in terms of scalability, reliability, and compatibility with evolving technologies.
- User Engagement:** Sustaining user engagement and activity on Study Emporium may be challenging, as users may face competing priorities or preferences for alternative platforms or methods of obtaining textbooks.
- Financial Sustainability:** Generating revenue through transaction fees or premium services may be necessary to sustain Study Emporium's operations and investments in growth and development.
- Regulatory Compliance:** Study Emporium must adhere to relevant laws and regulations governing e-commerce, data privacy, and consumer protection, which may vary across different jurisdictions and require ongoing monitoring and compliance efforts.
- Competition:** Study Emporium operates in a competitive landscape with other online marketplaces, textbook rental services, and educational technology platforms, which may offer similar or alternative solutions to students and bookstores.

RECOMMENDATIONS

To ensure the continuous improvement of the **Strategic Project Workflow System**, it is essential to regularly enhance its functionality and user experience. Future updates may include advanced analytics dashboards, AI-powered task recommendations, integration with third-party tools like Slack or Google Calendar, and improved real-time collaboration features.

Expanding the platform's reach by making it adaptable for different industries or enterprise use cases is another potential growth area. Promoting the system through digital marketing, partnerships

with organizations, and showcasing its effectiveness in workflow optimization can further increase adoption.

FUTURE SCOPE

In this project, the **Strategic Project Workflow System** aims to continuously evolve by incorporating innovative technologies and expanding its capabilities. Future enhancements will include the integration of **Artificial Intelligence** to provide personalized task recommendations and improve user experiences. To enhance **security** and ensure the integrity of data, **blockchain technology** will be explored for transparent and tamper-proof transactions. **Enhanced collaboration tools** will be developed to foster better communication and interaction among users, facilitating more efficient project execution. The system's scope will also expand to include additional features for project management, such as advanced **mobile applications** with offline functionality, making it easier for users to stay connected even when they don't have internet access. As part of its **global expansion**, the system will support multiple languages, ensuring accessibility for diverse, international teams. Furthermore, **gamification** and **rewards programs** will be introduced to increase user engagement and motivate teams to meet project milestones. By diversifying its offerings and continuously innovating, the Strategic Project Workflow System aims to become an indispensable tool for project management, supporting users throughout their entire workflow journey.

CONCLUSION

The **Strategic Project Workflow System** marks a significant step forward in the field of project management, offering a user-centric solution that addresses the needs of modern project teams. Throughout this project, we have explored the various facets of the system, from its initial design to its technical implementation, and analyzed its potential impact on organizations and team dynamics. In this conclusion, we summarize the key findings and insights gained from our work on the Strategic Project Workflow System and reflect on its potential for transforming project management practices.

In conclusion, the Strategic Project Workflow System represents a powerful tool for project management, providing teams with a seamless platform to create, assign, track, and collaborate on tasks. By leveraging advanced technologies like real-time updates, personalized workflows, and secure data management, the system enhances productivity, communication, and accountability within teams. As the system continues to evolve, it has the potential to reshape how organizations manage projects, optimize workflows, and drive team collaboration in an increasingly digital world.

Reflecting on the journey of the Strategic Project Workflow System, it is clear that its significance goes beyond simple task management—it represents a shift toward more efficient, interconnected, and flexible project environments. By offering tools that promote collaboration and streamline processes, the system is paving the way for more agile and adaptive project management practices.

Looking ahead, the journey of the Strategic Project Workflow System is just beginning. As technology continues to advance and user needs evolve, the system is well-positioned to adapt and innovate, driving positive change across industries. With its commitment to improving project management experiences and empowering teams, the system stands as a model of progress in the digital age.

In conclusion, the Strategic Project Workflow System is not just a tool—it is a testament to the power of technology to drive change and enhance collaboration. As we continue to improve and expand its features, the system holds the potential to reshape the future of project management, empowering teams to achieve greater success, one project at a time.

REFERENCES

1. Bassett, T.J. and Walter, C.M., 2001. Booksellers and Bestsellers: British Book Sales as Documented by The Bookman, 1891-1906. *Book History*, 4(1), pp.205-236.
2. CLUB, F., 2013. Used book sale.
3. Billington, S., 2005. Relief from Online Used Book Sales During New Book Launches. *Colum. JL & Arts*, 29, p.497.
4. Lorch, Y., 2014. Total Boox: a time for a shift: an emerging e-book vendor introduces a model that shifts the industry from "purchase-centric" to "reading-centric". *eContent Quarterly*, 1(4), pp.37-48.
5. Atlestam, I. and Myhre, R., 2010. Reading—what to purchase, and why?.
6. Chen, H., Hu, Y.J. and Smith, M.D., 2019. The impact of E-book distribution on print sales: analysis of a natural experiment. *Management Science*, 65(1), pp.19-31.
7. Bissett, C., 2013. A book sale how-to guide: more money, less stress.
8. Deschâtres, F. and Sornette, D., 2005. Dynamics of book sales: Endogenous versus exogenous shocks in complex networks. *Physical Review E*, 72(1), p.016112.
9. Chen, C.W. and Cheng, C.Y., 2013. How online and offline behavior processes affect each other: customer behavior in a cyber-enhanced bookstore. *Quality & Quantity*, 47, pp.2539-2555.
10. Lee, S., 2017. A Case Study of Bandi&Luni's Bookstore Using an Online to Offline (O2O) Service Design. *Journal of the Korea Industrial Information Systems Research*, 22(1), pp.117-126.
11. Koo, J.M., Ham, S.M., Kim, W.J., Shim, H.D. and Ryu, K.D., 2014. An Efficient Management Strategy of A Offline Second-Hand Bookstore With Camera Type OCR Technology. In *Proceedings of the Korean Society of Computer Information Conference* (pp. 283-286). Korean Society of Computer Information.
12. Li, M. and Shi, S., 2022, December. The Survival of Physical Bookstore Under Digital Transformation. In *2022 5th International Conference on Humanities Education and Social Sciences (ICHESS 2022)* (pp. 2265-2271). Atlantis Press.
13. Syed Alwi, S.F. and Da Silva, R.V., 2007. Online and offline corporate brand images: do they differ?. *Corporate Reputation Review*, 10, pp.217-244.
14. DI, Q., 2017. PHYSICAL BOOKSTORE DEVELOPMENT STRATEGY AND CUSTOMER CONSUMER BEHAVIORAL ANALYSIS A CASE STUDY OF TAIWAN ESLITE BOOKSTORE. *E-Proceeding*, 2017, p.101.
15. Sornette, D., Deschâtres, F., Gilbert, T. and Ageon, Y., 2003. Endogenous versus exogenous shocks in complex networks: an empirical test using book sale ranking. *arXiv preprint cond-mat/0310135*.
16. <https://link.springer.com/article/10.1007/s12109-000-0005-9>
17. <https://ieeexplore.ieee.org/abstract/document/6779375/>
18. <https://ieeexplore.ieee.org/abstract/document/7019651>
19. <https://ieeexplore.ieee.org/abstract/document/10063065>
20. https://link.springer.com/chapter/10.1057/9781137595775_11

Atul Chaurasiya

 [LinkedIn](#)  [LeetCode](#)  [GitHub](#)  +91-6306545105  atulchaurasiya66602@gmail.com  Lucknow, Uttar Pradesh

TECHNICAL SKILLS & INTERESTS

- **Soft Skills:** Verbal and Written Communication, Team Collaboration, Task Prioritization and Multitasking, Adaptability and Flexibility, Time Management.
- **Languages:** Java, GoLang, JavaScript
- **Frontend Technologies:** HTML, CSS, JavaScript, React.js, Tailwind CSS
- **Backend Technologies:** Gin, Gorilla Mux, GORM
- **Development Tools:** IntelliJ IDEA, VS Code, Git & GitHub

- **Development Tools:** IntelliJ IDEA, VS Code, Git & GitHub
- **Databases:** SQL
- **Relevant Coursework:** Data Structures & Algorithms, Operating Systems, Object-Oriented Programming, Database Management Systems.

PROJECTS

- **Calorie Tracker App**
- **Tech Stack:** React.js, Redux, , Gin, JWT Authentication ,Bootstrap.
 - Developed a full-stack Calorie Tracker App to allow users to log food items and track their daily calorie intake.
 - Built secure and scalable RESTful APIs using Golang (Gin framework) for backend operations.
 - Designed a responsive and dynamic frontend using React.js and Bootstrap for a smooth user experience.
 - Implemented core features like user authentication, food management (add/edit/delete), and daily calorie progress monitoring.
- **Doctor Booking Appointment**
 - Built a doctor appointment booking platform using React.js and Tailwind CSS, improving accessibility for 1000+ daily users.
 - Optimized UI performance, reducing load times by 40% and enhancing responsiveness across devices.
 - Implemented React hooks for efficient state management, enabling seamless real-time updates.
- **Weather Application**
 - Developed a real-time weather application with city-based and geolocation search, serving 500+ daily users.
 - Integrated OpenWeatherMap API for live, accurate weather data updates, enhancing user engagement.
 - Reduced redundant API calls by 30%, optimizing performance and decreasing load times by 25%.

TRAINING AND INTERNSHIP

- **CodeGrip (Training)** Nov 2023–Feb 2024
 - Explored core Java concepts, including loops, arrays, and strings.
 - Studied SOLID principles, object-oriented programming, exception handling, multithreading, collections, and generics.
 - Applied key concepts in a project, focusing on OOP, exception handling, and collections.
 - Enhanced problem-solving skills through hands-on coding assignments and real-world case studies.
 - Gained exposure to industry best practices for writing clean, maintainable, and scalable Java code.

EDUCATION

- **Bansal Institute of Engineering and Technology** Oct 2021–Jun 2024
B.Tech in Computer Science and Engineering CGPA: 7.8
- **H R Inter College** Apr 2018–Jun 2021
Higher Secondary Certificate Percentage: 73%
- **H R Inter College** Apr 2016–Jun 2018
Senior Secondary Certificate Percentage: 72%

ACHIEVEMENTS

- Solved 300+ problems on LeetCode .
- **Winner** - Java Pattern Printing Competition. Secured first place with an optimized solution.

ASHISH RATHORE

Lucknow, Uttar Pradesh

📞 +91-8707878094 📩 rathoreashish778@gmail.com 💬 LinkedIn 🌐 Github

EDUCATION

Bansal Institute of Engineering And Technology
B.Tech - **CGPA - 7.32**

Oct – 2021 May 2023
Lucknow, India

COURSEWORK / SKILLS

- Data Structures & Algorithms
- Operating Systems
- Computer Network
- OOPS Concept
- Machine Learning

PROJECTS

Bluetooth Home Automation ↗ | Arduino UNO, ESP32,C++ Programming

May 2023

- Designed and implemented a Bluetooth-controlled LED blinking system for smart home applications.
- Used microcontrollers (such as Arduino/ESP32) to establish a wireless connection for real-time LED control via a mobile app.
- Wrote efficient code to handle Bluetooth commands and automate LED operations, enhancing user convenience and energy efficiency.

Salesforce Web Page ↗ | Html,Css

Dec 2022

- Creating a clone web page of salesforce using HTML and CSS which improve me some basic html ans css skill.
- In this project I use some basic tags of HTML and some CSS styling Displayflex,Div container etc .

Bank Management System ↗ | C++

May 2023

- Designed and developed a console-based Bank Management System to manage basic banking operations such as account creation, deposits, withdrawals, balance inquiries, and account updates.
- Improved understanding of object-oriented programming concepts like classes, objects, and inheritance.

TRAINING

Sofcon ↗

Trainee

2022

Lucknow, India

- Completed training in HTML, CSS, and JavaScript from Sofcon, gaining hands-on experience in front-end development.
- Learned to create responsive and visually appealing web pages using modern CSS techniques.
- Developed interactive and dynamic web applications using JavaScript, enhancing user experience with DOM manipulation and event handling.

TECHNICAL SKILLS

Programming Languages C , C++ , Java .

Database MySql (Basics)

Developer Tools: VS Code , IntelliJ Idea , Arduino IDE , Proteus

Web Technologies: HTML , CSS , Javascript, React.js (Basic Understanding)

CERTIFICATIONS

- C++ - Techvidvan
- Core Java - CodeGrip
- HTML, CSS, Javascript - Sofcon
- SQL - DataFlair
- Data Science Using Python - UPTEC

Kuldeep Giri

Email:- kuldeepgiri9792@gmail.com, Git:- <https://github.com/code-KuldeepGiri>, Mob:- 9792951797,
Link:- <https://www.linkedin.com/in/kuldeep-giri-aa78992a8/>

EDUCATION

Bansal Institute of Engineering and Technology, Lucknow	U P
Computer Science and Engineering	2021-2025
Intermediate	U P
C.D.I.C inter college	2019-2021
Percentage:- 79	

KEY SKILLS

HTML	CSS	JavaScript	Bootstrap
Java	DSA	SQL	Python

PROJECT

Tic-Toc-Toe-Game:-

- Using HTML, CSS, and Java Script to ensure a seamless user experience on all devices, prioritizing accessibility.
- Clear and concise navigation menus and search functionality, facilitating easy access to products and information, thus optimizing user satisfaction and retention
- Crafted engaging product descriptions with a focus on clarity and SEO optimization, enhancing visibility and driving conversions effectively.
- Link:- <https://code-kuldeepgiri.github.io/Tic-Toc-Toe-Game/>

E-Commerce Website:-

- Proficient in crafting responsive e-commerce websites using HTML, CSS, and JavaScript.
- Ensured seamless user experience across various devices and screen sizes to maximize accessibility and engagement.
- Ensured compatibility with multiple browsers (Chrome, Firefox, Safari, etc.) to reach a broader audience and maintain consistent functionality.
- Link:- <https://code-kuldeepgiri.github.io/Animation-Website/>



Abhishek Kumar

Engineering Student

A self-motivated Engineering student who loves to solve problems on coding platforms and developing innovative tech solutions.

akumar49323@gmail.com

linkedin.com/in/abhishek-kumar783

7481932579

github.com/Abhishek140820

EDUCATION

B.Tech

Bansal Institute of Engineering and Technology, Lucknow

08/2021 - 05/2025

lucknow, uttar pradesh

cse

- DBMS, OS ,OOPs compute network

Intermediate

C.B.I inter college

08/2018 - 02/2020

siwan, Bihar

TRAINING AND INTERNSHIP

Training

CodeGrip , Lucknow

10/2023 - 12/2023

lucknow

Achievements/Tasks

- Learned basic concepts of core Java like : Variables, Operators, Data Types, Loops, Arrays
- Aquaire the knowledge about Object Oriented Programming Principles, Multithreading, and Exception Handling
- Worked on project using OOPs concepts ,Variables, Loops and Exception Handling

Training

Sofcon India Pvt Ltd

09/2022 - 10/2022

lucknow

Achievements/Tasks

- Aquaire proficiency in HTML structure, tags and attributes
- Gained expertise in javascript concepts such as variables , Data Types, Loops, Arrays DOM manipulation
- Worked on project using HTML tags and attributes
- Applied knowledge of String, Arrays , DOM manipulation event handling in project implementation

SKILLS



PERSONAL PROJECTS

Weather-App

- Developed a weather application that delivers real-time weather information for global cities using HTML, CSS, and JavaScript.
- Integrated with a weather API to display current temperature, humidity, wind speed, and weather conditions, along with daily forecasts.
- Technologies Used: HTML, CSS, JavaScript, REST API integration.

Tic Toc Toe Game

- Developed using HTML, CSS, and JavaScript to ensure a seamless user experience on all devices, prioritizing accessibility and engagement.
- Implemented clear and concise navigation menus and search functionality, facilitating easy access to products and information, thus optimizing user satisfaction and retention.
- Crafted engaging product descriptions with a focus on clarity and SEO optimization, enhancing visibility and driving conversions effectively.

CERTIFICATES

Core Java (CodeGrip)

Certified in Core Java, demonstrating comprehensive knowledge and proficiency in fundamental Java programming concepts and syntax.

HTML and JavaScript (Sofcon India Pvt.Ltd)

Learned the HTML and JavaScript Concepts such as HTML Structure ,tags and attributes and Arrays , String, DOM , Event Handling

Data Science using Python (UPTEC Computer Consultancy Ltd.)

Learned python and various libraries such as NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn .

LANGUAGES

Hindi
Full Professional Proficiency

English
Full Professional Proficiency

StudyE
ORIGINALITY REPORT

20

%

SIMILARITY INDEX

17%

INTERNET SOURCES

2%

PUBLICATIONS

10%

STUDENT PAPERS

PRIMARY SOURCES

[**www.slideshare.net**](http://www.slideshare.net)

1

6

Internet Source %

[**5www.easychair.org**](http://www.easychair.org)

2

1

Internet Source %

Submitted to HCUC

3

1

Student Paper %

[**www.coursehero.com**](http://www.coursehero.com)

4

1

Internet Source %

fastercapital.com

5

1

Internet Source %

pdfcoffee.com

6

1

Internet Source %

theses.cz

7

1
Internet Source %

Submitted to University of East London

8

1

Student Paper %

9 Submitted to Higher Education Commission Pakistan

1%

Student Paper

10

11

12

13

14

15

16

17

18

19

20

Submitted to Alliance University

Student Paper

Submitted to University of Glamorgan

Student Paper

Submitted to Latvia University of Life Sciences and Technologies

Student Paper

[**www.grin.com**](http://www.grin.com)

Internet Source

Submitted to Tower Hamlets College

Student Paper

Submitted to National College of Ireland

Student Paper

[**www.freecodecamp.org**](http://www.freecodecamp.org)

Internet Source

Submitted to The University of Texas at Arlington

Student Paper

Submitted to University of Colombo

Submitted to Colorado Technical University

Student Paper

Submitted to Liberty University

Student Paper

www.blogarama.com

Internet Source

Submitted to Sim University

Student Paper

www.studocu.com

Internet Source

Submitted to Coventry University

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

33

34

35

36

37

Student Paper

fdp.aktu.ac.in

Internet Source

users.utcluj.ro

Internet Source

Submitted to The Hong Kong Polytechnic University

Student Paper

github.com

Internet Source

silo.tips

Internet Source

<1%
<1%
<1%
<1%
<1%
38

[**www.algorithmicbotany.jungletest.cpsc.ucalgary.ca**](http://www.algorithmicbotany.jungletest.cpsc.ucalgary.ca)

Internet Source

<1%
39
40
41
42

Submitted to Aston University

Student Paper

Babeş-Bolyai University

Publication

Submitted to Billy Blue Group

Student Paper

Submitted to De Montfort University

Student Paper

<1%
<1%
<1%
<1%
43
44
45
46
47
48
49
50
51
52

Deepak Bhandari, Manavjeet Kaur. "Advanced Security through Biometric Systems and Reporting Techniques", Journal of Advances in Mathematics and Computer Science, 2018

Publication

Submitted to University of Wales Institute, Cardiff

discovery.researcher.life

Internet Source

satprnews.com

Internet Source

www.indev-project.eu

Internet Source

webo.digital

Internet Source

www.digitalocean.com

Internet Source

www.nextgenerationinfrastructures.eu

Internet Source

Submitted to City University of Hong Kong

Student Paper

Submitted to Institute of Management Technology

Student Paper

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

53

54

55

56

57

58

59

60

61

62

63

64

Submitted to University of Brighton

dspace.daffodilvarsity.edu.bd:8080

Internet Source

escape.tech

Internet Source

hdl.handle.net

Internet Source

tudr.thapar.edu:8080

Internet Source

www.mdpi.com

Internet Source

Submitted to The University of Memphis

Student Paper

appsgeyser.com

Internet Source

open-innovation-projects.org

Internet Source

pdfcookie.com

Internet Source

spectrum.library.concordia.ca

Internet Source

www.etudier.com

Internet Source

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

<1%

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: Assignment title: Submission title:

File name: File size: Page count: Word count:

Character count: Submission date: Submission ID:

RISHU KUMAR 01

StudyEmp.docx StudyEmp.docx 5.22M

53

9,724

57,385

16-May-2024 09:33AM (UTC+0300)

2304843888

**A Project Report
on
STUDY EMPORIUM
to Submitted in Fulfilment of the
Requirements for the Degree of**

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Rishu Kumar (Roll No.-2004220100089)
Shivakant Shukla (Roll No.-2004220100102)
Vikrami Chaudhary (Roll No.-2004220100122)
Vishal Gupta (Roll No.-2004220100123)

Under the supervision of

Dr. Rohitashwa Pandey
Associate Professor & Head of Department



BANSAL INSTITUTE OF ENGINEERING AND
TECHNOLOGY LUCKNOW

Affiliated to
DR. APJ ABDUL KALAM TECHNICAL
UNIVERSITY
2023-24



Copyright 2024 Turnitin. All rights reserved.