SNIPPET GUIDE
A PROJECT REPORT

Submitted By

AKSHITA GUPTA (202410116100017)
ANSHU NISHAD (202410116100033)
AKANSHA TYAGI (202410116100014)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of

Ms. Divya Singhal
Assistant Professor

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

(DECEMBER- 2024)

CERTIFICATE

Certified that **ANSHU NISHAD(202410116100033), AKSHITA GUPTA(202410116100017), AKANSHA TYAGI(202410116100014)** has/ have carried out the project work having **"SNIPPET GUIDE" (Mini Project-I, K24MCA18P) 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 themselves 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.

Ms. Divya Singhal
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Arun Kr. Tripathi
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

SNIPPET GUIDE

AKSHITA GUPTA (202410116100017)
ANSHU NISHAD (202410116100033)
AKANSHA TYAGI (202410116100014)

ABSTRACT

A snippet guide is a concise resource designed to help users or developers quickly understand and apply specific functionality or solve targeted problems. Snippets are compact, reusable pieces of code, text, or instructions tailored for efficiency and clarity. These guides streamline workflows by providing a structured approach to addressing common tasks or challenges.

Typically, a snippet guide includes a brief explanation of the problem or context it addresses, followed by the snippet itself in a clean, easy-to-understand format. Step-by-step instructions or usage notes often accompany the snippet, ensuring users can implement it effectively. Examples and variations demonstrate its application in real-world scenarios, while best practices offer insights on optimization, error handling, and adherence to industry standards.

Snippet guides are particularly valuable in technical fields like programming, data analysis, and web development, where speed and precision are essential. They enable rapid prototyping, enhance learning, and foster consistency across projects by providing ready-to-use solutions. By focusing on clarity and usability, snippet guides empower users to achieve results efficiently, making them an indispensable tool for both beginners and experienced professionals

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her 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 Dean, 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 in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds 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.

Anshu Nisad

Akshita Gupta

Akansha Tyagi

TABLE OF CONTENTS

CHAPTER 1

Introduction

1.1PROJECT DESCRIPTION

The **Snippet Guide** is a full-stack web application designed to provide users with a seamless and personalized learning experience. It allows users to log in and log out securely, select courses from a comprehensive catalog, and access course content in the form of concise, easy-to-understand snippets. This platform streamlines the learning process, making educational resources more accessible, organized, and user-friendly.

The Snippet Guide includes essential features like user authentication, a dynamic course selection system, and an intuitive content display. With user-friendly navigation, it enables learners to track their progress and bookmark important sections for future reference.

Built using modern web development technologies, the Snippet Guide leverages **React** for a responsive frontend, **Node.js with Express.js** for backend logic, and **MongoDB** for data storage. Secure authentication is implemented using **JWT (JSON Web Tokens)**.

The Snippet Guide is designed to be modular and scalable, making it adaptable for future expansion. It benefits users by providing a centralized learning platform, while for developers, it serves as a comprehensive full-stack project that highlights essential web development skills, such as frontend design, backend development, and database management.

1.2 PROJECT SCOPE

The **Snippet Guide** project aims to develop a comprehensive, user-friendly, and secure full-stack web application that facilitates efficient access to educational content. This platform will enable users to log in, browse and select courses,

and access relevant course materials in a structured and organized manner. The scope of this project outlines the core functionalities, features, development requirements, and limitations to ensure successful project completion.

1.3 PROJECT OVERVIEW

The **Snippet Guide** is a full-stack web application designed to enhance the way users access, manage, and consume educational content. This platform allows users to log in, select courses, and access relevant course materials through concise, easy-to-read snippets. The system is designed to provide a smooth, user-friendly experience, combining modern design principles with robust back-end support to ensure security, scalability, and performance.

The project aims to bridge the gap between learners and structured course content by offering a personalized learning journey. By incorporating essential features such as user authentication, course selection, and content management, the Snippet Guide empowers users to navigate learning materials at their own pace. It also serves as a comprehensive project for developers to demonstrate their proficiency in full-stack development, user authentication, and database integration.

CHAPTER 2

Feasibility Study

Feasibility Study of Snippet Guide

A feasibility study is essential to assess whether the **Snippet Guide** project can be successfully developed and implemented. This study evaluates the project's technical, operational, economic, and schedule feasibility, ensuring that it aligns with the intended goals, available resources, and potential challenges. Below is a comprehensive feasibility analysis for the Snippet Guide project.

2.1 Technical Feasibility

Technical feasibility examines whether the technology required to develop the Snippet Guide is available, accessible, and sufficient for the project's goals.

2.1.1. Required Technologies

- **Frontend**: React js for the frontend development and tailwind css for the

  For the design.

- **Backend**: Node.js with Express.js for server-side logic and API development.
- **Database**: MongoDB to store user data, course information, and content snippets.
- **Authentication & Security**: JWT (JSON Web Token) for secure session management.

The Snippet Guide is technically feasible as the required tools, technologies, and developer skills are readily available. The stack (React.js, Node.js, MongoDB) is commonly used in modern web development, ensuring ease of development and future scalability.

2. Operational Feasibility

Operational feasibility determines if the Snippet Guide will function as intended and meet the needs of its target users.

2.2. User Requirements

- Users should be able to register, log in, and log out securely.
- Users should be able to browse, select, and access relevant courses.
- Users should have access to personalized dashboards that display their selected courses, bookmarked snippets, and learning progress.

2.2.1 Ease of Use

- The user interface will be intuitive and user-friendly, leveraging modern design principles.
- Features like course search, filtering, and progress tracking will enhance user experience.
- Mobile responsiveness will ensure accessibility on mobile devices, tablets, and desktops.

2.2.2. Maintainability

- The platform will be modular and scalable, allowing new features to be added easily.
- Course content updates can be done by administrators or content creators, if the admin dashboard is included.
- Regular updates, bug fixes, and patches can be easily deployed using Continuous Integration/Continuous Deployment (CI/CD) pipelines.

2.3. Economic Feasibility

Economic feasibility evaluates the cost-benefit ratio of developing the project.

2.3.1. Cost Analysis

- **Development Costs**: If the project is developed in-house, the cost will mainly involve developer salaries. However, for self-driven or learning-based projects, this cost is negligible.
- **Hosting and Deployment Costs**:
    - Cloud platform like vercel for the deployment.
    - Domain registration and SSL certification for secure HTTPS access.

- **Software Tools and Resources**: Most tools (like Visual Studio Code, Git, and Postman) are free.

2.3.2. Benefits of the Project

- **Revenue Generation**: Monetization opportunities exist through subscription models, premium courses, or advertisements.
- **Cost Saving**: Open-source technologies (React.js, Node.js, MongoDB) reduce software costs.
- **Educational Benefits**: The platform can be marketed to schools, colleges, and individual learners as an e-learning tool.

CHAPTER 3

Project Objectives

The **Snippet Guide** is a full-stack project aimed at creating a user-friendly, interactive platform for learners to access and consume course content efficiently. Below are the key project objectives, each explained in detail.

1. User Authentication and Security

Objective: To create a secure login, registration, and logout system for users.

Details:

- Ensure users can create accounts, log in, and log out securely using industry-standard security protocols.
- Sessions will be managed using **JWT (JSON Web Tokens)** to enable secure and stateless authentication.
- Implement **multi-factor authentication (optional)** for an added layer of security.
- Users will have the option to reset passwords securely via an email link.

Benefit: Enhances user confidence and trust in the platform, ensuring only authorized users have access to their personal data and courses.

2. Course Selection and Enrollment

Objective: To allow users to add courses of their choice.

Details:

- Display course details, including course title, description, difficulty level, and content highlights.

- Allow users to **enroll in multiple courses** and store the list of active courses in their user profile.
- Store course selections in the database, enabling users to resume learning even after logging out.

Benefit: Offers users a personalized learning experience, allowing them to choose courses that align with their goals and preferences.

3. Content Delivery Through Snippets

Objective: To deliver course content in small, digestible snippets that enhance the learning experience.

Details:

- Structure content into **bite-sized snippets** for better user comprehension and retention.
- Content can include text, images, infographics, and interactive examples.
- Each snippet will be connected to a larger course structure, making it easier for users to follow the learning path.
- Snippets will be **mobile-friendly** for accessibility across devices.

Benefit: Reduces cognitive overload and improves learning efficiency, as users can focus on short, well-defined content pieces.

4. Multi-Device Compatibility and Responsiveness

Objective: To ensure the platform works on desktops, tablets, and mobile devices.

Details:

- Use **responsive web design** principles to create a UI that adjusts seamlessly to different screen sizes and devices.
- Ensure that course content, dashboards, and snippets are accessible on mobile, tablet, and desktop devices.
- Provide a **mobile-first design** to make learning on-the-go possible for users using smartphones.
- **Benefit:** Increases accessibility for users, allowing them to learn anytime, anywhere, and on any device.

CHAPTER 4

Hardware and Software Requirements

To successfully develop and deploy the **Snippet Guide** project, certain hardware and software resources are required. These requirements ensure smooth development, testing, and deployment of the platform. Below is a brief overview of the hardware and software essentials.

1. Hardware Requirements

1. **Development Machine (Local System)**
    - **Processor**: Intel i5 (or equivalent) or higher for faster development and testing.
    - **RAM**: Minimum 8 GB (16 GB recommended) to run multiple tools like IDEs, browsers, and testing environments simultaneously.
    - **Storage**: At least 250 GB of available disk space to store development tools, code files, and dependencies.
    - **Display**: A screen with a resolution of at least 1366 x 768 pixels for better IDE visibility.
    - **Peripherals**: A keyboard, mouse, and optional external monitor for better productivity.
2. **Server (For Deployment)**
    - **Processor**: Cloud server with at least 2 vCPUs to handle concurrent requests.
    - **RAM**: 2 GB for small-scale projects; 4 GB or more for larger traffic.
    - **Storage**: SSD storage (20-50 GB) for storing user data, course content, and system files.
    - **Network**: High-speed internet connection for continuous deployment and server requests.
    - **Backup and Recovery**: Backup services to prevent data loss due to server failure.

CHAPTER 5

Project Flow

The **project flow** outlines the step-by-step process from user interaction to backend processing, providing a clear understanding of how users engage with the system and how different components work together. The flow describes the journey of users as they log in, select courses, access learning content, and track their progress.

1. User Authentication and Authorization

This is the first step in the project flow. Users must register, log in, and gain access to the platform.

Steps Involved:

1. **User Registration**:
    - New users sign up by entering their email, username, and password.
    - A welcome email may be sent to the user's registered email address (optional).
2. **User Login**:
    - Users enter their email and password.
    - The backend verifies the email and password combination.
    - If verified, a **JWT (JSON Web Token)** is generated and sent to the user's browser for session tracking.
    - The JWT is stored in local storage or as an HTTP-only cookie for secure, stateless authentication.
3. **User Authorization**:
    - The system checks if the user is authorized to access specific resources (like courses, dashboard, etc.).
    - If the user is not authorized, they are redirected to the login page or shown an error message.

Outcome: The user is now logged in, and the system is aware of their identity and access rights.

2. User Dashboard and Course Selection

Once the user is authenticated, they are redirected to the dashboard, where they can view courses and track progress.

Steps Involved:

1. **Dashboard Display**:
    - The user sees their enrolled courses, progress tracking, and a list of suggested or recommended courses.
    - A "Resume Learning" button appears, allowing users to continue where they left off.
2. **Course Browsing and Selection**:
    - Users can browse available courses displayed in categories such as "New Courses" or "Trending Courses."
    - A search bar and filter options (by category, difficulty, etc.) help users locate specific courses.
3. **Course Enrollment**:
    - Users can click on a course to view its description, objectives, and content outline.
    - If interested, the user clicks "Enroll."
    - The backend updates the user's enrolled courses in the database.
    - The course is now visible on the user's dashboard under the "My Courses" section.

Outcome: The user is now enrolled in one or more courses, which are available for further interaction.

3. Course Content and Snippet Navigation

Once a user selects a course, they can start consuming content in the form of snippets.

Steps Involved:

1. **Course Content Structure**:
    - The course is divided into chapters, sections, or modules.

    - Each section contains multiple **snippets** — small, digestible pieces of content, such as text, code samples, or images.
2. **Content Navigation**:
    - The user sees a progress bar for the entire course.
    - Users can navigate between snippets using "Next" and "Previous" buttons.
    - A table of contents allows the user to jump to specific snippets or chapters.
3. **Snippet Interaction**:
    - Users can bookmark snippets to review them later.
    - Users can mark snippets as "Completed" once they've finished reading them.
    - The system tracks which snippets have been completed and updates the user's course progress accordingly.
4. **Interactive Content (Optional)**:
    - Snippets may include quizzes, coding challenges, or exercises.
    - The user submits their answers, and the system provides feedback in real time.

Outcome: The user consumes the learning content, interacts with snippets, and tracks their progress as they move through the course.

4. Progress Tracking and Bookmarking

As the user navigates through snippets and completes them, the system tracks their learning journey.

Steps Involved:

1. **Progress Updates**:
    - Each time a user marks a snippet as "Completed," the backend updates the user's progress.
    - The system calculates the overall completion percentage for the course.
2. **Bookmarking**:
    - Users can bookmark specific snippets for later reference.
    - Bookmarked snippets are stored in the database and displayed on the user's dashboard.
3. **Resume Learning**:

- If a user logs out, the system remembers where they left off.
- When the user logs back in, they see a "Resume Learning" button on the dashboard, allowing them to pick up where they left off.

Outcome: User progress is saved, and the user can resume the course at any time. The system provides insights into the user's learning progress, which serves as a motivational factor

5. User Controls

user role is responsible for managing the platform, including users, courses, and content.

Steps Involved:

1. **User Login**:
   - Admins log in through a secure admin portal.
   - The system verifies that the user has admin privileges before granting access.
2. **User Management**:
   - Admins can view, delete, or modify user accounts.
   - Admins can reset passwords for users who forget their login details.
3. **Course Management**:
   - Admins can add, update, or delete courses and snippets.
   - Admins can edit course descriptions, objectives, and snippet content.
4. **Reports and Analytics**:
   - Admins can generate reports on user engagement, course popularity, and completion rates.

Outcome: User have full control of there data, course content, and system operations, ensuring smooth platform management.

6. Error Handling and User Support

To ensure a smooth user experience, error handling and support mechanisms are built into the system.

Steps Involved:

1. **Error Handling**:
    - When a user encounters an error (e.g., course not found, snippet not loading), a user-friendly message is displayed.
    - System errors (like "500 Internal Server Error") are logged, and administrators are notified for quick resolution.
2. **Customer Support**:
    - Users can submit queries or issues using a "Contact Us" form.
    - User queries are stored in the database or sent via email to the support team.
3. **System Alerts**:
    - If the server crashes or an API fails, the system sends an alert to the development team via tools like **Sentry** or **AWS CloudWatch**.

Outcome: User issues are resolved quickly, ensuring a smooth and frustration-free learning experience.

7. Logging Out and Session Termination

When users finish their learning session, they can log out.

Steps Involved:

1. **User Logout**:
    - The user clicks the "Logout" button on the dashboard.
    - The system deletes the **JWT token** from local storage or the HTTP-only cookie.
    - The user's session is terminated, and they are redirected to the login page.
2. **Session Expiry**:
    - If the user stays inactive for a specific duration, the session automatically expires.
    - The user is prompted to log in again for security reasons.

Outcome: The user's session is securely ended, protecting their account from unauthorized access.

FLOW CHART:

It outlines the main steps, from user login or registration to selecting a course, reading its content, and logging out.

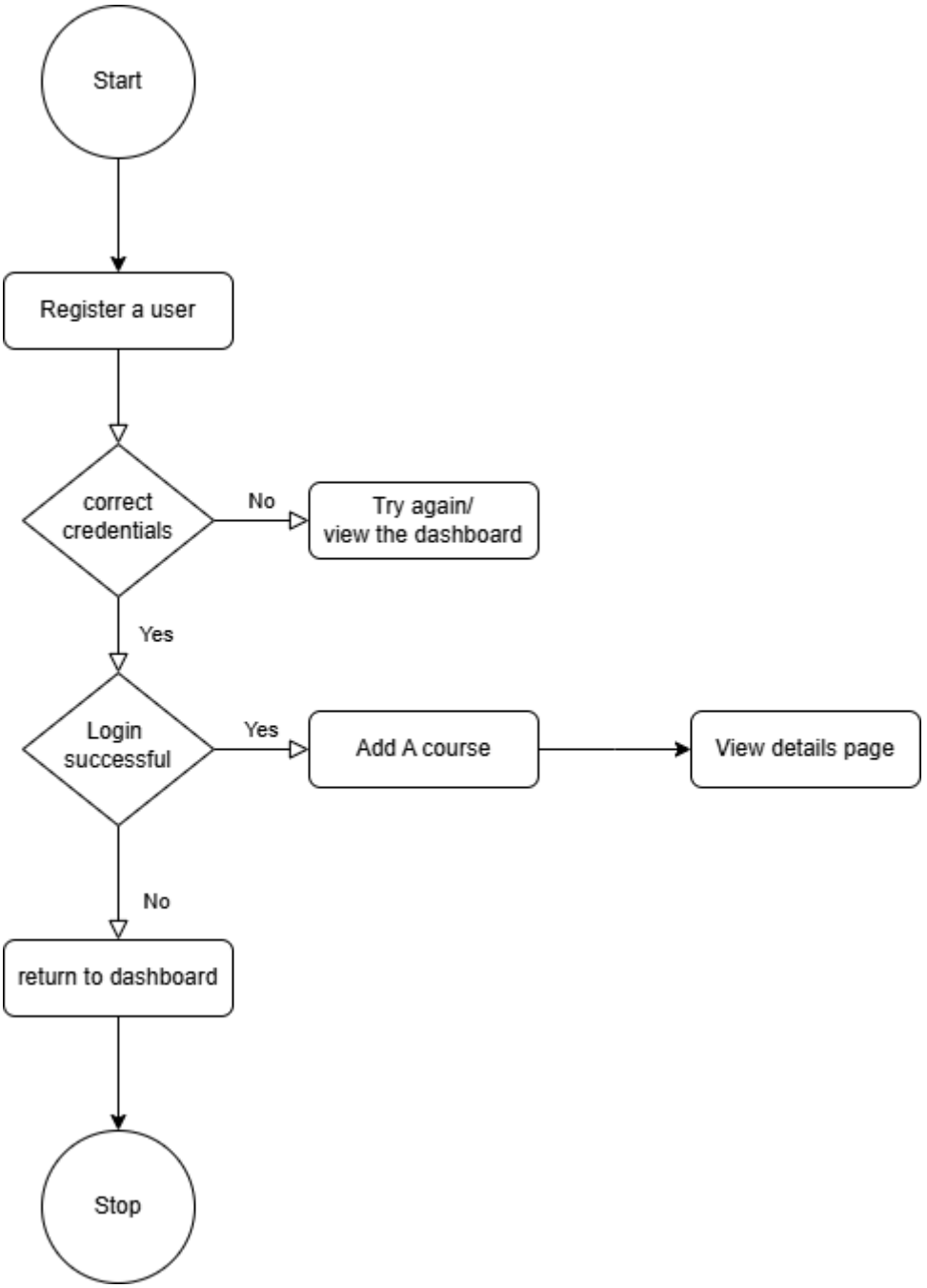

Fig1: Flow chart of Snippet Guide

Steps in the Flowchart

1. **Start**
    - The process begins here.

2. **Register a User**
    - The system prompts the user to register by providing their details such as email, username, and password.
3. **Correct Credentials Check**
    - The system checks if the credentials entered are correct:
        - **Yes**: Proceed to the next step.
        - **No**: The user can either **try again** or **view the dashboard**.
4. **Login Successful**
    - The system verifies if the login is successful:
        - **Yes**: Proceed to the "Add a Course" step.
        - **No**: Redirect the user to **return to the dashboard**.
5. **Add a Course**
    - The user adds a course of their choice to the system.
6. **View Details Page**
    - After adding the course, the system displays the **details page** for the course.
7. **Return to Dashboard (if Login Fails)**
    - If login fails, the user is redirected to the dashboard for further actions.
8. **Stop**
    - The process ends here.

USE CASE DIAGRAMS:

USER AUTHENTICATION:

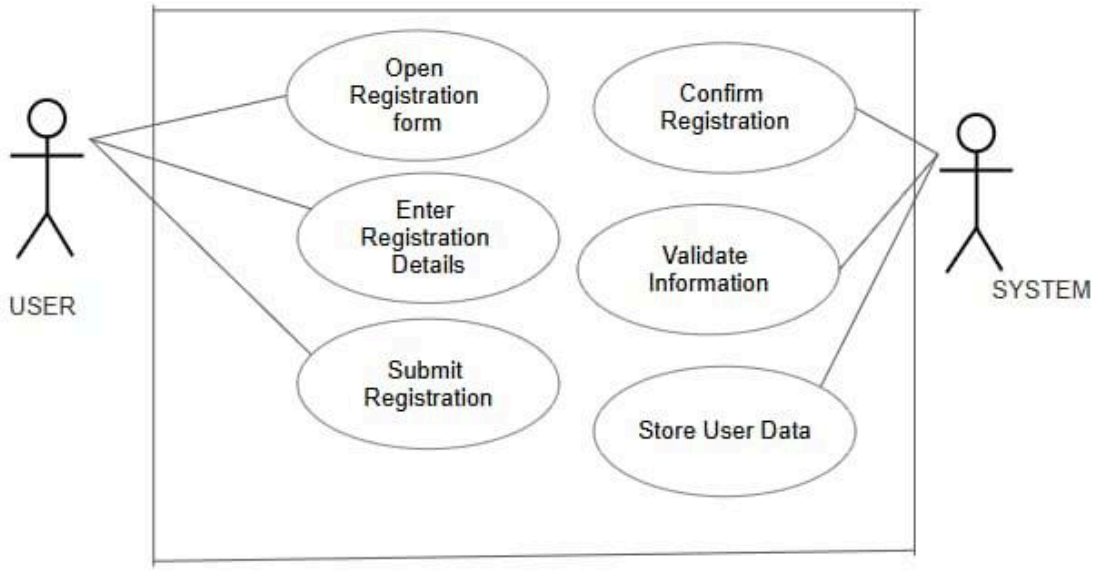

Fig1:Use Case showing user authentication

The actors are users and the system where users can register through a registration form and the system validates and stores the credentials.

SELECTING A COURSE:

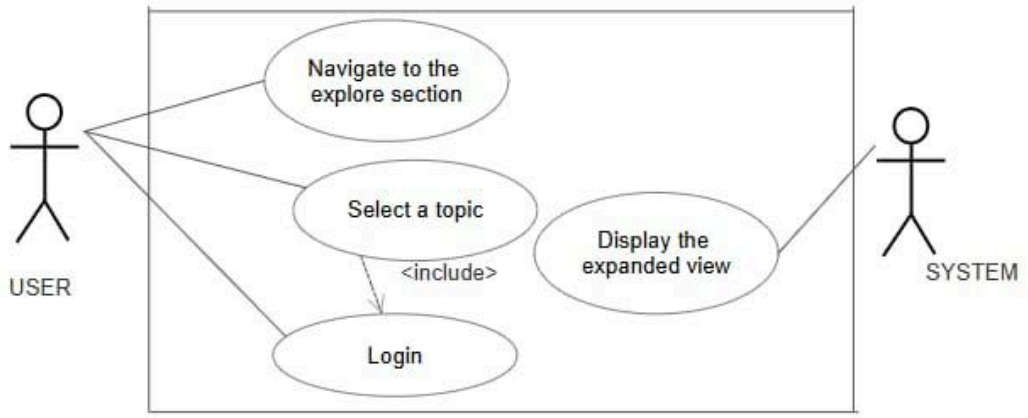

Fig2:Use Case showing selection of a course

The actors are the user and the system where the user can select a topic and read the information needed by him and the system provides the expanded view.

SAVING A COURSE:

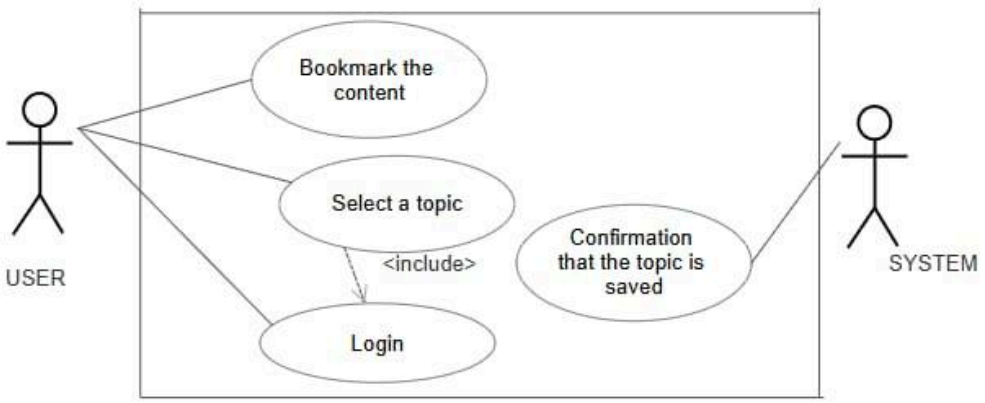

Fig3 :Use Case showing saving of a course

The actors are the user and the system where the user can bookmark a content and the system confirms that the topic has been saved.

SEQUENCE DIAGRAM:

Purpose of a Sequence Diagram To model high-level interaction among active objects within a system. To model interaction among objects inside a collaboration realizing a use case. It either models' generic interactions or some certain instances of interaction.

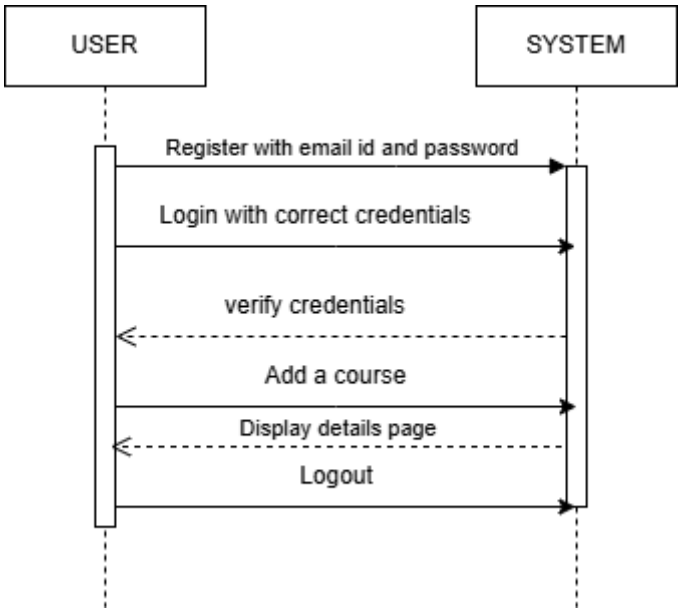

Fig1:Sequential Diagram of Snippet Guide

Actors and Lifelines

1. **User**: Represents the person interacting with the system.
    - Lifeline: A dashed vertical line extending from the "User" label at the top.
2. **System**: Represents the backend or server processing the user's requests.
    - Lifeline: A dashed vertical line extending from the "System" label at the top.

Sequence of Interactions

1. **Login/Register**:
    - The **User** sends a "Login/Register" request to the **System** (represented by an arrow from "User" to "System").

- The **System** processes the request and responds with a "Login Response" (arrow from "System" to "User").

2. **Add Course**:
   - The **User** sends an "Add Course" request to the **System**.
   - The **System** processes the request and responds with a "Course Added Confirmation" message.
3. **Select Course**:
   - The **User** sends a "Select Course" request to the **System**.
   - The **System** processes the request and responds by "Loading Course Content."
4. **Logout**:
   - The **User** sends a "Logout" request to the **System**.
   - The **System** processes the request and responds with a "Logout Confirmation."

ER DIAGRAM:

• ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

• It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.

• In ER modelling, the database structure is portrayed as a diagram called an entity-relationship diagram

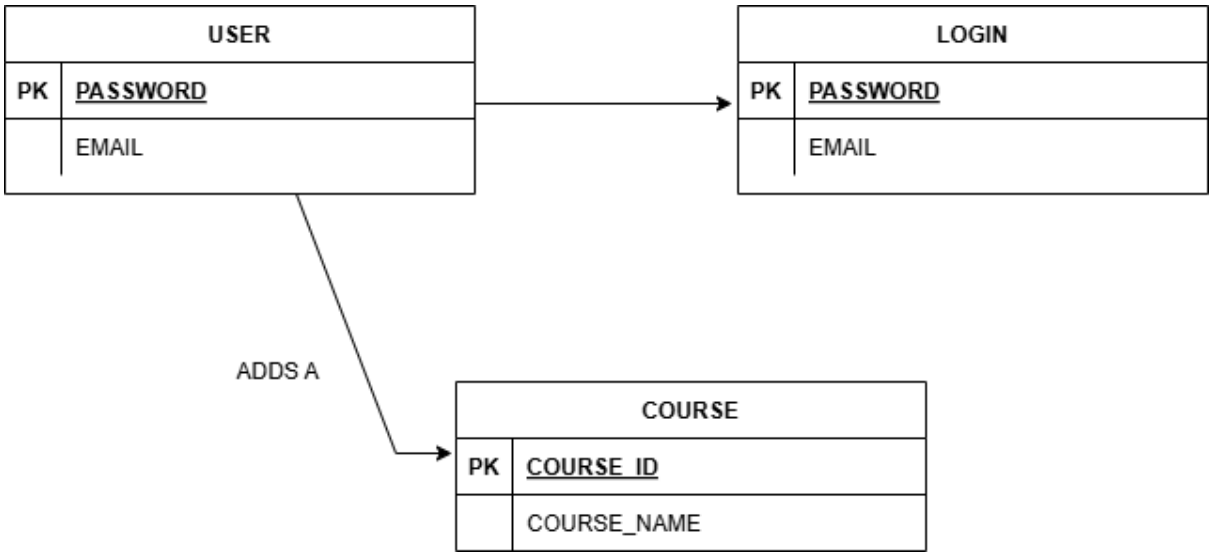

Fig1:E-R Diagram of Snippet Guide

CHAPTER 6

Project Outcome

The **Snippet Guide** project aims to deliver a user-friendly platform for managing courses and related content. Below are the expected outcomes:

Key Functional Outcomes

1. **User Authentication**:
    - Secure **login and registration** functionality.
    - Users can log in using their credentials and log out safely.
2. **Course Management**:
    - Users can **add new courses** with a description and title.
    - All added courses are stored in the database and linked to specific users.
3. **Content Management**:
    - Users can select a course and access its **associated content**.
    - Dynamic loading of course content ensures users can retrieve course materials easily.
4. **Logout Functionality**:
    - Users can log out, ensuring their session is terminated securely.

System Efficiency Outcomes

1. **Data Integrity**:
    - The database ensures that every course and its content are linked to the correct user via **primary and foreign key constraints**.
2. **Scalability**:
    - The platform is designed to support multiple users and a large number of courses efficiently.

3. **Responsiveness**:
    - The interface provides fast and seamless interactions for users (e.g., adding/selecting courses or fetching content).

User Experience Outcomes

1. **Ease of Use**:
    - The platform is simple and intuitive, making it easy for users to log in, add courses, and access content.
2. **Customization**:
    - Users have the freedom to add courses of their choice, tailoring the experience to their needs.
3. **Consistent Navigation**:
    - The flow from login to course selection and logout is smooth and user-friendly.

Learning Outcomes

1. **Backend Integration**:
    - Effective use of server-side technologies to manage authentication, data storage, and business logic.
2. **Frontend Development**:
    - Building a user interface that interacts with backend APIs for seamless functionality.
3. **Database Design**:
    - Structuring an efficient database with clear relationships between entities (User, Course, Content).
4. **Full-Stack Development**:
    - Demonstrates a complete Mern-stack development workflow with technologies like Reactjs , Nodejs, Expressjs, Mongodb.

Final Deliverable

A fully functional web application where users can:

- Log in and out.
- Add courses.
- Access content related to the courses they select.

Our application ensures security, efficiency, and scalability while providing a seamless user experience.

SNAPSHOTS:

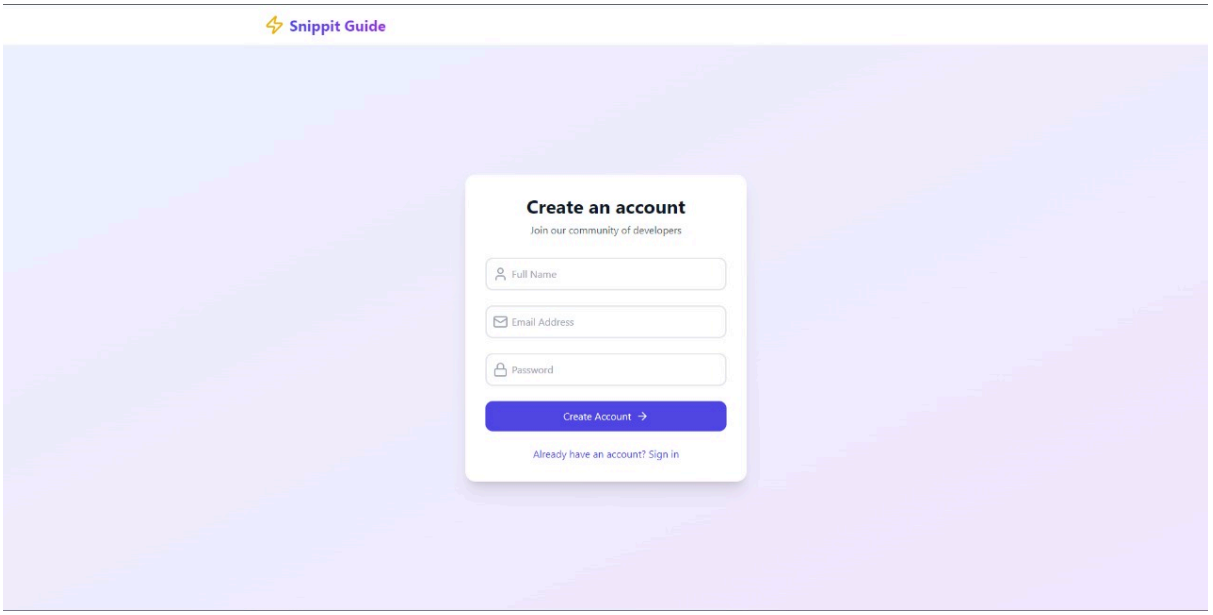

Fig1:Home Page

This is the sign up page where a new user can create his account and get access to the features of the website.

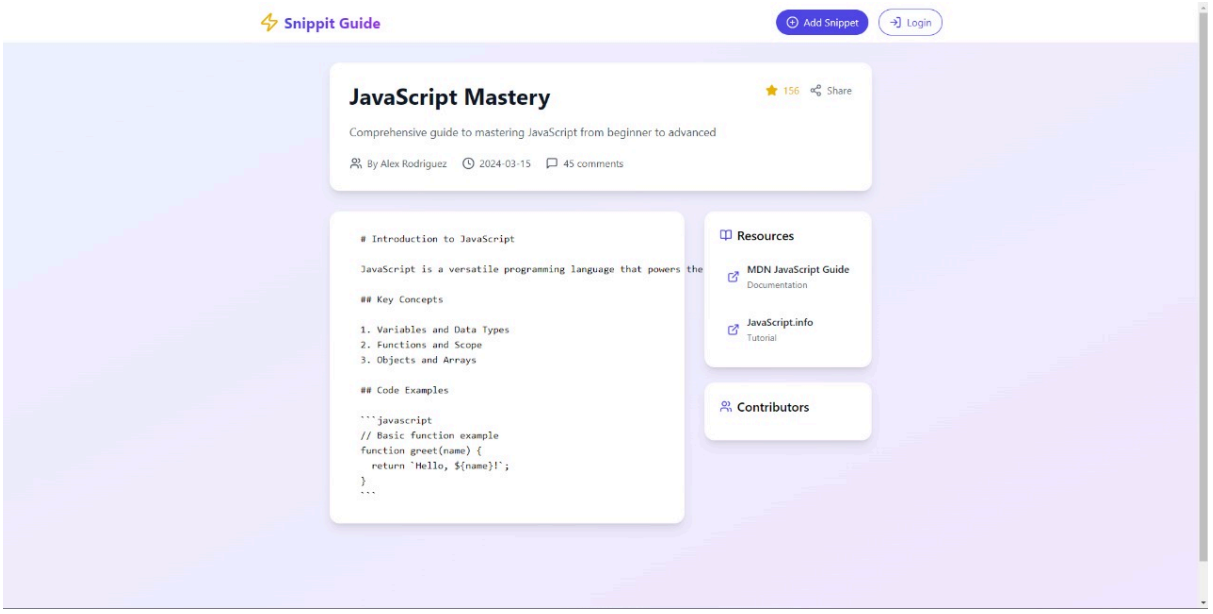

Fig2 :Details page

This section involves the details of the courses selected by the user after login.

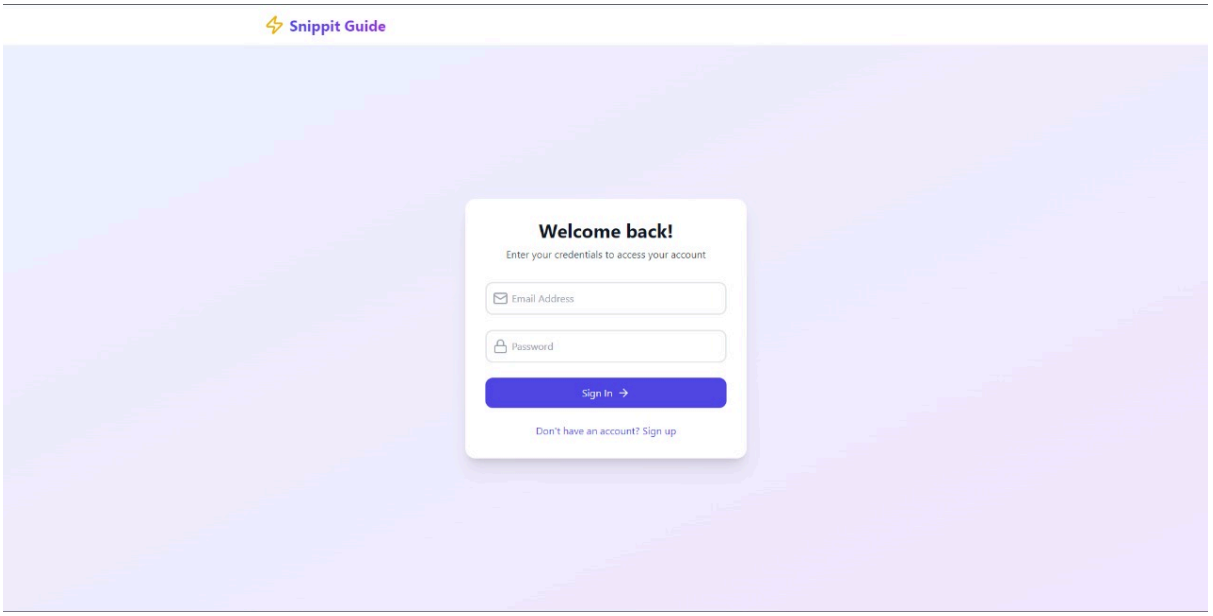

Fig3:Login Page

This is the login page of our website where the user can enter his details after signing up.

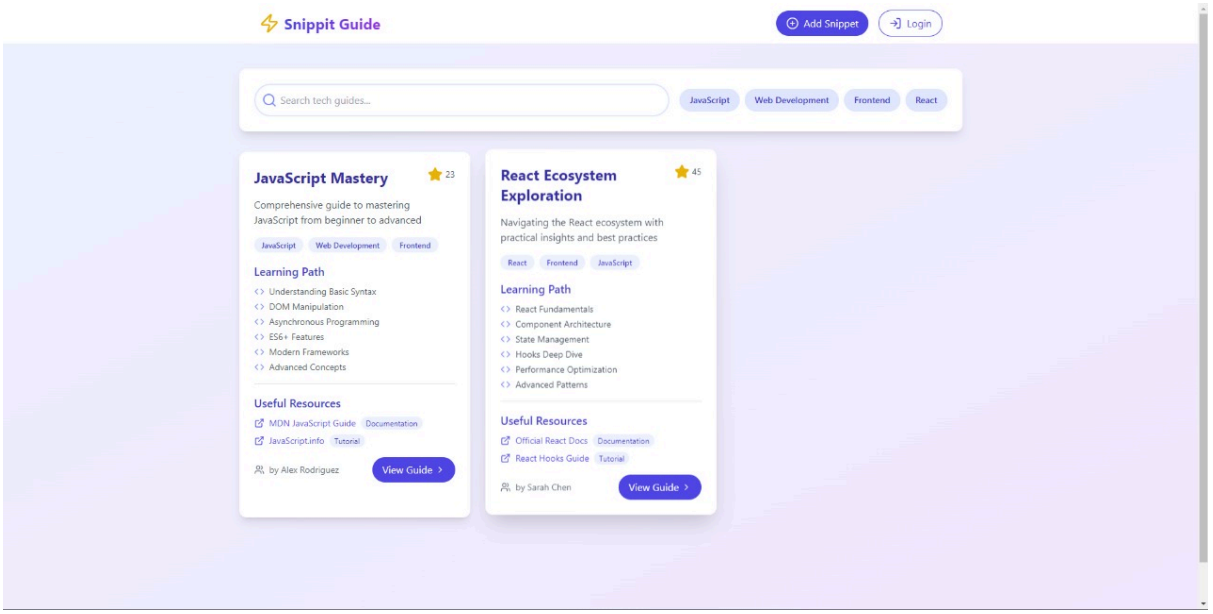

Fig4: Interface after adding a course

This section shows that the user has added a course of his choice.

REFERENCES:

1. https://www.coursera.org/learn/ibm-cloud-native-full-stack-development-capstone
2. https://www.freecodecamp.org/news/how-to-build-a-full-stack-application/
3. https://dev.to/soumyajit4419/full-stack-development-with-react-node-express-and-mongodb-1bk9
4. https://blog.logrocket.com/creating-full-stack-app-react-express/
5. https://www.geeksforgeeks.org/full-stack-development-with-mern-stack/
6. https://www.freecodecamp.org/news/how-to-build-a-full-stack-authentication-app/
7. https://hackr.io/blog/full-stack-projects
8. https://www.section.io/engineering-education/user-login-and-registration-nodejs/
9. https://www.newtonschool.co/full-stack-web-development-project-ideas/
10. https://www.edureka.co/blog/full-stack-development/