E-NOTES

A Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF COMPUTER APPLICATION

by

HARSHIT SINGHAL
(University Roll No. 2200290140067)
HARSH KUMAR SINGH
(University Roll No. 2200290140066)

Under the Supervision of
Prof. RABI N. PANDA
(ASSOCIATE PROFESSOR)

Submitted
to the

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY (Formerly Uttar Pradesh Technical University) LUCKNOW

(MARCH, 2024)

CERTIFICATE

Certified that **Harshit Singhal (2200290140067), Harsh Kumar Singh(2200290140066)** has carried out the project work having “E-Notes” (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the students themself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date: 15-FEB-2024

HARSHIT SINGHAL (2200290140067)
HARSH KUMAR SINGH (2200290140066)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 15-FEB-2024

Prof. RABI N. PANDA
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. ARUN TRIPATHI
Head
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

E-Notes
Harshit Singhal
Harsh Kumar Singh

ABSTRACT

The E-notes project is a comprehensive web-based note-taking application built using JSP (JavaServer Pages) and Servlets, with a primary focus on providing users with a secure and efficient platform for managing personal notes. This project encompasses essential features such as user registration, login authentication, and the ability to create, update, and delete notes seamlessly. Leveraging JDBC (Java Database Connectivity), the application ensures robust interaction with the underlying database, facilitating efficient data storage and retrieval. Emphasizing both user experience and data security, E-notes prioritizes simplicity and effectiveness in its design, offering users a user-friendly interface to organize and access their notes securely. Through the integration of JSP, Servlets, and JDBC, coupled with meticulous attention to user interface and data security, E-notes presents a compelling solution for individuals seeking a reliable online platform for note-taking and management.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Prof. Rabi N. Panda** for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

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

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions and my team partner to develop the entire project alongside.

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

HARSHIT SINGHAL

HARSH KUMAR SINGH

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

The E-notes project is a web-based note-taking application developed using JSP (JavaServer Pages) and Servlets, designed to provide users with the ability to register, log in, create, update, and delete their personal notes. Upon registration, users can securely log in to the application to manage their notes, which are stored in a database. The application uses JDBC (Java Database Connectivity) for database interaction, ensuring that user data is stored and managed efficiently. With a focus on user experience and data security, E-notes offers a simple yet effective platform for managing personal notes online.

1.1 **Key Features**

1.1.1 **Registration:** Users can create an account by providing their details such as username, email, and password. This information is typically stored in a database after being validated for uniqueness and correctness.

1.1.2 **Login:** Once registered, users can log in using their credentials. The application verifies the username and password against the stored data in the database.

1.1.3 **Notes Management:** After logging in, users can create new notes, view existing notes, update their notes, and delete notes they no longer need. Each note is associated with the user who created it, allowing for personalized note management.

1.1.4 **Database Integration:** The application uses a database to store user information (like username, email, and password) and notes. The database is typically accessed using JDBC (Java Database Connectivity) to perform CRUD (Create, Read, Update, Delete) operations on user data and notes.

1.1.5 **Security:** To ensure the security of user data, the application should implement measures such as password hashing (using algorithms like bcrypt) to store passwords securely, session management to handle user sessions securely, and input validation to prevent common security vulnerabilities like SQL injection and cross-site scripting (XSS) attacks.

1.1.6 **Frontend:** The frontend of the application is built using JSP for dynamic content generation and HTML/CSS for styling. JavaScript may also be used for client-side validation and interaction.

1.1.7 **Deployment:** Once the application is developed, it needs to be deployed on a web server such as Apache Tomcat to make it accessible over the internet.

1.2 Project Description

The E-notes project is a web-based application that aims to provide users with a convenient platform for managing their personal notes. Developed using JSP (JavaServer Pages) and Servlets, the project offers functionalities such as user registration, authentication, and note management. Upon visiting the application, users are presented with a user-friendly interface where they can register for a new account by providing basic details such as username, email address, and password. The registration process includes validation checks to ensure data accuracy and security.

Once registered, users can log in to their accounts using their credentials. The application verifies the user's input against the stored data in the database to authenticate them. After successful authentication, users gain access to their personal dashboard, where they can create new notes, view existing notes, update the content of their notes, and delete notes they no longer need. Each note is associated with the user who created it, ensuring privacy and personalized note management.

Behind the scenes, the application uses JDBC to interact with a relational database management system (RDBMS) such as MySQL or PostgreSQL. User information, including usernames, email addresses, and hashed passwords, is stored securely in the database. Similarly, the notes created by users are also stored in the database, allowing for efficient retrieval and management. The application incorporates security best practices such as password hashing to protect user credentials and prevent unauthorized access.

Overall, the E-notes project showcases the use of JSP and Servlets to create a dynamic web application with user registration, authentication, and note management features. It emphasizes user experience and data security, providing a reliable platform for users to store and organize their notes online.

1.3 Project Scope

The scope of the E-notes project encompasses the development of a robust web application that allows users to create, store, update, and delete their notes in a secure and user-friendly manner. The project aims to provide a seamless user experience by offering features such as user registration and authentication, ensuring that each user has a personalized space for managing their notes. The application will be developed using JSP (JavaServer Pages) and Servlets, leveraging their capabilities for dynamic content generation and server-side logic.

The project scope includes the implementation of a database system, such as MySQL or PostgreSQL, to store user information and notes. This database will be accessed using JDBC (Java Database Connectivity) to perform CRUD (Create, Read, Update, Delete) operations on user data and notes. The application will prioritize data security by employing best practices such as password hashing to store user passwords securely and prevent unauthorized access.

Additionally, the project will focus on scalability and maintainability to accommodate potential future enhancements and updates. This includes designing the application architecture in a modular and extensible manner, allowing for the seamless integration of new features and improvements. The scope also involves thorough testing of the application to ensure its reliability and robustness, covering aspects such as functional testing, security testing, and performance testing. Overall, the E-notes project aims to deliver a well-designed and feature-rich note-taking application that meets the needs of its users while adhering to high standards of security and usability.

1.4 Hardware / Software Used In Project

The E-Notes will involve a combination of hardware and software components to ensure its development, deployment and functionality. Here is detailed List:

1.4.1 Server-side Hardware:

1. RAM (Random Access Memory):

- 8GB to 16GB (for moderate-sized application and user load).

- Consider higher capacities (e.g., 32GB or more) for scalability and handling a large number of concurrent users.

2. ROM (Storage):

- SSD storage for faster read and write operations.
- Allocate storage based on the application codebase, database size, and media storage requirements.

3. Processor:

- Multi-core processor (quad-core or higher) for efficient handling of concurrent user requests.

4. Operating System:

- Windows-based operating system (e.g., Windows Server 2012, Windows Server 2016) for stability and performance.

5. Network Equipment:

- Network infrastructure to facilitate secure data transfer between users and the server.

1.4.2 Database Server:

1. RAM:

- 16GB or more for efficient handling of concurrent database queries.

2. ROM(Storage):

- SSD storage for faster data retrieval.
- Allocate storage based on the anticipated size of the database and data storage needs.

2. Processor:

- Multi-core processor with sufficient processing power for complex database operations.

4. Operating System:

- Windows-based operating system for the database server.

1.4.3 User Devices:

1. Smartphones/Tablets:

- Compatibility with iOS and Android operating systems.
- Optimization for various screen sizes and resolutions.

2. Web Browsers:

- Compatibility with major web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

1.4.4 Development Environment:

1. Programming Languages:

- Backend: Jsp, Servlet or another suitable language.
- Frontend: HTML5, CSS3, Bootstrap.

2. Framework:

- Web application framework (e.g., Eclipse,IntelIj) for backend development

3. Database Management System:

- Choose a suitable DBMS (e.g., MySQL, PostgreSQL, Sql Server) for efficient data storage and retrieval.

4. Authentication and Authorization:

- Implement secure authentication protocols (RBAC) and authorization mechanisms.

5. APIs:

- Develop APIs to enable communication between the frontend and backend components.

6. Version Control:

- Version control system (e.g., Git) for managing and tracking changes in the source code.

7. Integrated Development Environment (IDE):

- IDEs such as Eclipse or IntelliJ IDEA for coding and debugging.

8. Containerization:

- Containerization tools like Docker for efficient deployment and scalability.

9. Continuous Integration/Continuous Deployment (CI/CD) :

- CI/CD pipelines (e.g., Jenkins, Travis CI) for automated testing and deployment.

10.Security Tools:

- Integrate security tools and practices to ensure the application's resilience against potential threats.

11. Monitoring Tools:

- Monitoring tools (e.g., APM, New Relic) for tracking application performance.

12. Collaboration Tools:

- Collaboration tools (e.g., Google Meet, Microsoft Teams) for effective communication among project team members.

CHAPTER 2

FEASIBILITY STUDY

2. INTRODUCTION

The introduction of the E-Notes project aims to revolutionize note-taking by providing users with a convenient and versatile digital platform. This feasibility study is conducted to evaluate the viability of the project from technical, economic, and operational perspectives. The project's objectives include developing a user-friendly platform for note-taking, implementing secure user authentication and authorization mechanisms, providing functionalities for note creation, viewing, updating, and deletion, ensuring compatibility across devices and operating systems, and enabling users to share their notes. The project scope encompasses the development of a web-based platform with user registration and login functionality, a note-taking interface with editing capabilities, features for sharing notes, and a secure database for storing user data and notes. Compatibility testing across multiple devices and browsers is also included within the scope.

From a technical standpoint, the E-Notes project is feasible due to the availability of web development technologies like HTML, CSS, JavaScript, and databases. Skilled developers and resources are accessible for implementation, and the project can leverage existing systems and technologies. Economically, the project needs careful cost estimation for development, maintenance, and support, along with consideration of potential revenue streams such as subscription fees, ads, or premium features. Operational feasibility depends on user acceptance and usability testing, as well as training requirements for users and support staff. The project also needs to integrate seamlessly with existing workflows and processes to ensure operational success.

2.1 Key Objectives

2.1.1 **User-Friendly Platform:** Develop a digital platform that is intuitive and easy to use for note-taking and management.

2.1.2 **Secure Authentication:** Implement robust user authentication and authorization mechanisms to ensure data security and user privacy.

2.1.3 **Note Management:** Provide functionalities for users to create, view, update, and delete their notes efficiently.

2.1.4 **Cross-Device Compatibility:** Ensure that the platform is accessible and works seamlessly across various devices and operating systems.

2.1.5 **Sharing Capabilities:** Enable users to share their notes with others, promoting collaboration and information exchange.

2.1.6 **Database Security:** Implement a secure database system to store user data and notes, protecting them from unauthorized access.

2.1.7 **Scalability:** Design the platform to be scalable, accommodating a growing user base and increasing data volumes.

2.2 Technical Feasibility

The E-Notes project demonstrates strong technical feasibility due to the availability of well-established web development technologies such as HTML, CSS, JavaScript, and various database systems. Skilled developers are accessible for the implementation of the platform, and the project can leverage existing frameworks and libraries to streamline development. Furthermore, the scalability and cross-device compatibility of web-based applications align well with the project's requirements, allowing for a wider reach and ease of access for users across different devices and platforms. Additionally, the availability of secure authentication protocols and encryption technologies ensures that the platform can meet stringent security requirements, safeguarding user data and privacy. Overall, the technical foundation for the E-Notes project is robust, offering a solid basis for its successful implementation.

2.2.1 Infrastructure Requirements:

- **Server Infrastructure:** Asses the capacity and scalability of cloud-based servers (e.g., AWS, Azure) to accommodate potential user growth and ensure seamless performance.
- **Database Management**: Evaluate the suitability of database systems (e.g., MySQL, PostgreSQL) for efficient storage and retrieval of user data.

2.2.2 Software Development:

- **Programming Languages:** Choose appropriate backend (e.g., Jsp,Servlet) and frontend (e.g., HTML5,CSS3,BootStrap) technologies based on developer expertise and project requirements.

- **Framework Selection**: Select a web application framework (e.g., Eclipse, IntelIj) to streamline development and enhance maintainability.

2.2.3 Security Measures:

- **Authentication Protocols:** Implement secure authentication mechanisms (e.g., APM, New Relic) to protect user accounts and ensure data security.

2.2.4 User Interface (UI) Design:

- **Responsive Design:** Optimize the app's UI for various devices (smartphones, tablets, web browsers) to provide a consistent and user-friendly experience.

2.2.5 Deployment and Monitoring:

- **Docker:** Implement containerization using Docker for efficient deployment, scalability, and consistency across different environments.

2.3 Operational Feasibility

The operational feasibility of the E-Notes project is strong, as it aims to provide a user-friendly and intuitive platform that aligns with users' existing note-taking habits. By focusing on ease of use and seamless integration into users' workflows, the platform can minimize resistance to adoption and encourage widespread use. Additionally, the project's emphasis on security and privacy aligns with users' expectations for safeguarding their notes and personal information.

2.3.1 User Acceptance:

- **User Feedback Surveys:** Conduct surveys or gather feedback from potential users to gauge their acceptance of the E-Notes App. Understand user preferences and expectations.

2.3.1 Usability Testing:

- **User Interface (UI) Testing:** Evaluate the user interface for intuitiveness and ease of use. Conduct usability testing to identify any potential issues in navigation or functionality.

2.3.2 User Engagement Strategies:

- **Communication Plans:** Develop communication strategies to keep users informed about new features, updates, and any changes in the app. Foster ongoing engagement.

2.3.3 Operational Impact Analysis:

- **Operational Workflow Analysis:** Assess how the E-Notes app will fit into users' daily workflows. Identify potential impacts on existing operational processes.

2.3.4 Change Management Strategies:

- **Change Management Plans:** Develop strategies to manage organizational and user-level changes resulting from the introduction of the E-Notes app. Address any potential resistance.

2.3.5 Legal and Compliance Considerations:

- **Compliance Analysis:** Ensure that the app complies with relevant legal and regulatory requirements related to financial transactions, data protection, and user privacy.

2.4 Behavioral Feasibility

Behavioral feasibility of the E-Notes project involves assessing how users and other stakeholders are likely to respond to the introduction of the digital note-taking platform. Understanding user behavior and preferences is crucial for the successful adoption and long-term usage of the platform.

Behavioral feasibility hinges on the willingness of users to adopt the E-Notes platform as their primary tool for note-taking. Factors such as the ease of use, availability of key features, and compatibility with existing workflows will influence users' decisions to adopt the platform. Conducting user surveys and interviews can provide insights into user preferences and expectations, helping to tailor the platform to their needs.

Introducing a new digital platform for note-taking represents a change in behavior for users accustomed to traditional methods. Managing this change effectively is critical for behavioral feasibility. Providing adequate training and support, communicating the benefits of the platform, and addressing any concerns or resistance from users are essential aspects of change management.

In addition to users, other stakeholders such as administrators, IT teams, and potential investors play a role in the behavioral feasibility of the project. Engaging with these stakeholders to gather feedback, address concerns, and build support for the project is essential for its success. Clear communication and transparency about the project's objectives, benefits, and potential challenges can help align stakeholders' behavior with the project's goals.

The user experience (UX) design of the E-Notes platform plays a crucial role in its behavioral feasibility. A well-designed interface that is intuitive, visually appealing, and responsive to user actions can enhance user engagement and satisfaction. Conducting usability testing and incorporating user feedback into the design process can ensure that the platform meets users' expectations and encourages the desired behavior, such as regular usage and engagement with the platform.

2.5 Schedule Feasibility

Schedule feasibility for the E-Notes project refers to the project's ability to be completed within a reasonable timeframe, considering various factors such as development timelines, resource availability, and potential risks. Assessing schedule feasibility involves creating a realistic timeline for the project's stages and ensuring that it aligns with the organization's objectives.

The development timeline for the E-Notes project should be carefully planned to account for various stages, including design, development, testing, and deployment. Each stage requires time for planning, execution, and review, and the timeline should consider the complexity of the project's features and functionalities. Additionally, milestones and checkpoints should be established to track progress and make necessary adjustments to the schedule.

Schedule feasibility depends on the availability of resources, including human resources (developers, designers, testers) and technological resources (software, hardware, tools). Adequate staffing and access to necessary technologies are essential for adhering to the project timeline. Any potential constraints or limitations in resource availability should be identified early in the planning phase to mitigate risks to the schedule.

Identifying and addressing potential risks that could impact the project schedule is crucial for schedule feasibility. Risks such as technical challenges, scope creep, resource constraints, or external dependencies should be carefully evaluated, and contingency plans should be developed to minimize their impact on the project timeline. Regular risk assessments and proactive risk management strategies can help maintain schedule feasibility throughout the project lifecycle.

CHAPTER 3

DATABASE DESIGN

3. INTRODUCTION

The database design for the E-Notes app is a foundational element that a pivotal role in organizing and managing data efficiently. A well-structured database is essential for facilitating seamless interactions between the application and its uses, ensuring robust data storage, retrieval and manipulation. The introduction outlines of the fundamental principles guiding the database design process and emphasizes the importance of creating a scalable and secure data architecture to support the diverse needs of notes management within collaborative settings.

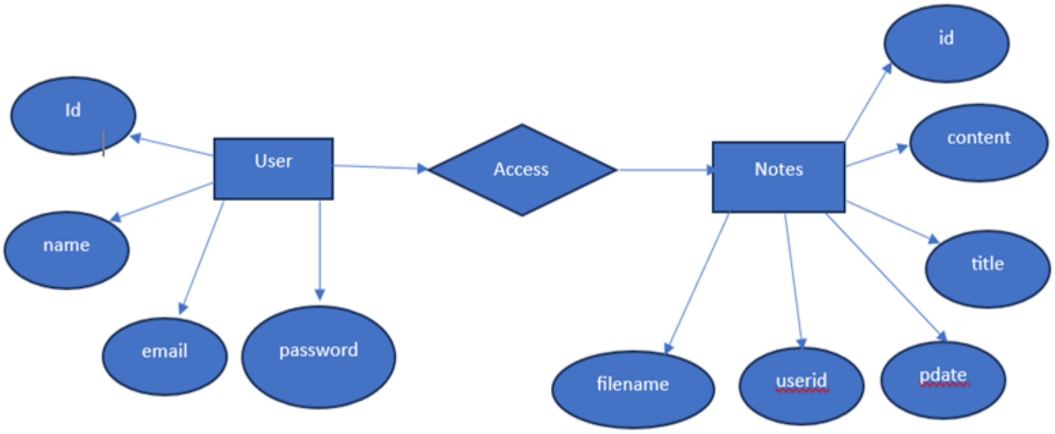


Fig. 3.1 Database Design

3.1 Database Tables

Creating a comprehensive database table for the Share Expense app involves considering the key entities and their attributes. In a simplified example, let's focus on two main entities: Users and Expenses. Here's a basic representation:

3.1.1 Users Table:

- **user_id (Primary Key):** Unique identifier for each user.
- **email:** User's email address for communicate and login.
- **name:** User's full name.
- **password:** Securely hashed password for authentication.

user_id	Email	Name	Password
1	harshkumar2097@gmail.com	Harsh	#2122223fsdx
2	a@gmail.com	Aman	#3c2223rsdx
3	Harshit1234@gmail.com	Harshit	#2122253ftdx

Table 3.1. User's Table

3.1.2 Notes Table:

- **P_Date:** Date when the notes was created.
- **Id:** Unique id given to every notes.
- **Content:** Stores the content written inside the notes.
- **Title:** Tilte for each Notes.
- **User_id:** foreign key
- **File Name:** Stores the name of the file.

P_Date	Id	Content	Title	User_id	File Name
2024-01-09	1	4900	Java	1	
2023-01-09	2	3900	Python	2	
2024-02-09	3	3800	DBMS	3	

Table 3.2. Notes Table

3.2 Flowchart

Introduction to the Flowchart for E-Notes App:

The flowchart begins with the user's decision to register for the E-Notes platform. Upon visiting the platform's website or app, the user is presented with the option to register. The flow then branches into two paths: one for new users to register by providing their details such as username, email, and password, and another for existing users to log in using their credentials. Once logged in, the user gains access to the platform's features.

After logging in, the user can proceed to create a new note. The flowchart illustrates the steps involved in creating a note, including entering the note title, adding content, and saving the note. Additionally, the flowchart depicts how users can manage their notes, such as editing existing notes, deleting notes they no longer need, and organizing notes into categories or folders for easier management.

The flowchart also includes a path for users to share their notes with others. This might involve selecting a note to share, choosing the sharing method (e.g., email, link sharing), and specifying the recipients. The flowchart should outline the various options available for sharing notes and the steps users need to follow to complete the sharing process successfully.

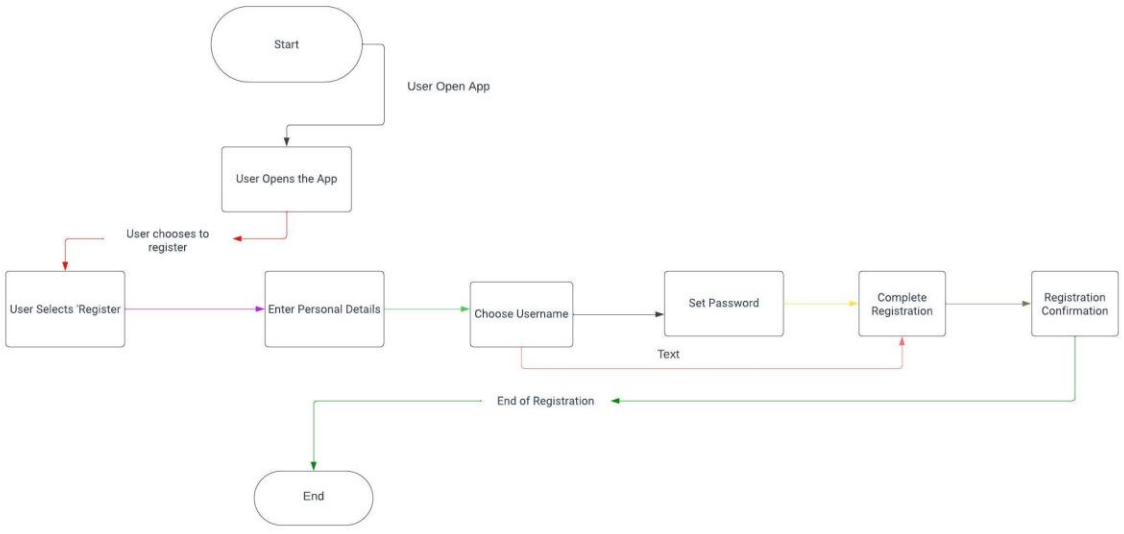


Fig. 3.2 Flowchart Diagram for E-Notes App

3.3 Use Case Diagram

The use case diagram for the e-notes project illustrates the various interactions and functionalities available to users. The primary actors are the user and the system. The user can register by providing necessary details such as username, email, and password. After successful registration, the user can log in using their credentials. Upon logging in, the user can perform several actions related to notes management. They can create new notes, delete existing notes, search for specific notes based on keywords or tags, and update the content of their notes. All these operations are performed through the user interface, which interacts with the database to store and retrieve note data. Finally, the user can log out to end their session securely.

The system, as a secondary actor, manages the authentication process, note management functionalities, and database operations. It validates user credentials during login, ensures the security and integrity of user data in the database, and provides a seamless interface for users to interact with their notes. The database stores user information, such as login credentials and notes, ensuring that the data is persistently available for future use. Overall, the use case diagram outlines the key features and interactions of the e-notes system, focusing on user registration, login, note management, and database integration.

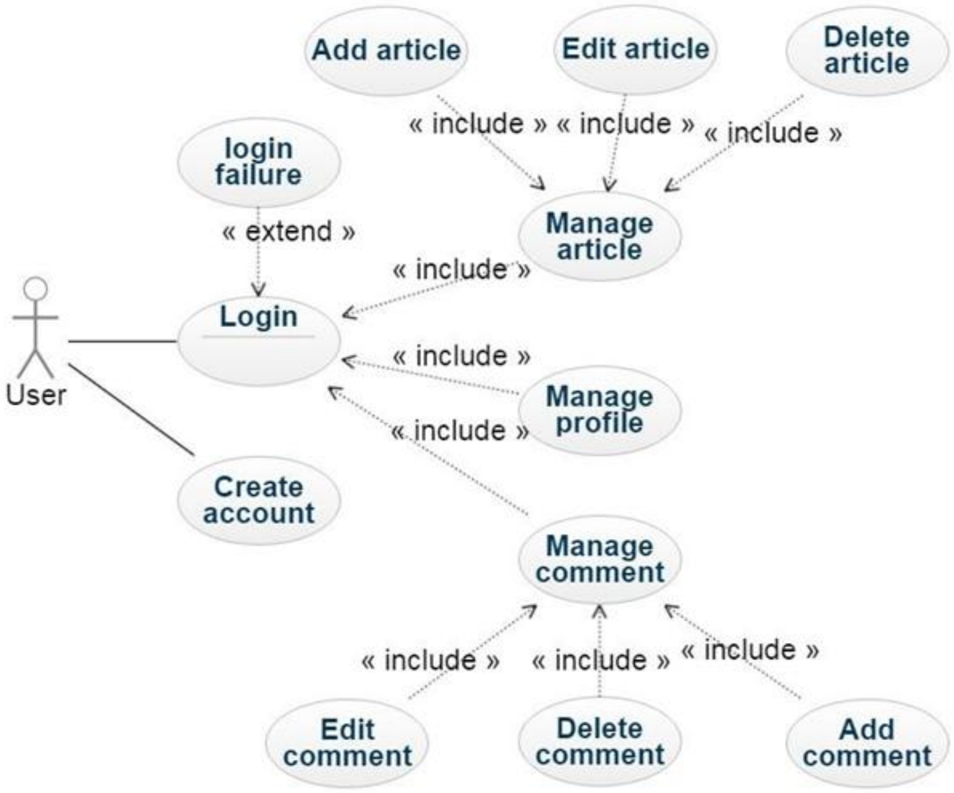


Fig. 3.3 Use Case Diagram for E-Notes App

3.3.1 Actors:

- User
- System

3.3.2 Use Case

- Register
- Login
- Sign Up
- Open Dashboard
- Add Notes
- View Notes

In the E-Notes app, when the user opens the web page, they encounter two modes: registration or login. If he/she is a new user than they have to register themselves otherwise they just have to login with the valid credentials such as email id and password which they have created at the time of registration. When the user fill their credentials then those credentials are checked in the database if the credentials are true or not. If the credentials are correct than the user encounters with the dashboard otherwise an error message is displayed on the screen. When the user encounters with the dashboard than he has multiple choices for example:- they can create their notes, they can access their previously written notes, they can update their notes or they can delete their notes as per the demand of the user.A user can also search their notes in the search box provided above and can access their notes. In this project user cannot copy paste the data from some other source because of the validations applied in this project i.e. user can only write their own notes as well as they can also add images with the notes if they wanted to do so. After the work of the user is completed than they can logout from the application and their content is directly saved in the database and anytime in future if they want to access their notes they just have to login in the app and they can get access to their notes.

3.4 Data Flow Diagram

The Data Flow Diagram (DFD) for the Share Expense app provides a visual representation of the flow of data within the system, illustrating how information moves between various components. At its core, the DFD encapsulates the key processes, data stores, and data flows involved in the expense management application. Starting with user inputs, such as registering, logging in, and entering expense details, the diagram delineates how these interactions trigger processes like data validation, expense calculations, and database updates. The DFD also portrays the storage and retrieval of user data in the database, emphasizing the seamless exchange of information between users and the application.

By encapsulating the fundamental data movements and transformations, the DFD serves as a valuable blueprint for understanding the E-Notes app operational dynamics.

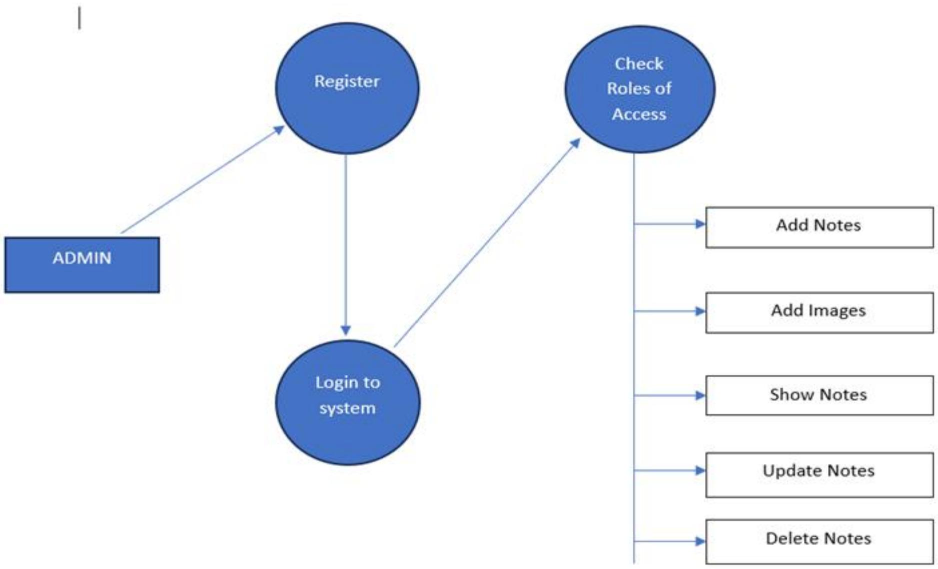


Fig. 3.4 Data Flow Diagram for E-Notes App

CHAPTER 4

FORM DESIGN

4. INTRODUCTION

In the e-notes project, the form design plays a crucial role in providing a user-friendly interface for users to interact with the system. The forms are designed to seamlessly guide users through the registration and login processes, ensuring that they can securely access their accounts and manage their notes. The forms for creating, updating, and deleting notes are intuitively designed, with fields for entering note content, selecting tags, and performing actions such as saving or deleting notes. The search form allows users to enter keywords or tags to find specific notes quickly. Overall, the form design focuses on clarity, simplicity, and functionality, aiming to enhance the user experience and streamline the note management process.

4.1 Input/Output Form (Screenshot)

4.1.1 Main Page

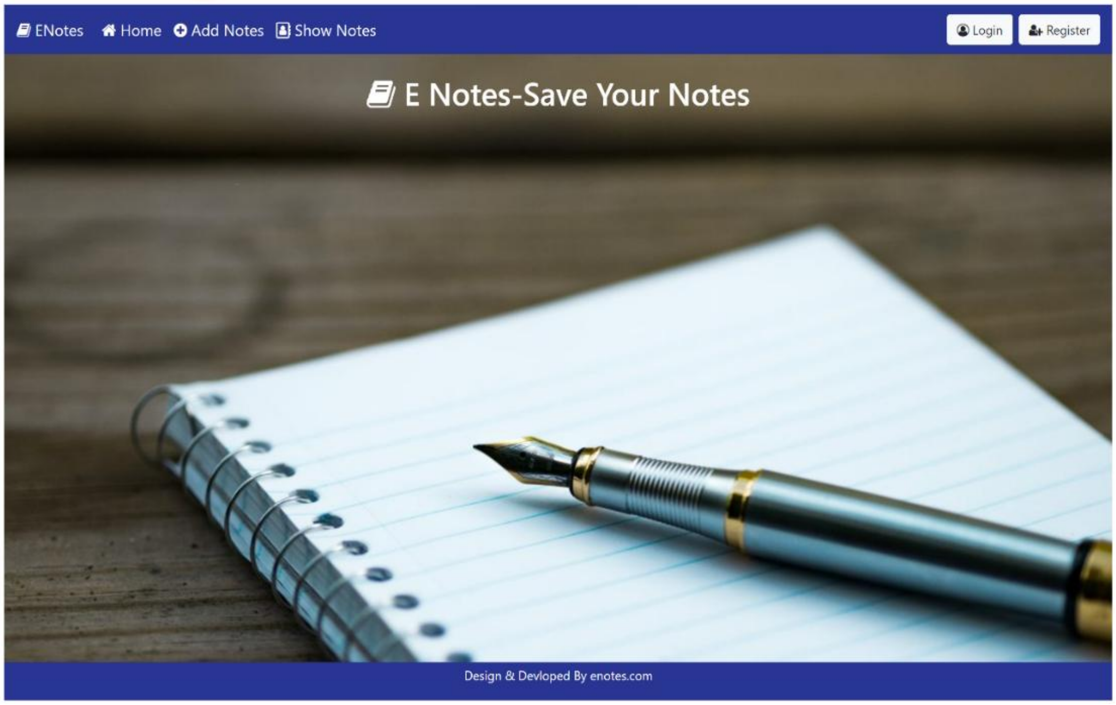


Fig. 4.1 Initially user select Register/Login

4.1.2 Registration Form

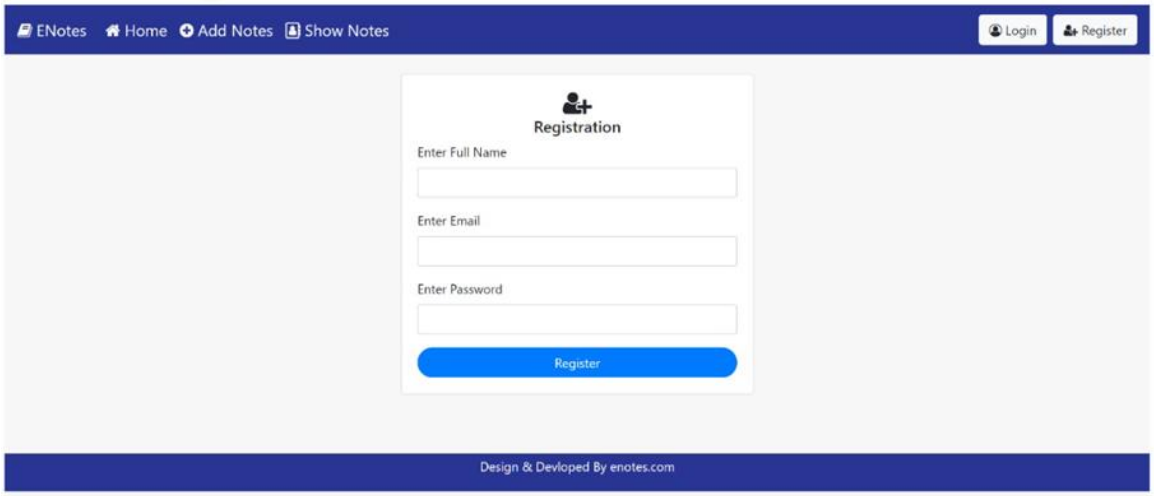


Fig. 4.2 User Registration

4.1.3 **Login Module:** User can login with valid Credentials

- ✓ If admin entered incorrect credentials, then alert will generated but if credentials are matched with the admin credentials then admi can login.

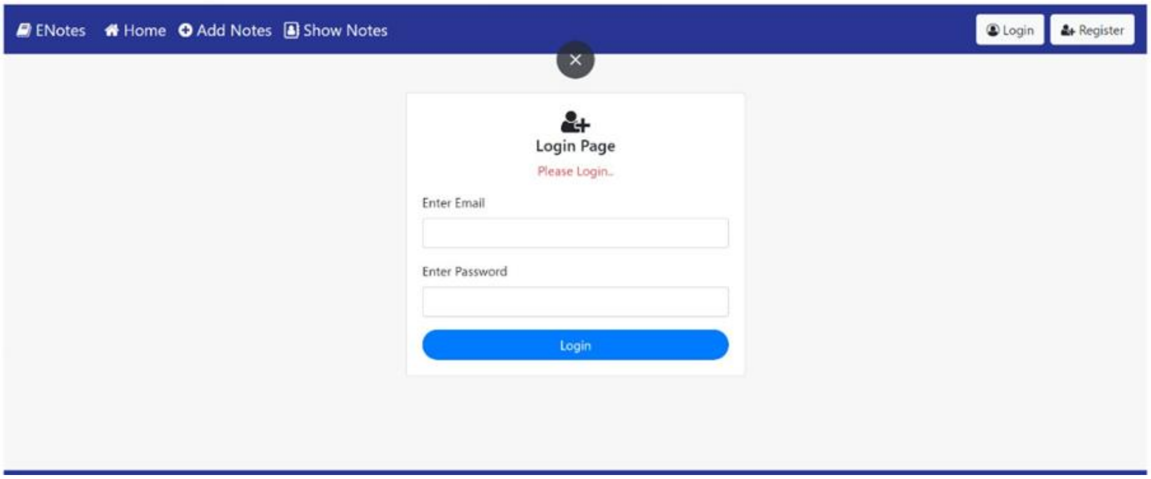


Fig. 4.3 User Enter Correct Credentials

4.1.4 User Dashboard:

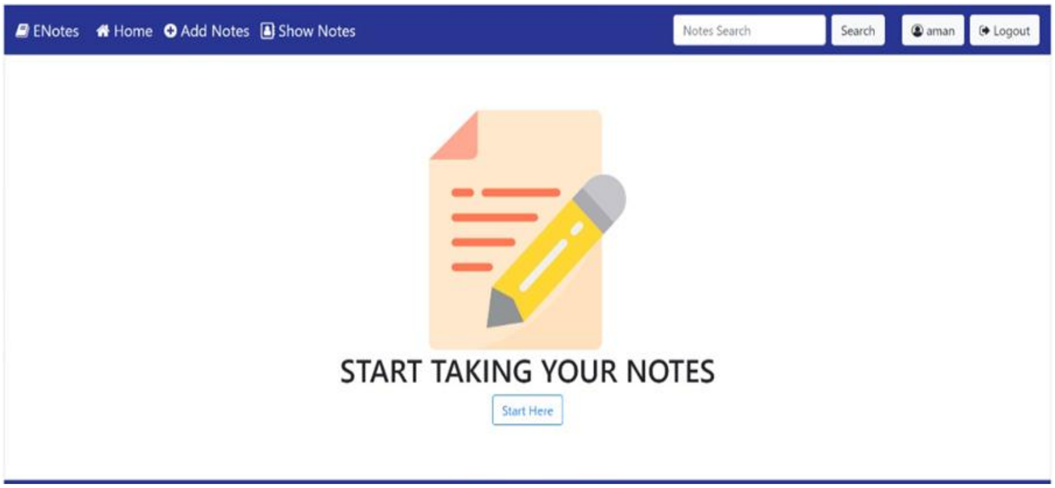


Fig. 4.4 User Dashboard

4.1.5 Create/Add Notes Form:

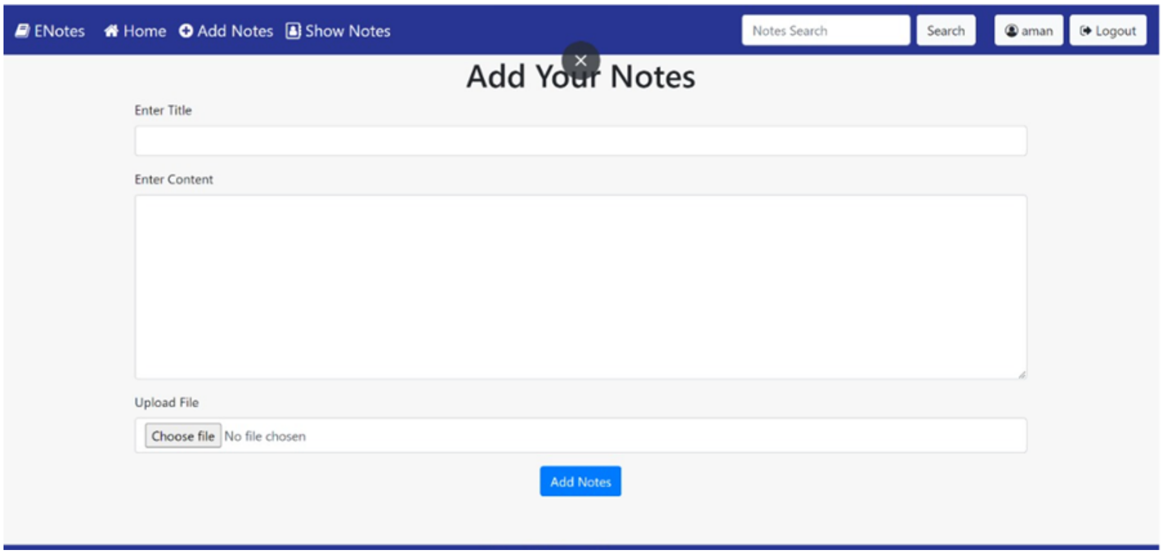
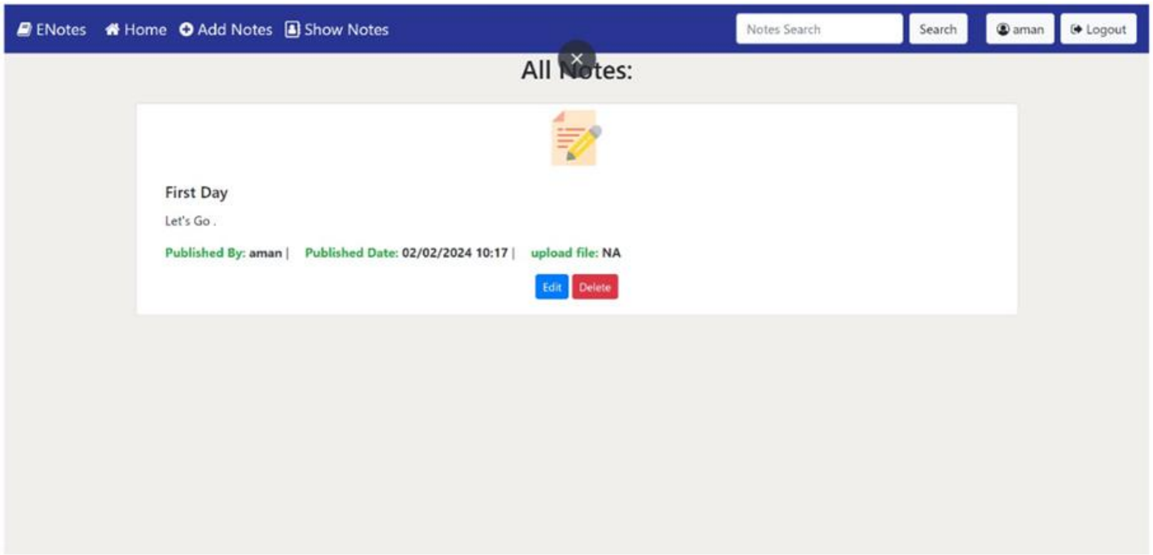


Fig. 4.5 Create/Add Notes

4.1.6 Save Notes Form:



First Day

Let's Go .

Published By: aman | Published Date: 02/02/2024 10:17 | upload file: NA

Fig. 4.6 Save Notes Module

CHAPTER 5

TESTING

INTRODUCTION

Testing is a critical phase in the development lifecycle of the E-Notes app, serving as a systematic and thorough examination of its functionality, performance, and reliability. This essential process involves evaluating the application's features, identifying potential defects, and ensuring that it meets specified requirements. Various testing methodologies, including unit testing, integration testing, and user acceptance testing, are employed to assess different facets of the application. Testing not only validates that each component operates as intended but also verifies the seamless interaction between these components. By rigorously testing the E-Notes app, developers aim to deliver a high-quality product that aligns with user expectations, minimizes the likelihood of errors, and provides a robust and reliable platform for effective shared financial management.

5.1. Test Case-1

5.1.1 Test Case 1: User Registration

Objective: To ensure that users can successfully register for the E-Notes app.

5.1.1.1 Preconditions:

- The E-Notes app is accessible and running.
- The user is on the app's registration page.

5.1.1.2 Test Steps:

- Enter valid information into the registration form, including a unique email address, a full name and a secure password.
- Click on the "Submit" button.

5.1.1.3 Expected Results:

- The user should be successfully registered, and a confirmation message should be displayed.
- The user's information, including their email should be stored in the database.

5.1.1.4 Postconditions:

- The user should be able to log in using the registered credentials.

5.2. Test Case-2

5.2.1 Test Case 1: Add Notes

Objective: To verify that users can add a note successfully.

5.2.1.1 Preconditions:

- The user is logged into the E-Notes app.

5.2.1.2 Test Steps:

- Navigate to the add notes form.
- Fill in the necessary title, including content.
- Click on the "Save" button.

5.2.1.3 Expected Results:

- The notes should be successfully saved, and a confirmation message should be displayed.
- The saved notes details, including the title, content and user information, should be stored in the database.

5.2.1.4 Postconditions:

- The saved notes should be visible in the user's history.

BIBLIOGRAPHY

Books:

1. Sommerville, I. (2016). Software Engineering. Pearson Education Limited.
    - This book provides insights into software engineering principles and practices, which are essential for developing a BAMS.
2. "Head First Servlets and JSP" by Bryan Basham, Kathy Sierra, and Bert Bates
    - This book provides a thorough introduction to Servlets and JSP with a focus on practical examples and hands-on learning.

Academic Journals:

1."IEEE Internet Computing" - This journal covers a wide range of topics related to internet computing, including web development using Java technologies.

2."International Journal of Web Engineering and Technology" - This journal focuses on web engineering and technology, making it a potential source for research on JSP, Servlets, and related topics.

Online Resources:

1. Oracle Java EE Documentation: The official documentation for Java EE provides detailed information on Servlets, JSP, and related technologies. You can find tutorials, guides, and API references on the Oracle website.
2. W3Schools Java Servlets Tutorial: W3Schools offers a beginner-friendly tutorial on Java Servlets, covering topics like servlet lifecycle, request handling, and session management.