

CAPSTONE PROJECT REPORT

ON

ENOTE WEB APPLICATION

Submitted in Partial Fulfillment of the Requirement
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

BY

ANJANI KUMAR SHUKLA

(ID. No :20BTCSE013)

UNDER THE SUPERVISION OF

Er. SANJAY T. SINGH



**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
VAUGH INSTITUTE OF AGRICULTURAL ENGINEERING
AND TECHNOLOGY
SAM HIGGINBOTTOM UNIVERSITY OF AGRICULTURE,
TECHNOLOGY AND SCIENCES
NAINI, PRAYAGRAJ-211007
2024**

DECLARATION

I, Anjani Kumar Shukla, declare that the work presented in this project report entitled ‘**ENOTE WEB APPLICATION**’ submitted to the Department of Computer Science and Information Technology in Faculty of Engineering and Technology, **Sam Higginbottom University of Agriculture, Technology and Sciences, Naini, Prayagraj**, for the award of the degree of B.Tech in Bachelor of Technology Computer Science & Engineering is an original work. I have neither plagiarized nor submitted the same work for the award of any other degree. In case this undertaking is found incorrect, my degree may be withdrawn unconditionally by the University.

Date:

Anjani Kumar Shukla

Place: Prayagraj

ID: 20BTCSE013

CERTIFICATE

Certified that the project report entitled '**ENOTE WEB APPLICATION**' submitted to **Sam Higginbottom University of Agriculture, Technology and Sciences, Naini, Prayagraj** in partial fulfillment of the requirement for the award of the degree of **Bachelor Of Technology** in Computer Science & Engineering **Faculty of Engineering and Technology** is a bonafide record of project work carried out by **Mr. Anjani Kumar Shukla (ID No:20BTCSE013)** under my supervision and guidance.

Date:

Place:Prayagraj

Er. Sanjay T. Singh

Project Advisor

CERTIFICATE OF ACCEPTANCE OF EVALUATION COMMITTEE

This is to certify that project entitled '**ENote Web Application**' has been prepared and submitted by **Anjani Kumar Shukla** in partial fulfillment of the requirement for the award of the degree of "Bachelor of Technology in Computer Science & Engineering" in Department of Computer Science and Information Technology, Vaugh Institute of Agricultural Engineering and Technology, Sam Higginbottom University of Agriculture, Technology and Sciences, Prayagraj (U.P.) INDIA.

Name	Evaluation	Signature
Er. Sanjay T. Singh Project Advisor Department of Computer Science & IT SHUATS, Prayagraj	Satisfactory / Unsatisfactory	
Dr. Tulika Project Coordinator Assistant Professor Department of Computer Science & IT SHUATS, Prayagraj	Satisfactory / Unsatisfactory	
Er. R. Dileep Kumar Project Coordinator Assistant Professor Department of Computer Science & IT SHUATS, Prayagraj	Satisfactory / Unsatisfactory	

The evaluation committee has thoroughly examined this project and has found it acceptable.

Dr. W. Jeberson

Chairman

HoD

Department of Computer Science and IT

SHUATS, Prayagraj

ABSTRACT

The “**ENOTES WEB APPLICATION**” is designed to facilitate efficient note-taking and organization for users in an online environment. The platform aims to provide a seamless and intuitive experience for creating, editing, and managing digital notes. Key features include real-time synchronization across devices, collaboration options, and advanced search functionalities.

ACKNOWLEDGEMENT

In the completion of this project, I extend my heartfelt appreciation to those individuals and entities who have played pivotal roles in bringing this project to completion. My sincere thanks go to **Er. Sanjay T. Singh**, whose guidance and insightful contributions have been invaluable throughout the developmental stages. I also express gratitude to Project Coordinators for their collaborative efforts and constructive feedback, enhancing the overall quality of this work. The support provided by **Er. Rishabh Chaudhary , Dr. Tulika, Er. Dileep Kumar** has been instrumental, and I acknowledge their contribution to the successful execution of this project. This project stands as a testament to the collective efforts and support received, and I am profoundly grateful to all who have contributed to the efforts that have shaped its completion.

TABLE OF CONTENTS

Title	Page No.
DECLARATION	
CERTIFICATE	
CERTIFICATE OF ACCEPTANCE OF EVALUATION COMMITTEE	
ABSTRACT	i
ACKNOWLEDGEMENT	ii
TABLE OF CONTENTS	iii
LIST OF TABLES	iv
LIST OF FIGURES	v
CHAPTER I INTRODUCTION	10-14
1.1 ENote Web Application	
1.2 Objectives	
1.3 Project Scope	
1.4 Users of the system	
1.5 Life cycle model used	
1.6 Implementation and Maintenance	
CHAPTER II REQUIREMENT SPECIFICATION	15-19
2.1 Functional Requirements	
2.2 Non-Functional Requirements	
2.3 Hardware requirement	
2.4 Software requirement	
CHAPTER III SYSTEM DESIGN	20-23
3.1 System Architecture	
3.2 Modules of the system	
3.3 Use Case Diagram	
3.4 Database Design	
3.5 Class Diagram	

CHAPTER IV IMPLEMENTATION AND TESTING	24-29
4.1 Frontend	
4.2 Backend	
4.3 User Experience (UX) Refinement	
4.4 Testing	
CHAPTER V CONCLUSION AND FUTURE SCOPE	30-32
5.1 User-Centric Experience	
5.2 Community Building	
5.3 Security and Reliability	
5.4 Scalability	
5.5 Future Scope	
REFERENCES	33
APPENDICES	34
Appendix A Sample Code	35-40
Appendix B Screenshots	41-44

LIST OF FIGURES

Figure No	Title	Page No.
Figure 1.5	Prototype model	15
Figure 3.3	Use case Diagram of ENOTE WEB APPLICATION	23
Figure 3.4	Class Diagram of ENOTE WEB APPLICATION	24

CHAPTER 1

INTRODUCTION

Introduction

Enotes is a web application for creating and managing personal notes. It is a powerful web application designed to revolutionize the way you engage with and manage electronic notes. Whether you're a professional student, , or anyone who values efficient note-taking, eNotes offers a seamless and intuitive platform to organize, share, and access your notes from anywhere .

1.1.1 Need for enotes web application

1. Digital Transformation:

With the increasing digitization of workflows and processes, there is a growing need for digital solutions in various aspects of life, including note-taking. An eNotes web application can facilitate this transition by providing a platform for users to create, organize, and access notes in a digital format.

2. Collaboration and Sharing:

In both academic and professional settings, collaboration is crucial. An eNotes application could fulfill the need for a centralized platform where individuals or teams can collaborate on note creation and sharing, enhancing efficiency and teamwork.

3. Accessibility Across Devices:

Modern users often switch between multiple devices throughout the day, such as laptops, tablets, and smartphones. An eNotes web application can meet the need for seamless accessibility across these devices, ensuring users can access their notes anytime, anywhere.

4. Organization and Searchability:

As the volume of information individuals deal with continues to grow, the need for effective organization and searchability becomes essential. An eNotes application can offer features like tags, categories, and search functionalities to help users quickly locate specific information within their notes.

5. Security and Privacy:

With the increasing concern about data security and privacy, an eNotes web application can address these needs by implementing robust security measures. This ensures that users can trust the platform with their sensitive information, fostering confidence in the digital note-taking process.

6. Integration with Other Tools:

Many users rely on a variety of digital tools and platforms for different tasks. An eNotes web application can serve as a central hub by integrating with other productivity tools, calendars, or communication platforms, providing a seamless user experience.

7. Customization and Rich Features:

Users often seek note-taking applications that offer a range of features to enhance their experience. An eNotes web application can meet this need by providing customization options, rich text formatting, multimedia support, and other features that cater to diverse user preferences.

8. Education Sector Requirements:

In educational settings, students and educators often require efficient note-taking tools. An eNotes application tailored for the education sector can include features like annotation, collaborative study groups, and integration with educational resources, addressing specific needs in this domain.

1.2 Objectives

The primary objective of the eNotes web application is to provide users with a digital platform for efficient and organized note-taking. This addresses the need for users to transition from traditional paper-based note-taking to a digital format.

1.3 Project Scope

The scope of the project is to develop a website which shows the latest posts and announcements. One can add, edit and delete new posts and people can like and comment on that post. The users of the project can register, edit his/her profile pic etc. Categories can also

be added. The system is used only by university students and staff. It is not integrated with any external system. Bloggers can not earn money with their posts.

1.4 Users of the system

The users of the system are students and staff of the University. They can add categories, add posts, like posts, delete posts and comment on a particular post. They can ask questions and anybody can answer.

1.5 Life cycle model used

The lifecycle model used to develop this project is a prototype model. The prototype model is a software development model where a prototype (a working model of the system) is created, tested, and refined before the final product is developed. Prototypes allow for early user involvement and feedback. This is crucial for a enotes web application where user experience and engagement are key factors. Gathering feedback from potential users during the prototype stage helps in understanding their needs and preferences. The prototype model supports an iterative and incremental development approach. This is beneficial for a enotes web application project as it allows developers to refine and enhance features based on ongoing feedback. Any issues related to usability, navigation, or functionality can be identified and addressed early in the development cycle.

The Prototyping Model is one of the most popularly used Software Development Life Cycle Models (SDLC models). This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested, and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product. Figure 1.1 shows the phases of the prototype model. The phases are

- 1. Requirement Gathering and Analysis:** This is the initial step in designing a prototype model. In this phase, users are asked about what they expect or what they want from the system.
- 2. Quick Design:** This is the second step in Prototyping Model. This model covers the basic design of the requirement through which a quick overview can be easily described.

3. **Build a Prototype:** This step helps in building an actual prototype from the knowledge gained from prototype design.
4. **Initial User Evaluation:** This step describes the preliminary testing where the investigation of the performance model occurs, as the customer will tell the strength and weaknesses of the design, which was sent to the developer.
5. **Refining Prototype:** If any feedback is given by the user, then improving the client's response to feedback and suggestions, the final system is approved.
6. **Implement Product and Maintain:** This is the final step in the phase of the Prototyping Model where the final system is tested and distributed to production, here the program is run regularly to prevent failures.

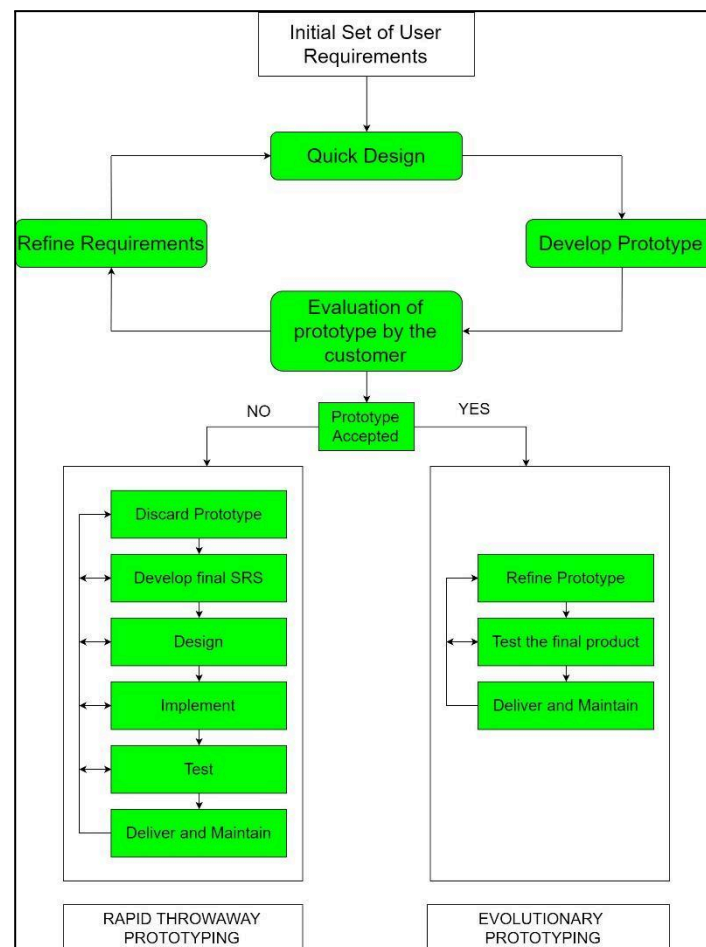


Figure 1.5 Prototype model

CHAPTER 2

REQUIREMENT

SPECIFICATION

REQUIREMENT SPECIFICATION

A Requirement Specification provides a comprehensive outline of the functional and nonfunctional requirements. This document serves as a blueprint for the project, guiding developers, designers, and stakeholders in understanding the scope, features, and characteristics of the web app. This chapter presents all the requirements for developing the blogging website “**eNOTE WEB APPLICATION**”.

2.1 Functional Requirements

The functional requirements of the system are listed below.

1. User Management:

User Registration:

- Users can create accounts with a valid email address and password.
- Include email verification during the registration process.

2. User Interface:

- Design an intuitive and responsive user interface for seamless navigation and optimal user experience.

3. Authentication:

- Users must be able to log in securely using their credentials.

4. Note Creation and Editing:

- Provide functionality for users to create, edit, and format digital notes with various text formatting options.

5. Note Organization:

- Implement tools for users to categorize, tag, and organize their notes efficiently, supporting a structured and easily accessible system.

6. Search and Filtering:

- Users can search for specific content or topics.
- Include filtering options for content categories, tags, or popularity.

2.2 Non-Functional Requirements:

The non-functional requirements of the system are listed below.

1. **Response Time:** The website should load pages and content within 2 seconds.
2. **Scalability:** The system must be scalable to handle an increasing number of users and content without significant degradation in performance.
3. **Availability:** The system must aim for 99.9% uptime.
4. **Fault Tolerance:** The system should gracefully handle errors and minimize service disruptions.
5. **Authorization:** The system must implement access controls to ensure users can only access appropriate information.
6. **UI:** The system should have an intuitive user interface with clear navigation.
7. **Compatibility:** The system should be compatible with major web browsers.
8. **Modularity:** Codebase should be modular for easy updates and maintenance.

2.3 Hardware requirement

The hardware requirements for a eNote Web Application web project developed in Java can vary based on factors such as the scale of the project, expected traffic, and the complexity of the application. However, I can provide you with a general guideline for the hardware requirements. Keep in mind that these are just recommendations, and you may need to adjust them based on your specific project needs.

1. Web Server:

- A robust web server is essential to handle HTTP requests and serve the web application. Common choices include Apache Tomcat.
- Recommended: At least 4 GB of RAM, multi-core processor.

2. Database Server:

- Java web applications often use relational databases like MySQL, PostgreSQL, or Oracle.
- Recommended: Sufficient disk space for the database, at least 4 GB of RAM, multi-core processor.

3. Network Infrastructure:

- Ensure a reliable network infrastructure with sufficient bandwidth to handle incoming and outgoing traffic.
- Recommended: High-speed internet connection, reliable networking equipment.

4. Backup System:

- Implement a robust backup system for both the database and the application code to prevent data loss.
- Recommended: Regular automated backups stored in a secure location.

2.4 Software requirement

The software requirements for a eNote Web Application web project developed in Java include various components and tools that are necessary for the development, deployment, and maintenance of the application. Here are the essential software requirements:

1. Java Development Kit (JDK):

- Java is the primary programming language for your web project. Install the latest version of the Java Development Kit (JDK) on your development machine.
- Recommended: JDK 8 or later, depending on your project's compatibility and requirements.

2. Integrated Development Environment (IDE):

- Choose a Java IDE to streamline the development process. Popular choices include Eclipse, IntelliJ IDEA, and NetBeans.
- Recommended: IntelliJ IDEA is widely used in the Java development community for its features and ease of use.

3. Web Framework:

- Select a Java web framework to simplify web application development. Popular choices include Spring MVC, JavaServer Pages (JSP).
- Recommended: Spring Boot is a popular choice for its simplicity and extensive features.

4. Database Management System (DBMS):

- Choose a relational database management system (RDBMS) to store and manage data. MySQL, PostgreSQL, and Oracle Database are common choices.
- Recommended: The choice of the DBMS may depend on your organization's preferences and requirements.

5. Version Control System:

- Implement version control to manage source code changes efficiently. Git is the most widely used version control system.
- Recommended: Use Git for version control, and platforms like GitHub or GitLab for collaboration and code hosting.

6. Frontend Technologies:

- Depending on your project requirements, incorporate frontend technologies such as HTML, CSS, Bootstrap and Font Awesome. Consider using a frontend framework like React, Angular, or Vue.js if needed.

7. Security Libraries:

- Integrate security libraries to protect your web application from common vulnerabilities. Spring Security is a widely used security framework for Java applications.

8. Testing Frameworks:

- Use testing frameworks to ensure the quality of your code. JUnit is a popular choice for unit testing in Java, and tools like Selenium can be used for automated testing of web interfaces.

CHAPTER 3

SYSTEM DESIGN

Enote Web Application Architecture

Frontend Interface:

- HTML for structure.
- CSS for styling.
- Bootstrap for responsive design and UI components.
- Font Awesome for icons.
- Includes a text editor component for user input.
- Displays analysis results using HTML/CSS.
- Handles user authentication and authorization forms using HTML forms.

Backend Server:

- Implemented using Java Servlets for handling HTTP requests and responses.
- Utilizes JSP (JavaServer Pages) for dynamic content generation.
- JDBC (Java Database Connectivity) for interacting with the MySQL database.
- Implements business logic and integrates with external services.
- Provides RESTful endpoints for frontend-backend communication.

Text Analysis Service Integration:

- Utilizes Java libraries for making HTTP requests to the Articut API or similar service.
- Sends text data to the analysis service for processing.
- Receives and processes the analyzed results.
- Implements error handling and retry mechanisms for reliable integration.

Database:

- MySQL database for storing user data, including notes, analysis results, and user profiles.
- JDBC for connecting Java Servlets to the MySQL database.
- Stores data in tables designed to support the application's functionality.

Authentication and Authorization:

- Manages user accounts, authentication, and authorization using Java Servlets.
- Utilizes session management for maintaining user sessions.
- Implements secure password storage and authentication mechanisms.

Additional Features:

- Collaboration tools: Allows users to collaborate on notes or share analysis results with others.
- Search functionality: Enables users to search through their notes or analysis results.
- Export/import functionality: Supports exporting notes in various formats (e.g., PDF, CSV) and importing text documents for analysis.

- Notifications: Sends notifications to users for various events, such as completed analyses or shared notes.

3.3 Use Case Diagram

For a ENote Web Application project developed in Java, a use case diagram can illustrate the various interactions between external entities (actors) and the system. Here's a potential actors and use cases:

User: Manages user accounts, oversees content moderation, and has access to administrative features.

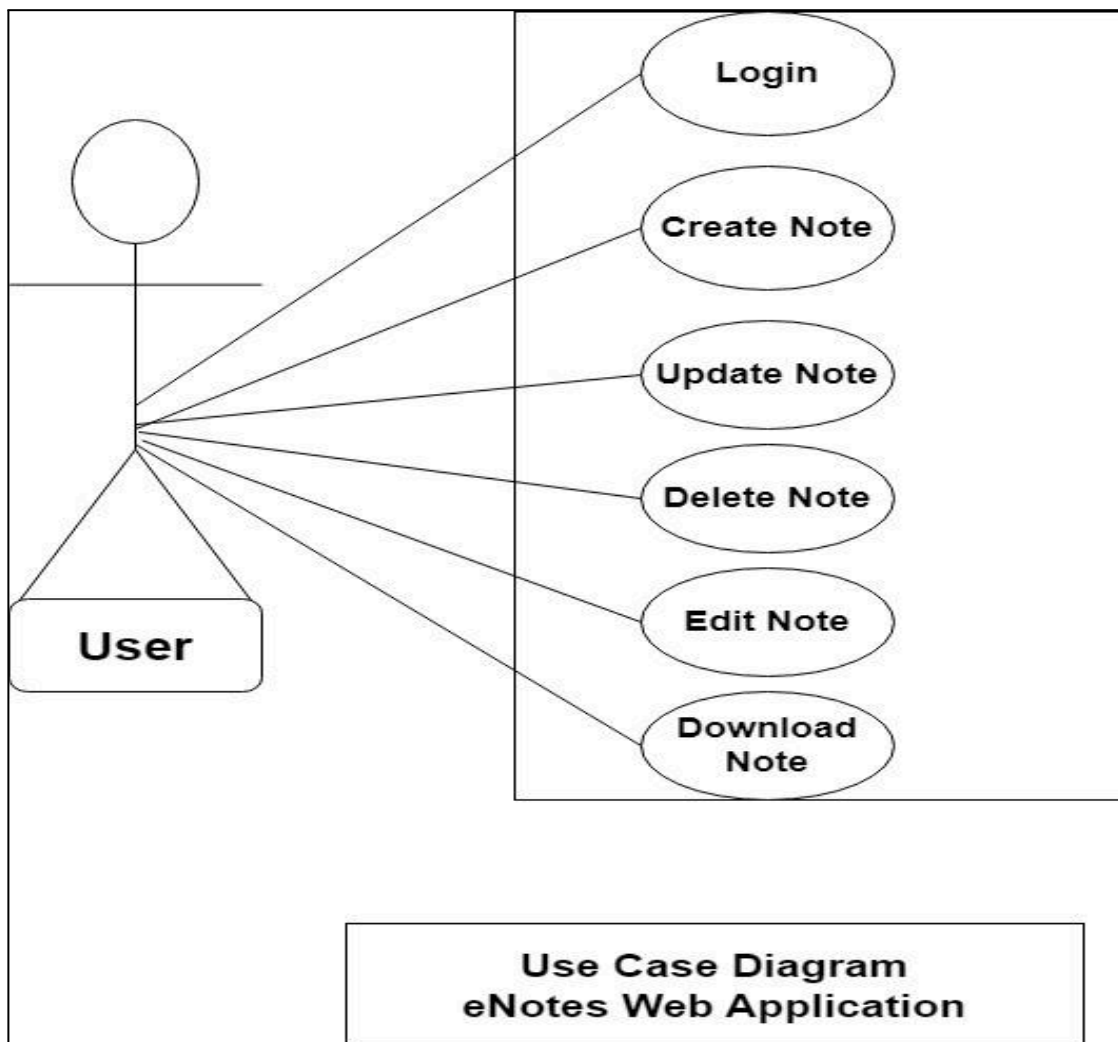


Figure 3.3 Use case diagram of ENOTE WEB APPLICATION

3.4 Class Diagram

A class diagram for the "ENote Web Application" web project visually represents the system's structure by illustrating the classes (objects), their attributes, and the relationships between them. Here's a key classes and their associations:

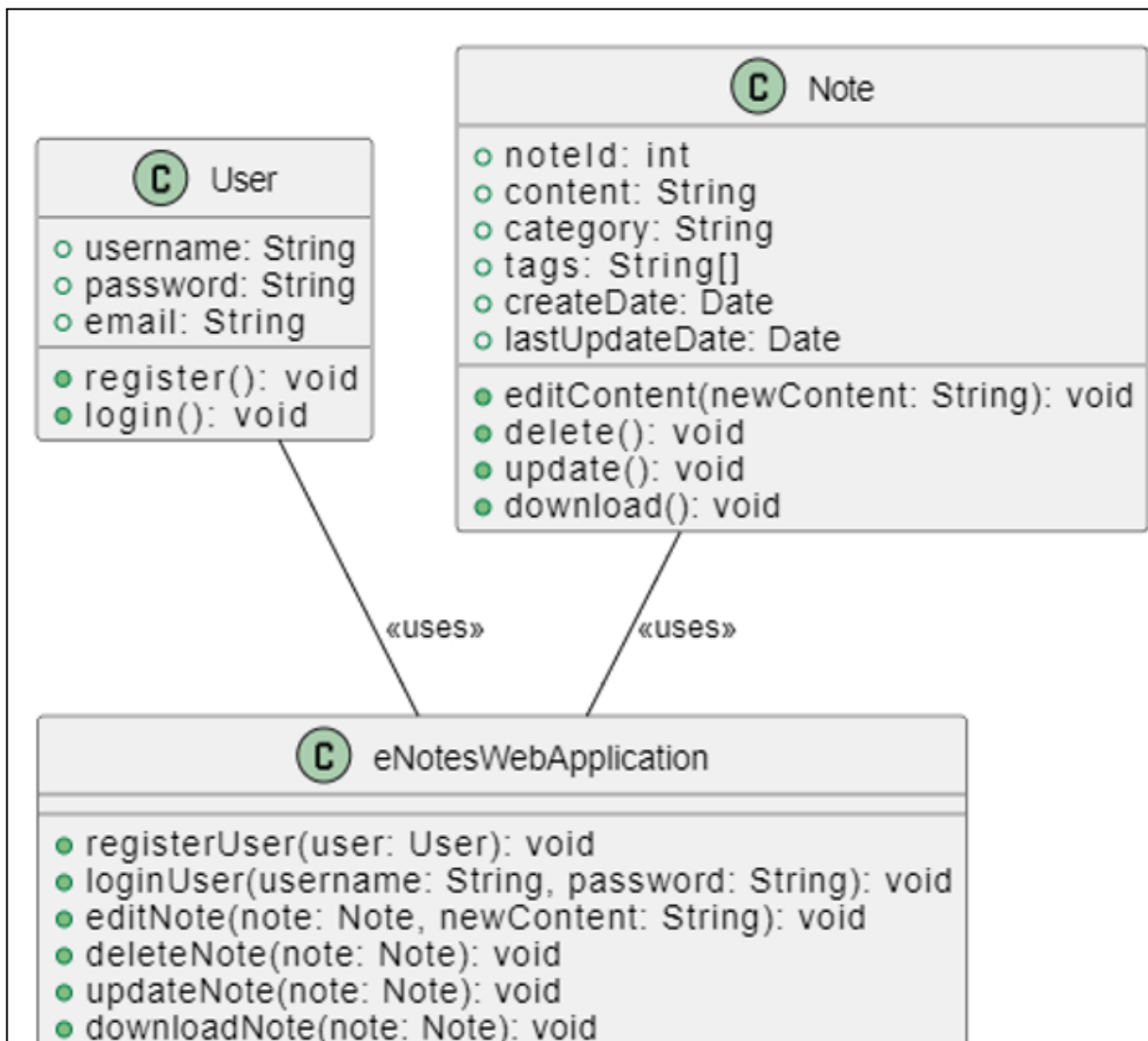


Figure 3.4 Class Diagram of ENOTE WEB APPLICATION

CHAPTER 4

IMPLEMENTATION

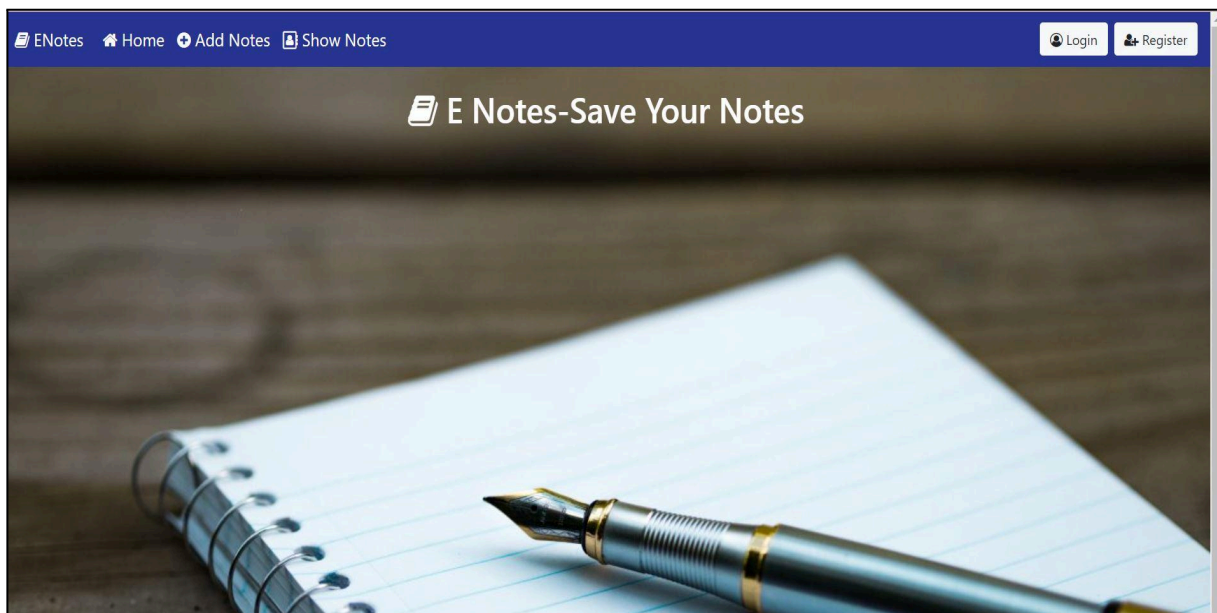
AND TESTING

Implementation

The implementation phase involves turning the system design into a functioning "ENote Web Application" web application. This phase encompasses both frontend and backend development, database creation, and the integration of various components.

4.1 Frontend Development

As already discussed in chapter 2, for front end development, HTML, CSS and Bootstrap are used for the basic structure and styling. Its JSP is used for a dynamic and responsive user interface.



4.2 Backend Development

1. Technology Stack:

- Choose Java as the primary backend language, utilizing a web framework like Spring Boot.
- Develop RESTful APIs to handle communication between the frontend and backend.

2. Server Setup:

- Set up a robust server architecture to handle incoming requests.
- Implement load balancing to distribute traffic efficiently.



3. Database Integration:

- Connect the backend to the MySQL database.
- Implement data access and manipulation logic using database queries.

4. Database Schema:

- Create a well-designed relational database schema that includes tables for users, blog posts, interactions, etc.
- Establish relationships between tables to maintain data integrity.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	A
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
title	VARCHAR(500)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
content	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
pdate	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	A
pdate	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
userid	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
filename	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5. Data Security Measures:

- Implement encryption for sensitive user data stored in the database.
- Set up proper access controls to restrict unauthorized access to the database.

6. User Authentication:

- Implement secure user authentication with features like password hashing and token-based authentication.
- Integrate multi-factor authentication options for enhanced security.

```
String name=request.getParameter("fname");
String email=request.getParameter("uemail");
String password=request.getParameter("upassword");
```

```
UserDetails us=new UserDetails();
us.setName(name);
us.setEmail(email);
us.setPassword(password);
```

```
UserDAO dao=new UserDAO(DBConnect.getConn());
boolean f=dao.addUser(us);
HttpSession session;
```

```
if(f)
{
```

4.3. User Experience (UX) Refinement:

1. Responsive Design:

- Optimize the user interface for responsiveness on various devices.
- Conduct thorough testing to ensure a seamless user experience across browsers and screen sizes.

2. Feedback Mechanisms:

- Implement visual feedback for user actions, such as successful content uploads or interaction notifications.
- Ensure clear error messages and guidance for users in case of issues.

4.4 Testing:

4.4.1 Unit Testing:

- Conduct unit tests for individual components, functions, and modules.
- Use testing frameworks such as JUnit for Java to verify the correctness of code at a granular level.

4.4.2 Integration Testing:

- Test the interaction between different components, APIs, and modules.
- Ensure that data flows seamlessly between the frontend and backend components.

4.4.3 User Acceptance Testing (UAT):

- Invite a group of users to test the website in a real-world environment.
- Gather feedback on the user interface, functionalities, and overall user experience.

4.4.4 Security Testing:

- Perform security audits and vulnerability assessments.
- Identify and address potential security threats, including SQL injection, cross-site scripting (XSS), and data breaches.

4.4.5 Performance Testing:

- Test the website's performance under various conditions, including normal and peak loads.
- Identify and address any bottlenecks or performance issues.

4.4.6 Compatibility Testing:

- Test the website on different web browsers and devices to ensure cross-browser and cross-device compatibility.
- Resolve any layout or functionality issues specific to certain browsers or devices.

4.4.7 Accessibility Testing:

- Verify that the website adheres to accessibility standards (e.g., WCAG).
- Test the website with assistive technologies to ensure inclusivity for users with disabilities.

4.4.8 Documentation Review:

- Review and update technical documentation to reflect any changes made during the implementation and testing phases.
- Ensure that documentation is comprehensive for both developers and administrators.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Conclusion

ENotes Web Application is your go-to-place for notes and you can register or login easily then once in, you can write, view,edit, and delete your notes hassle-free.Its designed to be straightforward and user-friendly.This application makes note taking simple and efficient. "ENote Web Application" webapp marks the realization of a creative and dynamic platform that empowers users to share their stories through visually compelling collages. Throughout the development process, the team has focused on creating a user-friendly interface, implementing robust functionalities, and ensuring the security and scalability of the platform.

5.1 User-Centric Experience:

The platform offers a unique and intuitive experience, allowing users to transcend traditional blogging by seamlessly integrating visual elements into their narratives. The responsive design and interactive features contribute to a positive and engaging user journey.

5.2 Community Building:

The incorporation of social interaction features fosters a sense of community among users. The ability to like, share, and comment on collages enhances the collaborative spirit of the platform, encouraging connections and conversations among creators.

5.3 Security and Reliability:

- Stringent security measures, including user authentication, authorization protocols, and data encryption, have been implemented to safeguard user information. The platform's reliability is underscored by its robust server architecture and fault-tolerant design.

5.4 Future Scope:

1. Mobile Applications:

- Consider developing mobile apps for iOS and Android platforms.

2. Enhanced Security Measures:

- Continuously update and strengthen security measures to protect user data and implement encryption, secure authentication methods, and regularly conduct security audits to identify and address vulnerabilities.

3. Collaborative Features:

- Introduce collaborative note-taking features that allow multiple users to work on and edit notes in real-time.

4. Advanced Search and Organization:

- Implement advanced search functionalities, tags, and categories to help users organize and find their notes more efficiently.

REFERENCES

YouTubeLink

(<https://youtu.be/Yb688IQjPnc?si=IiiDCdWq3qlDXww1>)

ChatGpt

(<https://chat.openai.com/>)

Tutorials Point

(<https://www.tutorialspoint.com/servlets-and-jsp-tutorial-for-beginners/index.asp>)

APPENDICES

Sample Code:

```

package com.DAO;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import com.entity.Note;

import com.entity.UserDetails;

public class PostDAO {

    private Connection conn;

    public PostDAO(Connection conn) {

        super();

        this.conn = conn;

    }

    public boolean AddNotes(String ti, String co, int ui, String time, String fileName) {

        boolean f = false;

        try {

            String qu = "insert into notes(title,content,pdate,userid,filename)
values(?,?,?,?,?)";

            PreparedStatement ps = conn.prepareStatement(qu);

            ps.setString(1, ti);

            ps.setString(2, co);

            ps.setString(3, time);

            ps.setInt(4, ui);

            ps.setString(5, fileName);int i = ps.executeUpdate();

            if (i == 1) f = true;

        }

        } catch (Exception e) {

```

```

        e.printStackTrace();
    }
    return f;
}

public List<Note> getData(int id) {
    List<Note> list = new ArrayList<Note>();
    Note po = null;
    try {
        String qu = "select * from notes where userid=? order by id DESC";
        PreparedStatement ps = conn.prepareStatement(qu);
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            po = new Note();
            po.setId(rs.getInt(1));
            po.setTitle(rs.getString(2));
            po.setContent(rs.getString(3));
            po.setPdate(rs.getString(4));
            po.setUserId(rs.getInt(5));
            po.setFileName(rs.getString(6));
            list.add(po);
        }
    } catch (Exception e) {
    }
    return list;
}

public Note getDataById(int noteId) {
    Note po = null;
    try {
        String qu = "select * from notes where id=?";

```

```

        PreparedStatement ps = conn.prepareStatement(qu);
        ps.setInt(1, noteId);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            po = new Note();
            po.setId(rs.getInt(1));
            po.setTitle(rs.getString(2));
            po.setContent(rs.getString(3));
            po.setPdate(rs.getString(4));
            po.setUserId(rs.getInt(5));
            po.setFileName(rs.getString(6));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return po;
}

public boolean PostUpdate(int nid, String ti, String co, String time, String fileName) {
    boolean f = false;
    try {
        String qu = "update notes set title=?,content=?,pdate=?,filename=? where id=?";
        PreparedStatement ps = conn.prepareStatement(qu);
        ps.setString(1, ti);
        ps.setString(2, co);
        ps.setString(3, time);
        ps.setString(4, fileName);
        ps.setInt(5, nid);
        int i = ps.executeUpdate();
        if (i == 1) {
            f = true;
        }
    }
}

```

```

        }
    } catch (Exception e) {
        e.printStackTrace();
    }
return f;
}

public boolean DeleteNotes(int nid) {
    boolean f = false;
    try {
        String qu = "delete from notes where id=?";
        PreparedStatement ps = conn.prepareStatement(qu);
        ps.setInt(1, nid);
        int x = ps.executeUpdate();
        if (x == 1) {
            f = true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return f;
}

public List<Note> getDataBySearch(String ch, int uid) {
    List<Note> list = new ArrayList<Note>();
    Note po = null;
    try {
        String qu = "select * from notes where title like ? and userid=? order by id DESC";
        PreparedStatement ps = conn.prepareStatement(qu);
        ps.setString(1, "%" + ch + "%");
        ps.setInt(2, uid);
        ResultSet rs = ps.executeQuery();

```

```

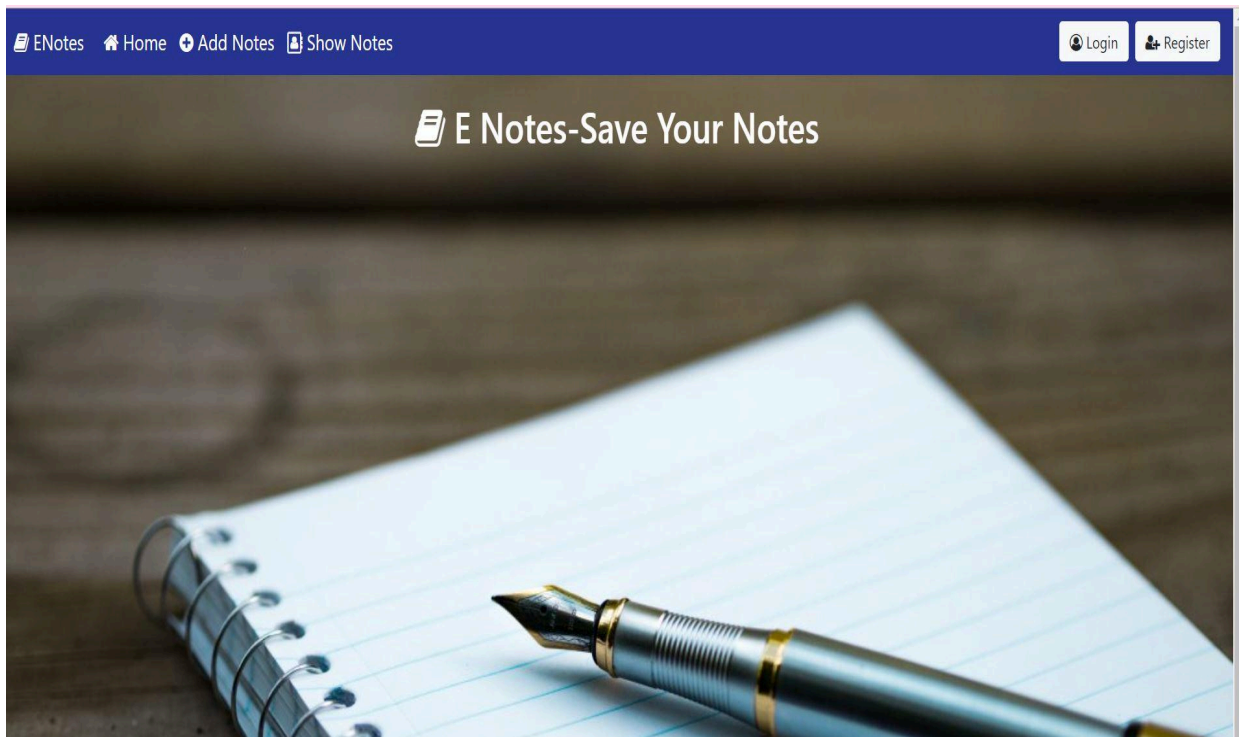
        while (rs.next()) {
            po = new Note();
            po.setId(rs.getInt(1));
            po.setTitle(rs.getString(2));
            po.setContent(rs.getString(3));
            po.setPdate(rs.getString(4));
            po.setUserId(rs.getInt(5));
            po.setFileName(rs.getString(6));
            list.add(po);
        }
    } catch (Exception e) {
    }
    return list;
}

public List<Note> getAllNotes() {
    List<Note> list = new ArrayList<Note>();
    Note po = null;
    try {
        String qu = "select * from notes order by id DESC";
        PreparedStatement ps = conn.prepareStatement(qu);
        ResultSet rs = ps.executeQuery();
        while (rs.next()) {
            po = new Note();
            po.setId(rs.getInt(1));
            po.setTitle(rs.getString(2));
            po.setContent(rs.getString(3));
            po.setPdate(rs.getString(4));
            po.setUserId(rs.getInt(5));
            po.setFileName(rs.getString(6));
            list.add(po);
        }
    }
}

```


```
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return list;  
}  
}
```


Screenshots:



User Interface

[ENotes](#) [Home](#) [Add Notes](#) [Show Notes](#) [Login](#) [Register](#)


Registration


Enter Full Name

Enter Email

Enter Password

Registration Page

[ENotes](#) [Home](#) [Add Notes](#) [Show Notes](#) [Login](#)


Login Page

Enter Email

Enter Password

Login

Login Page

[ENotes](#) [Home](#) [Add Notes](#) [Show Notes](#)


Notes Search

Search

Anjani Shukla

Logout


All Notes:



Blog
blog introduction .

Published By: Anjani Shukla | Published Date: 01/18/2024 10:30 | upload file: @ images1.png

Edit Delete



aasdsff
assfdfdg .

Published By: Anjani Shukla | Published Date: 12/02/2023 08:47 | upload file: @ ANJANI SHUKLA RESUME.jpg

Edit Delete

Show Notes