

## **1.1 Institute Profile:**

Institute of Management and Career Courses (IMCC) is a premier Management Institute, established in 1983 by Maharashtra Education Society (MES) for providing quality education and technical expertise at the Post Graduation Level in the Fields of Computers and Management. The Institute is recognized by SPPU under Section 46 of Pune University Act, 1974 and Section 85 of Maharashtra University Act, 1994 and Approved by AICTE New Delhi to conduct MCA and MBA programmes. The Institute is located at 131, Mayur Colony, Kothrud, Pune-411038 having 30,000 sq. Ft. built area & totally independent campus.

IMCC is recognised as a Ph.D. Research Centre under the Faculty of Management, SPPU. IMCC has 38 years standing & it is well-known for its conducive educational atmosphere. IMCC focuses on the all-round development of its students. Thus, apart from excellence in academics, students develop their inner potential by way of active participation in co-curricular & extra- curricular activities. IMCC has developed excellent rapport with Industry by way of Guest Lectures, Seminars, Workshops, Industrial Visits & Placements. The main motto of the Institute is to install the concepts of total personality development in the students. The emphasis is laid on Teacher Disciple Relationship in place of Boss Subordinate' relationship at their assignments.

The preamble of IMCC FACTA-NON-VERBA" lucidly means that the Institute produces the new breed of professionals, who's deeds will speak and there could be no requirement of pomposity The zooming enthusiastic, rational and excellent external endeavours are being imbibed in the students to prove their mettle The conducive milieu of the Institute Molds the budding managers to reveal in managing flexibility, integration, change and transformation. These would be professionals are channelised in such a way to 'orchestrate' and deplo-

y business and technological management skills in a synergistic manner to grab the tangible success. The faculty members put their relentless efforts in educating the students to synthesize business management acumen and technology insights in a creative manner.

## **1.2 Abstract:**

The StudyNotion E-learning Platform is designed to revolutionize the educational experience by providing a seamless online environment where students can access a wide range of courses and educational content. This platform caters to needs of courses and educational content. This platform caters to the needs of both students and instructors, facilitating direct interaction between them in a B2C model. Through user registration and authentication, individuals can engage with the platform, fostering a community-driven approach to learning. With a focus on meeting diverse educational needs, StudyNotion continually innovates to enhance accessibility, efficiency and affordability. By offering courses that cater to various interests and skill levels, the platform aims to make learning more convenient and rewarding for users. Additionally, features such as ratings and feedback mechanisms empower learners to make informed decisions and contributed to the platform's growth.

The local context is crucial, as StudyNotion aims to foster trust and familiarity among users. Just as individuals trust local merchants for their immediate needs, students can rely on StudyNotion to provide relevant and timely educational resources. Moreover, by empowering local instructors to showcase their expertise, the platform promotes the concept of "Vocal for Local" in education, encouraging support for indigenous knowledge and talent.

In essence, StudyNotion strives to create a vibrant online learning community where students can access quality education tailored to their needs while supporting local instructors and businesses. Through its user-centric approach and emphasis on accessibility and trust, StudyNotion aims to become a trusted and popular destination for online education.

### **1.3 Existing System and Need for System:**

In the current landscape of online learning platforms, there exists a plethora of options for students seeking educational resources. However, these platforms often lack the immediacy and personalized touch that local instructors and educational institutions can offer. Students may find themselves waiting for days or even weeks to access relevant course materials or receive feedback on their progress.

Need for System:

- **Reduced Access Time:** StudyNotion aims to bridge this gap by providing a platform where students can access courses materials and interact with instructors in real-time. This minimizes the waiting period for accessing educational resources, catering to the needs of individuals who require immediate assistance or guidance.
- **Trust in Local Expertise:** Students often place trust in local instructors or educational institutions due to their familiarity with their teaching methods and expertise. By incorporating local instructors into the platform, StudyNotion enhances trust and confidence among students, ensuring access to high-quality educational content.
- **Affordability:** Leveraging local instructors and educational resources can also lead to cost savings for students. StudyNotion's emphasis on local expertise may result in more affordable courses offerings compared to traditional online learning platforms.

- **Empowering Local Educators:** By providing a platform for local instructors to showcase their expertise and reach a broader audience, StudyNotion contributes to the growth and success of local educational business. This empowerment of local educators aligns with the platform commitment to supporting and promoting indigenous knowledge and talent in the educational sector.

## **1.4 Scope of System:**

The scope of the StudyNotion project encompasses the development of a comprehensive ed-tech platform aimed at providing a seamless and interactive learning experience for students while offering a platform for instructors to showcase their expertise. The project will involve the creation of various components and functionalities, including:

### **1. User Authentication and Authorization:**

- Implementing user registration, login, and logout functionalities.
- Integrating authentication mechanisms for ensuring user data security.

### **2. Course Management System:**

- Developing features for instructors to create, update and delete courses.
- Enabling instructors to manage courses content, pricing and availability.
- Allowing students to browse, search and enrol in courses.

### **3. Content Delivery:**

- Implementing a system for delivering courses content, including videos, documents and quizzes.
- Ensuring smooth playback and access to educational materials for students.
- Providing features for bookmarking, saving progress and accessing supplementary resources.

### **4. User Profiles and Preferences:**

- Designing user profiles for both students and instructors, including personal information, course history and preferences.
- Enabling users to customize their learning experience by setting preferences and goals.

## **5. Feedback and Rating System:**

- Implementing a system for student to provide feedback and ratings for courses.
- Allowing instructors to view and respond to feedback, and to monitor course performance metrics.

## **6. Admin Dashboard and Management Tools:**

- Developing an admin dashboard for managing courses, users and platform settings.
- Providing analytics and reporting features to track platform usage and performance.

## **7. Payment Integration:**

- Integrating payment gateways for courses purchases and enrolments.
- Ensuring secure and seamless transaction processing for users.

## **8. Mobile Responsiveness and Accessibility:**

- Ensuring the platform is responsive and accessible across various devices and screen sizes.
- Optimizing user experience for mobile devices through responsive design and performance optimization.

## **9. Scalability and Performance:**

- Designing the platform architecture to handle large volumes of users and course content.
- Optimizing performance for fast loading times and smooth users' interactions.

## **10. Testing and Quality Assurance:**

- Implementing thorough testing procedures, including unit testing, integration testing and user acceptance testing.
- Ensuring high quality and reliability of the platform through rigorous testing and quality assurance processes.

## **11. Deployment and Hosting:**

- Deploying the platform to reliable hosting environments ensuring high availability and scalability.
- Configuring deployment pipelines and automation processes for seamless updates and maintenance.

## **1.5 Operating Environment – Hardware & Software:**

### **I. Hardware Requirement:**

#### **➤ Server Infrastructure:**

- StudyNotion will require server infrastructure to host the application components, including the front end, back end, and database.
- The server hardware should have sufficient CPU, RAM, and storage resources to handle concurrent user requests and data processing.
- It is recommended to use cloud-based hosting services for scalability, reliability, and ease of management.

#### **➤ Networking Equipment:**

- Reliable networking equipment such as routers, switches, and firewalls will be necessary to ensure seamless communication between clients and servers.
- High-speed internet connectivity is essential for optimal performance and user experience.

### **II. Software Requirement:**

#### **➤ Operating System:**

- The servers hosting StudyNotion can run on various operating systems, including Linux distributions (e.g., Ubuntu, CentOS) or Windows Server.
- The choice of operating system will depend on factors such as familiarity, compatibility, and performance requirements.

#### **➤ Development Tools and Frameworks:**

- **Front-end Development:**
  - **ReactJS:** JavaScript library for building user interfaces, enabling effective and interactive web applications.
  - **Figma:** This design tool is instrumental in creating intuitive and visually appealing UI/UX designs, facilitating seamless user experiences.
- **Back-end Development:**
  - **Node.js:** Serving as a JavaScript runtime environment for server-side development, Node.js empowers developers to build scalable and efficient backend solutions.
  - **Express.js:** A web application framework designed for Node.js, Express.js simplifies the development of robust and feature-rich web applications.
- **Database:**
  - **MongoDB:** This NoSQL database is employed for storing applications data, offering flexibility and scalability for managing diverse data types efficiently.
- **Other Tools:**
  - **Git:** Serving as a version control system, Git facilitates collaboration among developers and efficient code management throughout the development lifecycle.
  - **Npm (node package manager):** Npm is a package manager essential for installing and managing project dependencies, ensuring seamless integration of various modules.
  - **Visual Studio Code:** An integrated development environment (IDE) renowned for its versatility, Visual Studio Code provides powerful features for code editing, debugging, and seamless integration with other development tools.

## ➤ Deployment and Hosting Platforms:

- **Front-end Hosting:**
  - **Vercel:** Renowned for its simplicity and scalability, Vercel provides hosting services tailored for static sites built with React, ensuring optimal performance and reliability.
- **Back-end Hosting:**
  - **Render:** Leveraging cloud-based hosting services, Render offers a robust platform for hosting applications developed with Node.js and MongoDB, ensuring high availability and scalability.
- **Database Hosting:**
  - **MongoDB Atlas:** A fully managed cloud database service, MongoDB Atlas provides a reliable and scalable solution for hosting MongoDB databases, with features such as automated backups and monitoring.
- **Media Storage:**
  - **Cloudinary:** This cloud-based media management platform offers comprehensive solutions for storing and managing images, videos, and documents, ensuring seamless integration and efficient media handling within StudyNotion.

## ➤ Additional Software:

- **Text Editors:** Developers may use text editors such as Sublime Text, Atom or Notepad ++ for code editing.
- **Browser Compatibility:** StudyNotion should be compatible with popular web browser such as Google Chrome, Mozilla Firefox, Safari and Microsoft Edge.

The hardware and software components listed above provide the necessary infrastructure and tools for developing, deploying, and hosting the StudyNotion ed-tech platform. It is essential to ensure compatibility, reliability, and scalability across all components to deliver a robust and high-performance application.

## **1.6 Brief Description of Technology Used:**

### **1.6.1: Operating System Used:**

#### **➤ Unix – Based System:**

- The server infrastructure hosting StudyNotion will predominantly use Unix-based operating systems such as Linux distributions (e.g., Ubuntu, CentOS).
- Unix-based systems are preferred for their stability, security, and scalability, making them well-suited for hosting web applications.

### **1.6.2 RDBMS/NoSQL Used to Build Database:**

#### **➤ NoSQL Database: MongoDB**

- StudyNotion utilizes MongoDB as the primary database management system (DBMS) for storing and managing application data.
- MongoDB is a document-oriented NoSQL database that provides flexibility, scalability, and performance for handling unstructured and semi-structured data.
- Its schema-less nature allows for agile development and easy scalability, making it suitable for applications with evolving data requirements.
- MongoDB's features such as replication, sharding, and flexible querying capabilities make it an ideal choice for modern web applications like StudyNotion.

The combination of Unix-based operating systems and MongoDB as the database management system forms the foundation of the StudyNotion platform, providing a robust and scalable infrastructure for delivering an interactive and engaging learning experience to users.

## **2.1 Study of Similar System**

A comprehensive study of existing ed-tech platforms and similar systems is essential to understand industry trends, best practices, and potential areas for innovation. By analysing the strengths and weaknesses of similar systems, StudyNotion can identify opportunities to differentiate itself and provide unique value to users. This study may involve.

➤ **Market Research:**

- Conducting market research to identify prominent ed-tech platforms and their features.
- Analysing user reviews, ratings, and feedback to understand user preferences and pain points.
- Identifying key competitors and assessing their market positioning, business models, and user demographics.

➤ **Comparative Analysis:**

- Comparing the features, functionalities, and user experience of leading ed-tech platforms.
- Identifying common patterns, trends, and innovative features that contribute to user engagement and retention.
- Evaluating the strengths and weaknesses of each platform in terms of content quality, usability, and value proposition.

➤ **Case Studies:**

- Reviewing academic research papers, industry reports, and case studies related to online education and e-learning platforms.
- Examining studies on user behaviour, learning outcomes, and technology adoption in the context of ed-tech platforms.

- Identifying emerging technologies, pedagogical approaches, and learning methodologies that can enhance StudyNotion's offering.

➤ **User Survey and Interviews:**

- Conducting surveys and interviews with potential users, including students, instructors, and administrators.
- Gathering insights into user preferences, pain points, and expectations from an ideal ed-tech platform.
- Incorporating user feedback into the design and development process to prioritize features and functionalities.

➤ **Emerging Trends and Technologies:**

- Monitoring emerging trends and technologies in the ed-tech industry, such as AI-driven personalized learning, virtual reality (VR) simulations, and blockchain-based credentialing.
- Assessing the feasibility and potential impact of integrating these technologies into StudyNotion to enhance its competitiveness and innovation.

By conducting a thorough study of similar systems, StudyNotion can leverage insights from industry benchmarks, academic research, and user feedback to design a differentiated and compelling ed-tech platform that meets the evolving needs of students, instructors, and administrators.

## 2.2 Feasibility Study

A feasibility study is essential to assess the viability and potential success of the StudyNotion project. This study evaluates various aspects, including technical, economic, operational, and scheduling considerations, to determine if the proposed system is feasible to develop and deploy. The following factors will be analysed:

### ➤ Technical Feasibility:

- **Technology Stack:** Evaluate the compatibility, scalability, and reliability of the chosen technology stack (MERN - MongoDB, Express.js, ReactJS, Node.js) for building StudyNotion.
- **Integration:** Assess the feasibility of integrating different components, such as front-end, back-end, database, and third-party services, to ensure seamless functionality.
- **Scalability:** Determine the scalability requirements and evaluate the ability of the system to accommodate future growth in terms of users, courses, and content.

### ➤ Economic Feasibility:

- **Return on Investment (ROI):** Assess the potential ROI based on revenue projections, user acquisition strategies, and market demand for online education platforms.
- **Cost Analysis:** Estimate the costs associated with development, hosting, maintenance, and support of StudyNotion.
- **Budget Constraints:** Evaluate the project budget and ensure that the estimated costs align with available resources and funding.

### ➤ Operational Feasibility:

- **User Acceptance:** Assess the willingness of users, including students, instructors, and administrators, to adopt and use StudyNotion.
- **Training Needs:** Identify training requirements for users and administrators to effectively use and manage the platform.
- **Operational Processes:** Analyze the impact of StudyNotion on existing operational processes and identify any necessary adjustments or optimizations.

➤ **Scheduling Feasibility:**

- **Timeline:** Develop a realistic project timeline, considering the scope of work, resource availability, and dependencies between tasks.
- **Milestones:** Define key milestones and deliverables to track progress and ensure timely completion of the project.
- **Risk Management:** Identify potential risks and challenges that may impact project schedules and develop mitigation strategies to address them.

➤ **Legal And Regularity Compliance:**

- **Data Privacy:** Ensure compliance with data protection regulations, such as GDPR, CCPA, and other applicable laws governing user data privacy and security.
- **Intellectual Property:** Address any legal considerations related to intellectual property rights, licensing agreements, and content ownership rights for courses and materials.

Based on the findings of the feasibility study, stakeholders can make informed decisions regarding the viability and implementation of the StudyNotion project. Any identified challenges or constraints will be addressed through appropriate strategies to ensure the successful development and deployment of the ed-tech platform.

## **2.3 Objectives of Proposed System:**

The Objectives of StudyNotion Proposed System are as follow:

### **1. Enhanced Learning Experience:**

- Provide a seamless and interactive learning experience for students, leveraging modern technologies and user-centered design principles.
- Offer a variety of multimedia content, including videos, documents, and interactive quizzes, to cater to diverse learning preferences and styles.

### **2. Global Accessibility and Reach:**

- Ensure accessibility to educational content from anywhere, anytime, enabling students to learn at their own pace and convenience.
- Facilitate connectivity between instructors and learners worldwide, fostering a global community of knowledge sharing and collaboration.

### **3. Empowerment of Instructors:**

- Empower instructors to showcase their expertise and creativity by offering tools and resources for course creation, customization, and management.
- Provide analytics and insights to instructors to track course performance, student engagement, and feedback for continuous improvement.

### **4. Vocal for Local Initiative:**

- Support small and local educational institutions and instructors by providing a platform to showcase their courses and reach a wider audience.
- Promote the "Vocal for Local" initiative by highlighting and promoting locally curated educational content and expertise.

### **5. Data-Driven Decision Making:**

- Utilize analytics and data insights to inform strategic decision-making, optimize course offerings, and enhance user experience based on user behaviour and preferences.
- Implement feedback mechanisms to gather user feedback and iterate on the platform's features and functionalities to meet evolving needs.

## **6. Security and Privacy Compliance:**

- Ensure robust security measures to protect user data, including authentication mechanisms, data encryption, and compliance with data protection regulations.
- Safeguard intellectual property rights and ensure compliance with copyright laws for course materials and content uploaded by instructors.

## **7. Scalability and Reliability:**

- Design the platform with scalability in mind to accommodate growth in user base, course offerings, and content volume over time.
- Implement robust infrastructure and deployment strategies to ensure high availability, reliability, and performance of the platform under varying load conditions.

## **8. Continuous Innovation and Improvement:**

- Foster a culture of innovation and continuous improvement by regularly introducing new features, technologies, and educational methodologies to enhance the platform's value proposition.
- Solicit user feedback, conduct usability testing, and monitor industry trends to identify opportunities for innovation and differentiation in the ed-tech market.

By achieving these objectives, StudyNotion aims to position itself as a leading ed-tech platform, providing a transformative learning experience for students, empowering instructors, and contributing to the advancement of education globally.

## **2.4 Users of System:**

The StudyNotion system is designed to cater to the following categories of users:

### **1. Students:**

- **Description:** Individuals seeking to enhance their knowledge and skills through online courses and educational content.
- **Role:** Enrol in courses, access learning materials, participate in discussions, track progress, and provide feedback.
- **Features:** Access to course catalogue, personalized recommendations, interactive learning tools, progress tracking, and communication with instructors and peers.

### **2. Instructors:**

- Description: Educators, subject matter experts, and content creators offering courses and educational content on the platform.
- Role: Create and publish courses, manage course content, interact with students, monitor course performance, and receive feedback.
- Features: Course creation tools, content management, analytics and insights, communication tools, and support for course customization and branding.

### **3. Administrators (Future Scope):**

- Description: Platform administrators responsible for managing users, courses, content, and overall system operations.
- Role: Monitor platform activity, manage user accounts, moderate content, configure platform settings, and ensure compliance with policies and regulations.
- Features: User management tools, content moderation, analytics dashboards, system configuration settings, and access to administrative controls.

#### **4. Guest Users:**

- Description: Individuals exploring the platform without creating an account or logging in.
- Role: Browse the course catalogue, view course previews, read platform information, and learn about available features.
- Features: Access to limited course previews, platform information pages, and promotional materials.

#### **5. Support Staff:**

- Description: Customer support representatives responsible for assisting users with technical issues, account inquiries, and general platform assistance.
- Role: Respond to user inquiries, troubleshoot technical problems, provide guidance on platform usage, and escalate issues as needed.
- Features: Ticketing system for user inquiries, knowledge base for FAQs, live chat support, and communication tools for internal coordination.

Each category of users plays a crucial role in the ecosystem of StudyNotion, contributing to the creation, consumption, and management of educational content on the platform. By addressing the needs and requirements of these diverse user groups, StudyNotion aims to deliver a comprehensive and inclusive learning experience for all stakeholders.

### **3.1 System Requirements: -**

The system requirements for StudyNotion encompass both functional and non-functional aspects to ensure the platform meets the needs of its users while adhering to performance, security, and usability standards.

#### **i. Functional Requirements:**

##### **➤ User Authentication and Authorization:**

- Users should be able to register, login, and manage their accounts securely.
- Differentiate between student, instructor, and administrator roles with appropriate permissions.

##### **➤ Course Management:**

- Instructors should be able to create, edit, publish, and delete courses.
- Students should be able to enrol in courses, access course content, and track their progress.

##### **➤ Content Management:**

- Support various types of multimedia content, including videos, documents, quizzes, and assignments.
- Allow instructors to organize and structure course content effectively.

##### **➤ Communication and Collaboration:**

- Provide communication tools such as messaging, discussion forums, and Q&A sections for interaction between students and instructors.
- Facilitate collaboration between students through group projects, peer reviews, and shared resources.

➤ **E-commerce Integration:**

- Enable students to purchase courses securely using payment gateways.
- Manage transactions, invoices, and payment history for users.

➤ **Search and Discovery:**

- Implement a robust search functionality for users to discover courses based on keywords, categories, or instructor names.
- Offer personalized recommendations based on user preferences, browsing history, and course interactions.

➤ **Analysis and Reporting:**

- Provide analytics dashboards for instructors and administrators to track course performance, user engagement, and revenue.
- Generate reports on key metrics such as enrollment rates, completion rates, and user feedback.

➤ **Administration and Moderation:**

- Admins should have tools to manage user accounts, roles, and permissions.
- Implement content moderation mechanisms to ensure compliance with platform guidelines and policies.

**ii. Non-Functional Requirements:**

➤ **Performance:**

- Ensure fast loading times and responsive user interface across devices and network conditions.
- Handle concurrent user sessions and peak traffic periods efficiently.

➤ **Scalability:**

- Design the system to scale horizontally and vertically to accommodate growth in user base and content volume.
- Support load balancing and auto-scaling mechanisms for optimal resource utilization.

➤ **Security:**

- Implement robust authentication and authorization mechanisms to protect user accounts and sensitive data.
- Encrypt data transmission and storage to prevent unauthorized access or data breaches.

➤ **Reliability:**

- Minimize downtime and system failures through fault-tolerant architecture and proactive monitoring.
- Implement backup and recovery procedures to ensure data integrity and continuity of service.

➤ **Usability:**

- Design an intuitive and user-friendly interface with clear navigation and consistent design patterns.
- Conduct usability testing to gather feedback and refine the user experience based on user preferences and behaviour.

➤ **Accessibility:**

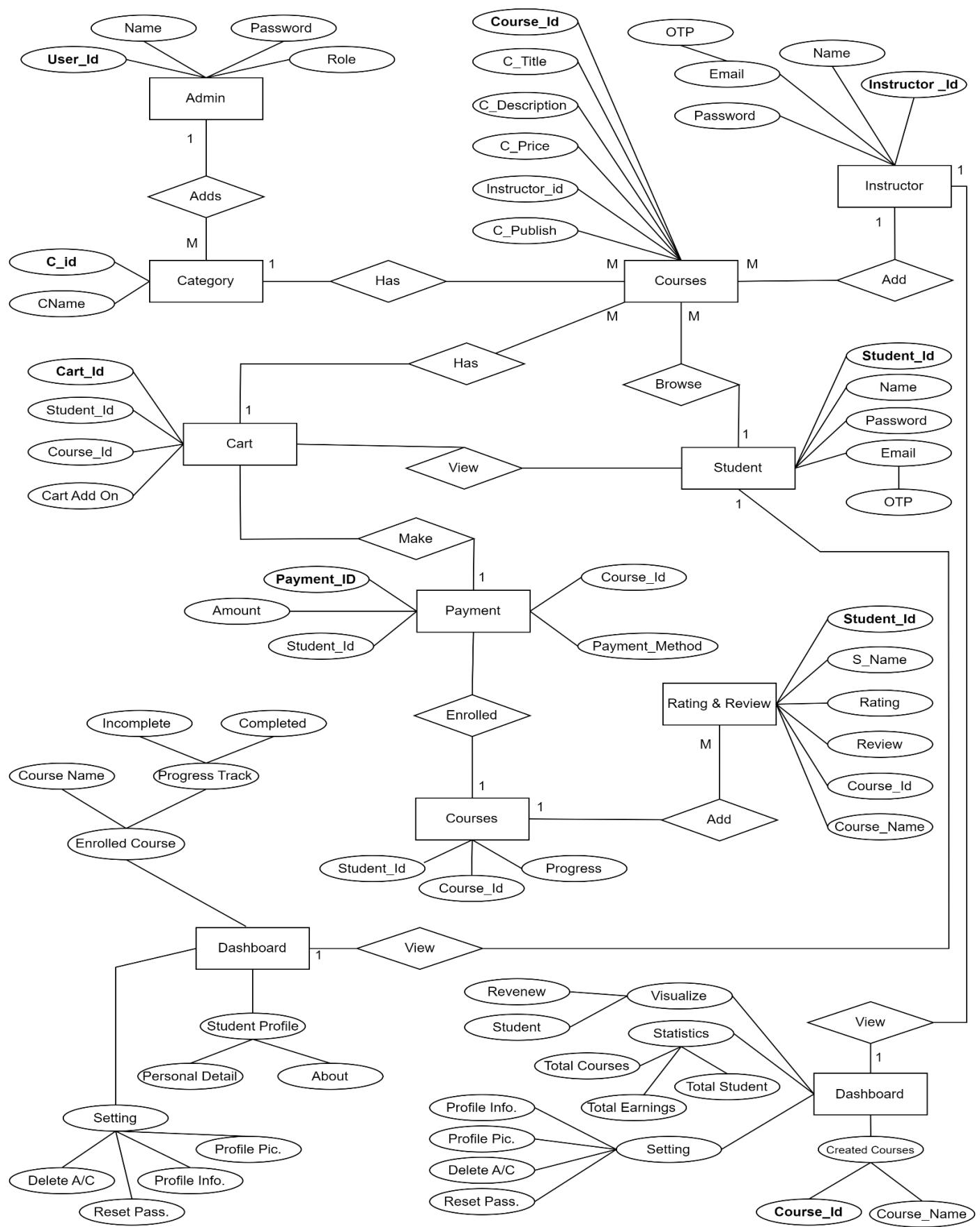
- Ensure compliance with accessibility standards such as WCAG (Web Content Accessibility Guidelines) to accommodate users with disabilities.
- Provide alternative text for images, keyboard navigation options, and other accessibility features

➤ **Compatibility:**

- Support multiple web browsers and devices to ensure compatibility with a diverse user base.
- Test the platform on different operating systems, screen sizes, and resolutions to ensure consistent performance and functionality.

By addressing these functional and non-functional requirements, StudyNotion aims to deliver a robust, secure, and user-friendly ed-tech platform that meets the needs of students, instructors, and administrators while providing an engaging and immersive learning experience.

### 3.2 Entity Relationship Diagram (ERD): -



### **3.3 Table Structure: -**

#### **1. User Table: -**

| <b>Sr. No</b> | <b>Field Name</b>  | <b>Data Type (width)</b> | <b>Description</b>                                   | <b>Occurrences</b> |
|---------------|--------------------|--------------------------|--|--------------------|
| 1             | ID                 | String                   | Unique identifier for each user                      | 10                 |
| 2             | First Name         | String                   | First name of the user                               | 10                 |
| 3             | Last Name          | String                   | Last name of the user                                | 10                 |
| 4             | Email              | String                   | Email address of the user                            | 10                 |
| 5             | Password           | String                   | Encrypted password of the user                       | 10                 |
| 6             | Account Type       | String                   | Type of account (Admin, Student, Instructor)         | 10                 |
| 7             | Active             | Boolean                  | Indicates if the account is active                   | 10                 |
| 8             | Approved           | Boolean                  | Indicates if the account is approved                 | 10                 |
| 9             | Additional Details | String                   | Additional details about the user                    | 10                 |
| 10            | Courses            | Array of Strings         | Courses associated with the user                     | 10                 |
| 11            | Image              | String                   | URL of the user's image                              | 8                  |
| 12            | Course Progress    | Array of Strings         | Progress of the user in various courses              | 8                  |
| 13            | Created At         | Date-Time                | Date and time when the user account was created      | 10                 |
| 14            | Updated At         | Date-Time                | Date and time when the user account was last updated | 10                 |
| 15            | __v                | Integer                  | Version of the document (used by MongoDB)            | 10                 |

## **2. Course Category Table: -**

| Sr. No | Field Name  | Data Type (width) | Description          | Occurrences |
|--------|-------------|-------------------|----------------------|-------------|
| 1      | _Id         | String            | Unique identifier    | 1           |
| 2      | Name        | String            | Name of category     | 1           |
| 3      | Description | String            | Category description | 1           |
| 4      | Courses     | Array             | Array of courses     | 1           |
| 5      | __v         | Number            | Version control      | 0           |

## **3. Course Table: -**

| <b>Sr. No</b> | <b>Field Name</b>   | <b>Data Type (width)</b> | <b>Description</b>               | <b>Occurrences</b> |
|---------------|---------------------|--------------------------|----------------------------------|--------------------|
| 1             | Course ID           | String                   | Unique identifier for the course | 1                  |
| 2             | Course Name         | String                   | Name of the course               | 1                  |
| 3             | Course Description  | String                   | Description of the course        | 1                  |
| 4             | Instructor          | String                   | Instructor of the course         | 1                  |
| 5             | What You Will Learn | String                   | Learning objectives              | 1                  |
| 6             | Course Content      | Array                    | Content of the course            | 1                  |
| 7             | Rating and Reviews  | Array                    | Reviews and ratings              | 2                  |
| 8             | Price               | Number                   | Price of the course              | 1                  |
| 9             | Thumbnail           | String (URL)             | Thumbnail image of the course    | 1                  |
| 10            | Tag                 | Array                    | Tags associated with the course  | 1                  |
| 11            | Category            | String                   | Category of the course           | 1                  |
| 12            | Students Enrolled   | Array                    | List of enrolled students        | 3                  |
| 13            | Instructions        | Array                    | Course instructions              | 1                  |
| 14            | Status              | String                   | Status of the course             | 1                  |
| 15            | Created At          | Date/time                | Date and time of creation        | 1                  |
| 16            | Updated At          | Date/time                | Date and time of last update     | 1                  |
| 17            | _v                  | Number                   | Version control                  | 1                  |

#### **4. Section Table: -**

| <b>Sr. No</b> | <b>Field Name</b> | <b>Data Type (width)</b> | <b>Description</b>                | <b>Occurrences</b> |
|---------------|-------------------|--------------------------|-----------------------------------|--------------------|
| 1             | _Id               | String                   | Unique identifier for the section | 1                  |
| 2             | Section Name      | String                   | Name of the section               | 1                  |
| 3             | Subsection        | Array                    | Array of subsections              | 1                  |
| 4             | __v               | Number                   | Version control                   | 1                  |

#### **5. Subsection Table: -**

| <b>Sr. No</b> | <b>Field Name</b> | <b>Data Type (width)</b> | <b>Description</b>                   | <b>Occurrences</b> |
|---------------|-------------------|--------------------------|--------------------------------------|--------------------|
| 1             | _Id               | String                   | Unique identifier for the subsection | 1                  |
| 2             | Title             | String                   | Title of the subsection              | 1                  |
| 3             | Time Duration     | String                   | Duration of the video                | 1                  |
| 4             | Description       | String                   | Description of the subsection        | 1                  |
| 5             | Video URL         | String                   | URL of the video                     | 1                  |
| 6             | __v               | Number                   | Version control                      | 1                  |

## **6. Profile Table: -**

| <b>Sr.<br/>No</b> | <b>Field Name</b> | <b>Data Type<br/>(width)</b> | <b>Description</b>           | <b>Occurrences</b> |
|-------------------|-------------------|------------------------------|------------------------------|--------------------|
| 1                 | _Id               | String                       | Unique identifier            | 1                  |
| 2                 | Gender            | String                       | Gender of the person         | 1                  |
| 3                 | Date of Birth     | Date                         | Date of birth of the person  | 1                  |
| 4                 | About             | String                       | Description about the person | 1                  |
| 5                 | Contact Number    | String                       | Contact number of the person | 1                  |
| 6                 | __v               | Number                       | Version control              | 1                  |

## **7. Rating and Review Table: -**

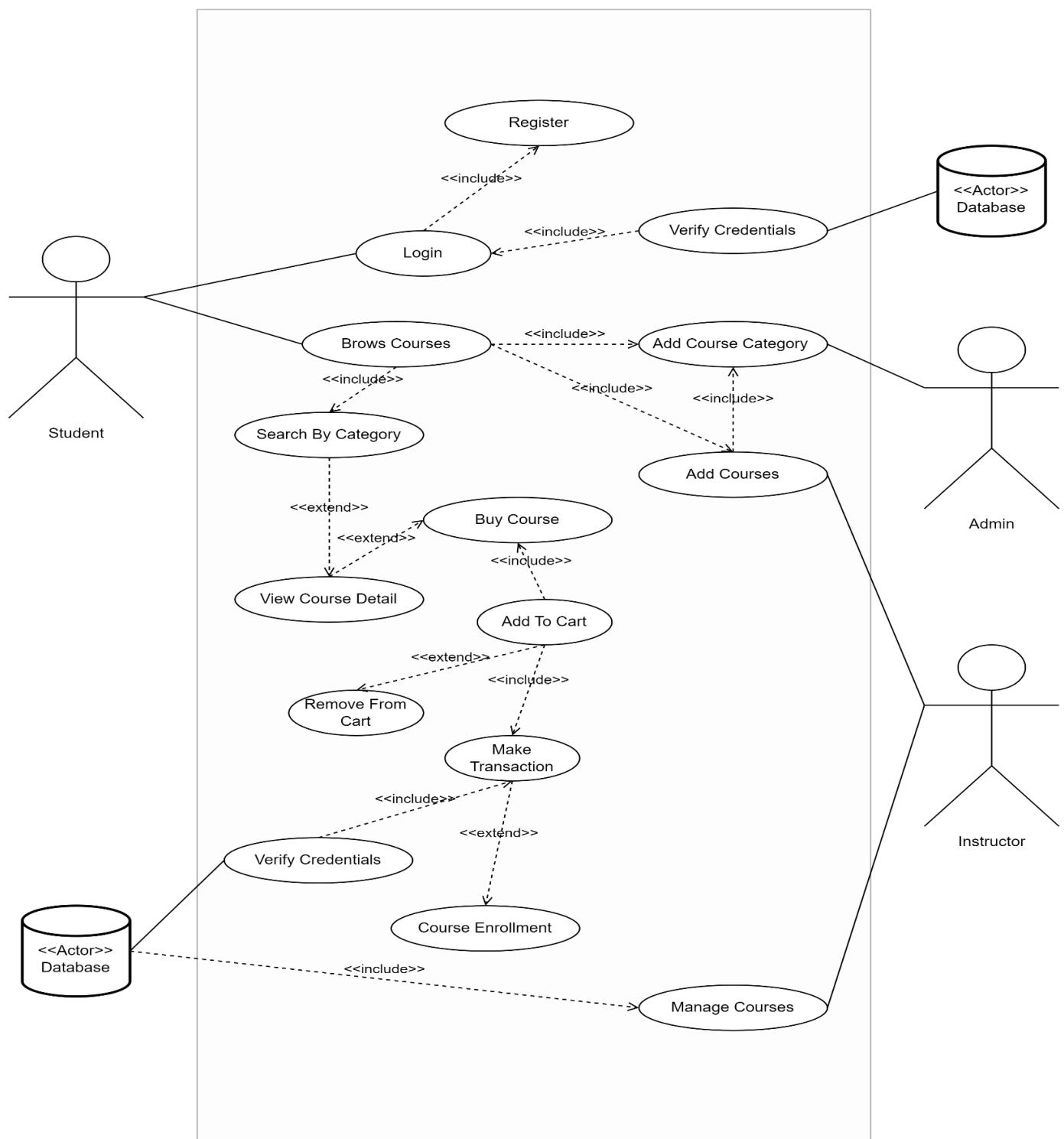
| <b>Sr.<br/>No</b> | <b>Field<br/>Name</b> | <b>Data Type<br/>(width)</b> | <b>Description</b>              | <b>Occurrences</b> |
|-------------------|-----------------------|------------------------------|---------------------------------|--------------------|
| 1                 | _Id                   | String                       | Unique identifier               | 1                  |
| 2                 | User                  | String                       | Unique identifier of the user   | 1                  |
| 3                 | Rating                | Number                       | Rating given by the user        | 1                  |
| 4                 | Review                | String                       | Review given by the user        | 1                  |
| 5                 | Course                | String                       | Unique identifier of the course | 1                  |
| 6                 | __V                   | Number                       | Version control                 | 0                  |

## **8. Rating and Review Table: -**

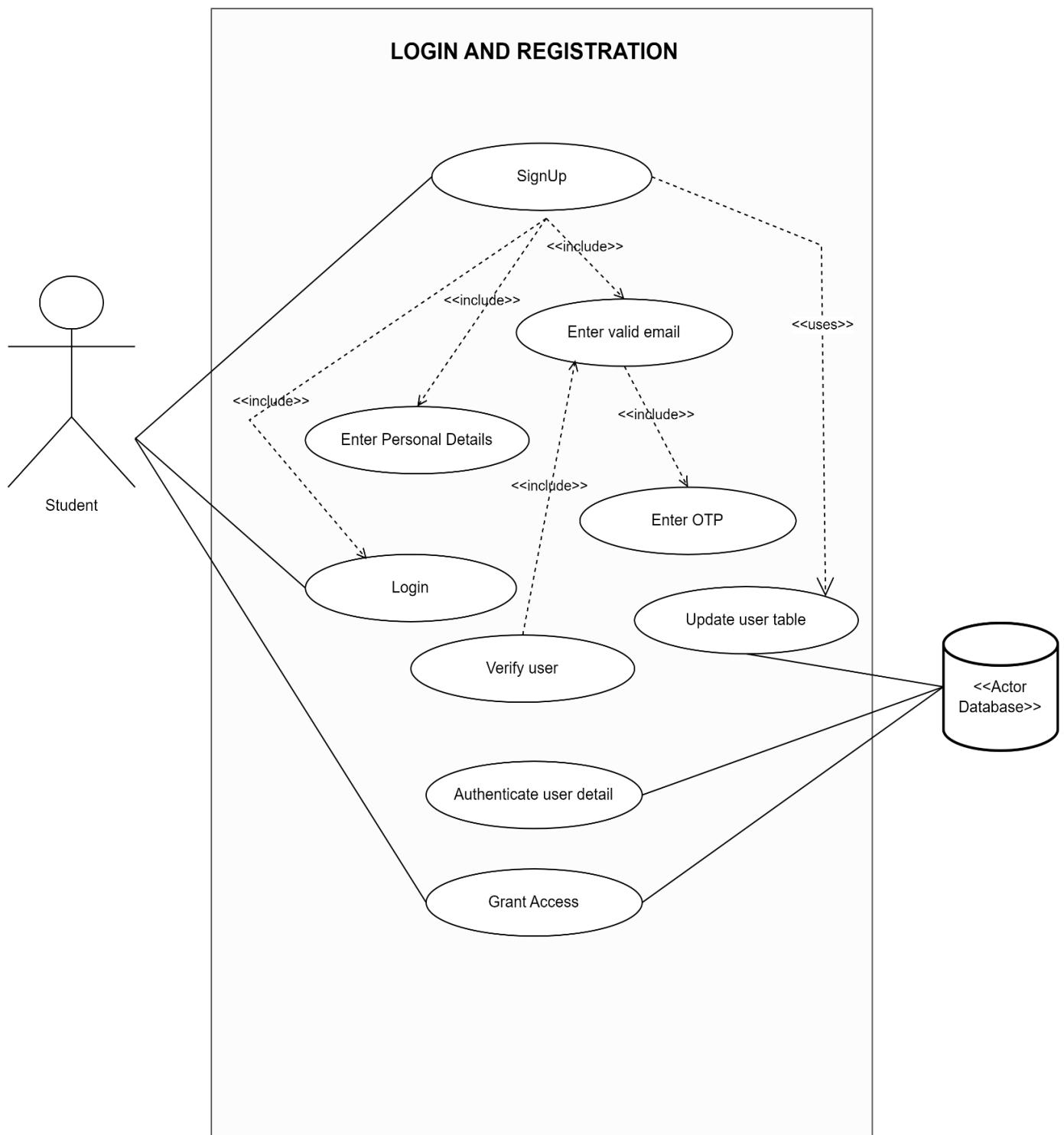
| <b>Sr.<br/>No</b> | <b>Field Name</b> | <b>Data Type<br/>(width)</b> | <b>Description</b>          | <b>Occurrences</b> |
|-------------------|-------------------|------------------------------|-----------------------------|--------------------|
| 1                 | _Id               | String                       | Unique identifier           | 1                  |
| 2                 | CourseID          | String                       | Unique identifier of course | 1                  |
| 3                 | UserID            | String                       | Unique identifier of user   | 1                  |
| 4                 | CompletedVideos   | Array                        | Array of completed videos   | 0                  |
| 5                 | _V                | Number                       | Version control             | 0                  |

### 3.4 Business Use Case Diagram: -

#### 3.4.1 Functional Use Case Diagram



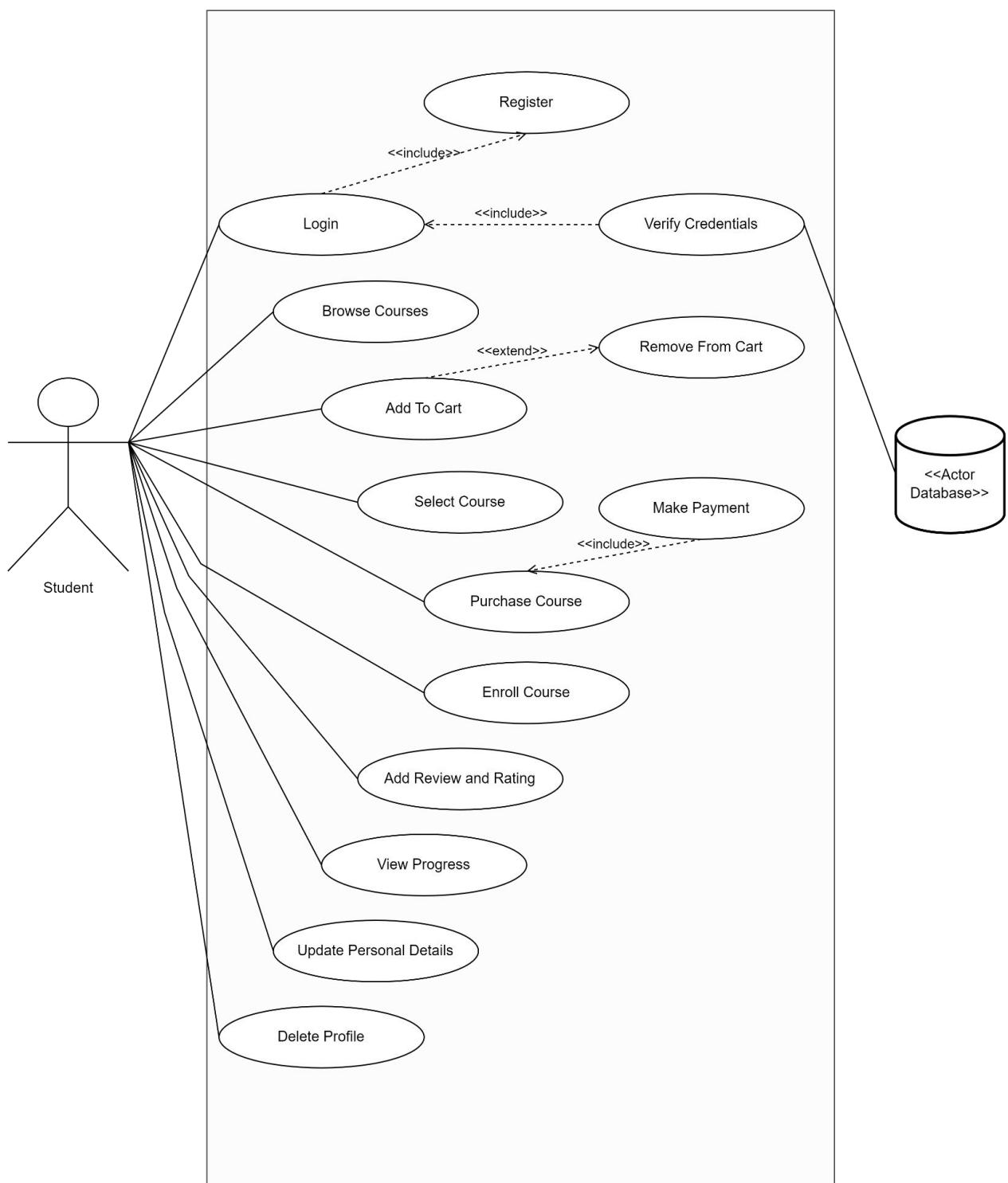
### 3.4.2 Login and Registration Use Case Diagram-



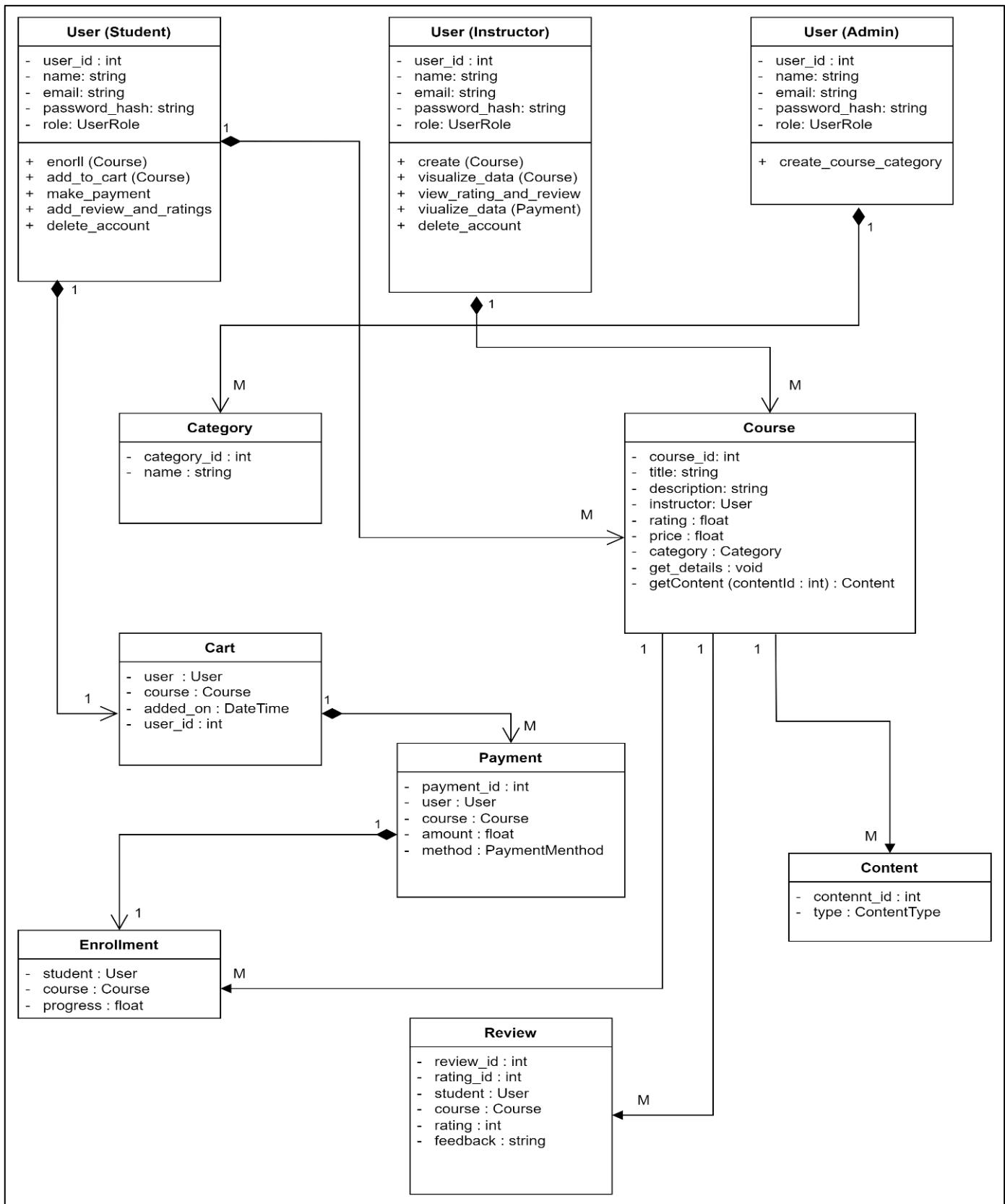
### 3.4.3 Instructor Use Case Diagram:



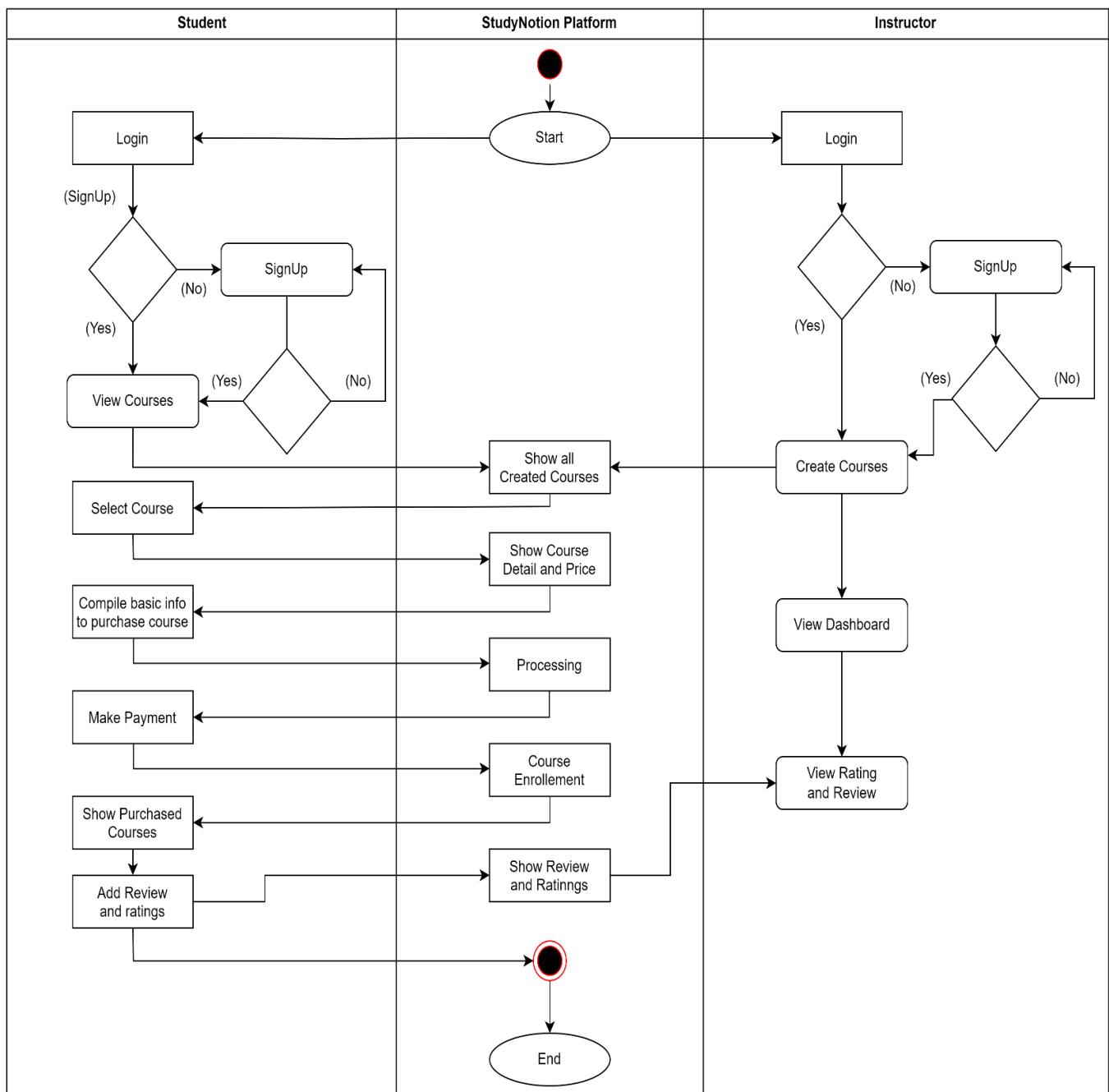
### 3.4.4 Student Use Case Diagram-



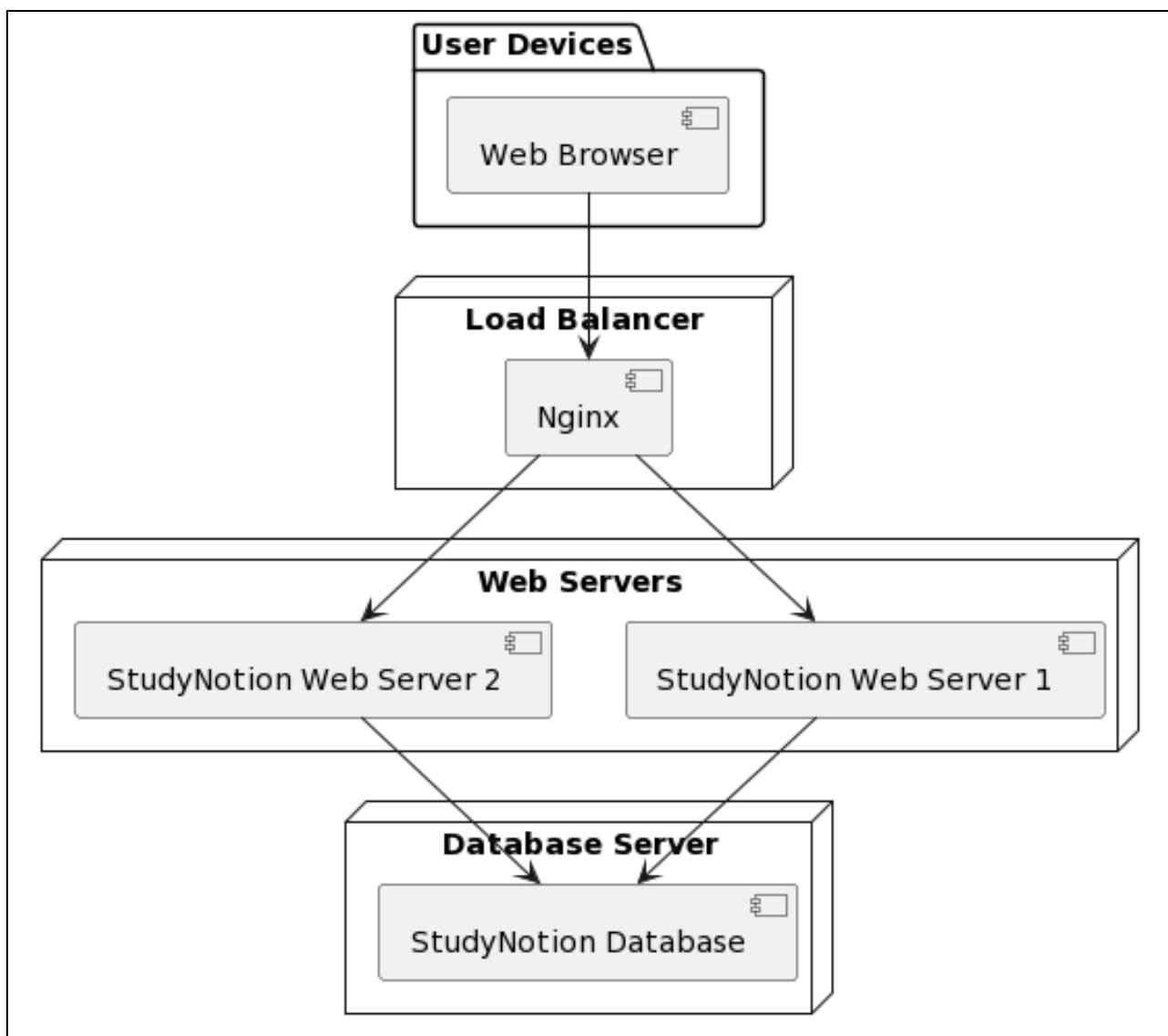
### 3.5: Class Diagram



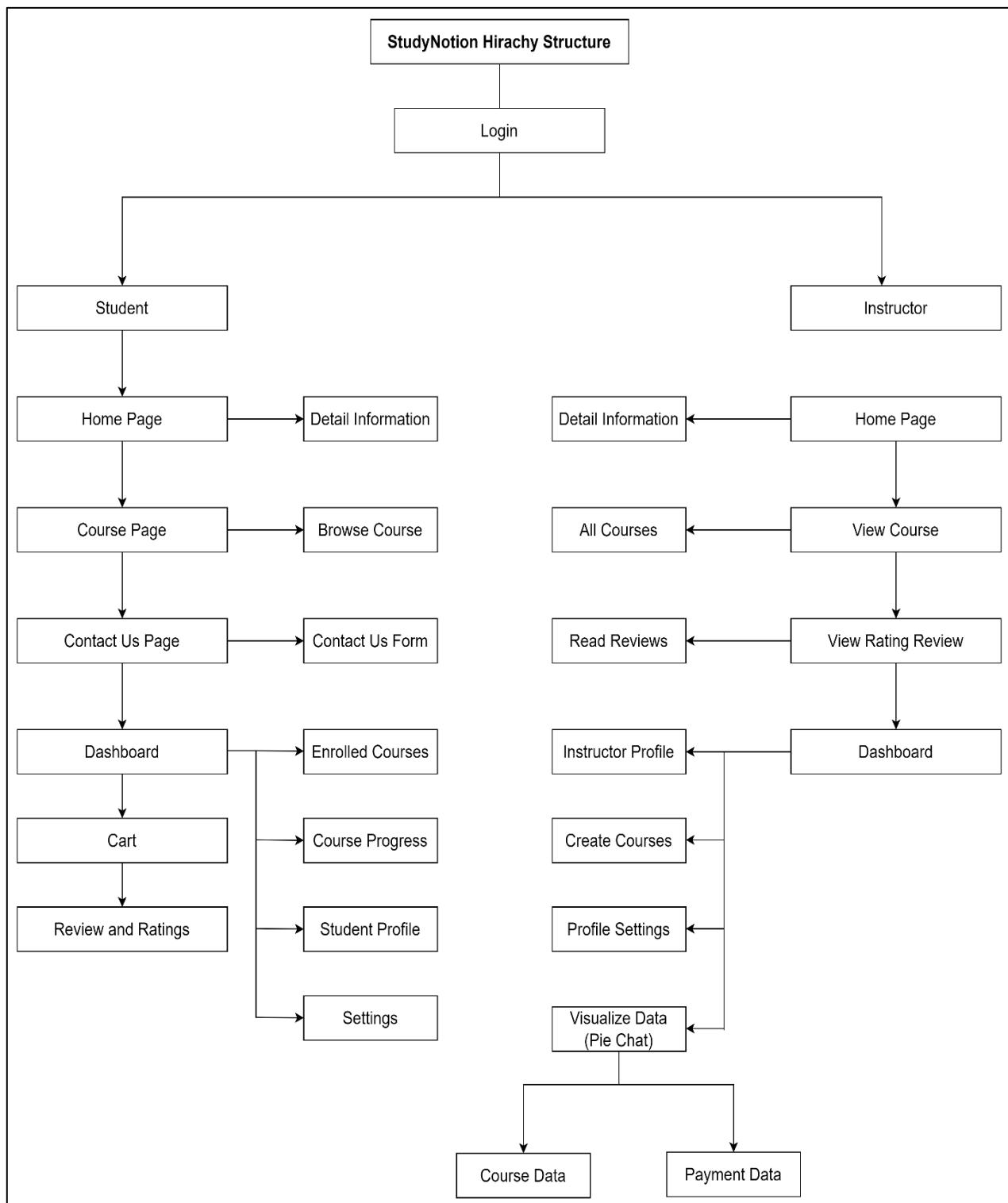
### 3.6: Activity Diagram



### 3.7: Deployment Diagram:



### 3.8: Module Hierarchy Diagram:



### **3.9 Sample Input and Output Screens:**

StudyNotion offers a user-friendly interface designed to enhance your learning and teaching experiences. Here's a brief overview of our interface:

## 1. Home Page:

The screenshot shows the homepage of StudyNotion, a platform for learning coding skills. At the top, there's a navigation bar with links for Home, Catalog, About Us, Contact Us, Search, Login, and Signup. A button to 'Become an Instructor' is also present. The main headline reads 'Empower Your Future with Coding Skills'. Below it, a sub-headline states: 'With our online coding courses, you can learn at your own pace, from anywhere in the world, and get access to a wealth of resources, including hands-on projects, quizzes, and personalized feedback from instructors.' Three buttons are available: 'Learn More', 'Book a Demo', and 'Watch Live'. A large image shows a person in a hoodie sitting at a desk, looking at a laptop screen which displays another video feed. Below this, a section titled 'Unlock your coding potential with our online courses.' features a snippet of HTML code. Another snippet of React code is shown further down. To the right, a section titled 'Start coding in seconds' encourages users to try it themselves. At the bottom, a section titled 'Unlock the Power of Code' highlights three free courses: 'Learn HTML', 'Learn CSS', and 'Responsive Web design', each with a brief description and a 'Beginner' level indicator.

StudyNotion

Home Catalog About Us Contact Us Search Login Signup

Become an Instructor →

### Empower Your Future with Coding Skills

With our online coding courses, you can learn at your own pace, from anywhere in the world, and get access to a wealth of resources, including hands-on projects, quizzes, and personalized feedback from instructors.

Learn More Book a Demo Watch Live



Alt+A

#### Unlock your coding potential with our online courses.

Our courses are designed and taught by industry experts who have years of experience in coding and are passionate about sharing their knowledge with you.

Try It Yourself → Learn More

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>[REDACTED]
5   </head>
6   <body>
7     <h1>Hello World</h1>
8     <p>This is my first web page!</p>
9   </body>
10 </html>
```

```
1 import React from "react";
2 import CTAButton from "./Button";
3 import TypeAnimation
```

#### Start coding in seconds

Go ahead, give it a try. Our hands-on learning environment means you'll be writing real code from your very first lesson.

Continue Lesson → Learn More

### Unlock the Power of Code

Learn to Build Anything You Can Imagine

Free New to coding Most popular Skills paths Career paths

**Learn HTML**  
This course covers the basic concepts of HTML including creating and structuring web pages, adding text, links, images, and more.

Beginner 6 Lesson

**Learn CSS**  
This course explores advanced topics in HTML5 and CSS3, including animations, transitions, and layout techniques.

Beginner 6 Lesson

**Responsive Web design**  
This course teaches responsive web design techniques, allowing web pages to adapt to different devices and screen sizes.

Beginner 6 Lesson

## 2. Courses Page:

S StudyNotion

Home Catalog About Us Contact Us Log In Sign Up

Home / Catalog / Web Development

### Web Development

Master the art of web development! Learn how to build engaging, user-friendly websites and applications using a variety of technologies and frameworks. Dive into HTML, CSS, JavaScript, and more. Whether you're a beginner or seasoned developer, this course empowers you to create innovative web solutions from scratch.

#### Courses to get you started

Most Popular New



**COMPLETE TAILWIND CSS FOR BEGINNERS IN ONE SHOT**  
Tailwind CSS Tutorial  
Hitesh Choudhary  
5 ★★★★★ 1 Ratings  
Rs. 999



**WANT TO BECOME JAVA DEVELOPER**  
Java Backend Development!  
Ravi Sastri  
5 ★★★★★ 1 Ratings  
Rs. 5999



**FULLY STACKED 30 DAYS!**  
Master the MERN Stack!  
Rohit Sharma  
0 ★☆☆☆☆ 0 Ratings  
Rs. 7999

#### Top Courses in Web3



**LEARN BLOCKCHAIN 12-HOUR COURSE**  
Blockchain Full-Stack Development!  
Rohit Sharma  
0 ★☆☆☆☆ 2 Ratings  
Rs. 14999



**SOLIDITY IN 4 HOURS**  
Master Solidity Smart Contract Development!  
Ravi Sastri  
0 ★☆☆☆☆ 1 Ratings  
Rs. 11999

#### Frequently Bought



**COMPLETE TAILWIND CSS FOR BEGINNERS IN ONE SHOT**  
Tailwind CSS Tutorial  
Hitesh Choudhary  
0 ★☆☆☆☆ 2 Ratings  
Rs. 999



**WANT TO BECOME JAVA DEVELOPER**  
Java Backend Development!  
Ravi Sastri  
0 ★☆☆☆☆ 1 Ratings  
Rs. 5999



**FULLY STACKED 30 DAYS!**  
Master the MERN Stack!  
Rohit Sharma  
0 ★☆☆☆☆ 0 Ratings  
Rs. 7999



**FULLY STACKED 30 DAYS!**  
Master the MERN Stack!  
Ashish Nehra  
0 ★☆☆☆☆ 0 Ratings  
Rs. 7999



**FULL KOTLIN COURSE COVERS EVERYTHING**  
Kotlin - Complete Course  
Rahul Davvid  
0 ★☆☆☆☆ 3 Ratings  
Rs. 1999



**ANDROID COURSE ZERO TO HERO**  
Build Android Apps with Java!  
Ravi Sastri  
0 ★☆☆☆☆ 1 Ratings  
Rs. 9999

#### Reviews from other learners

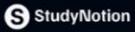
Rohit Sharma Master Data Science Fundamentals  
Great course with lots of projects.  
5 ★★★★★

Virat Kohli Build Android Apps with Java!  
Java is industry language and this course is best for android development in java.  
5 ★★★★★

Gautam Gambhir Master Data Engineering Fundamentals  
Data Engineer is best for who don't know too much coding.  
5 ★★★★★

Rishabh Pant DevOps Fundamentals  
After this course you will fall in love with command line.  
5 ★★★★★

### 3. About Us:

 StudyNotion

Home Catalog ▾ About Us Contact Us Search 

Login Signup

## Driving Innovation in Online Education for a Brighter Future

StudyNotion is at the forefront of driving innovation in online education. We're passionate about creating a brighter future by offering cutting-edge courses, leveraging emerging technologies, and nurturing a vibrant learning community.



We are passionate about revolutionizing the way we learn. Our innovative platform **combines technology expertise**, and community to create an **unparalleled educational experience**.

### Our Founding Story

Our e-learning platform was born out of a shared vision and passion for transforming education. It all began with a group of educators, technologists, and lifelong learners who recognized the need for accessible, flexible, and high-quality learning opportunities in a rapidly evolving digital world.

As experienced educators ourselves, we witnessed firsthand the limitations and challenges of traditional education systems. We believed that education should not be confined to the walls of a classroom or restricted by geographical boundaries. We envisioned a platform that could bridge these gaps and empower individuals from all walks of life to unlock their full potential.



Alt+A 

### Our Vision

With this vision in mind, we set out on a journey to create an e-learning platform that would revolutionize the way people learn. Our team of dedicated experts worked tirelessly to develop a robust and intuitive platform that combines cutting-edge technology with engaging content, fostering a dynamic and interactive learning experience.

### Our Mission

Our mission goes beyond just delivering courses online. We wanted to create a vibrant community of learners, where individuals can connect, collaborate, and learn from one another. We believe that knowledge thrives in an environment of sharing and dialogue, and we foster this spirit of collaboration through forums, live sessions, and networking opportunities.

|                               |                       |                        |                      |
|-------------------------------|-----------------------|------------------------|----------------------|
| <b>1k+</b><br>Active Students | <b>10+</b><br>Mentors | <b>200+</b><br>Courses | <b>50+</b><br>Awards |
|-------------------------------|-----------------------|------------------------|----------------------|

## 4. Contact Us Page

Chat on us  
Our friendly team is here to help.  
vishalgunjal0287@gmail.com

Visit us  
Come and say hello at our office HQ.  
Mayur Colony, Kothrud, Pune-411038

Call us  
Mon - Fri From 8am to 5pm  
+91 9022717020

Got a Idea? We've got the skills.  
Let's team up

Tell us more about yourself and what you're got in mind.

First Name

Last Name

Email Address

Phone Number

Message

**Send Message**

Go Through **Reviews** Of Previous Students To Make An Informed Decision

sandip pandhare  
Complete JAVA Language  
Best Course!!!  
★★★★★

sandip pandhare  
Complete MERN  
Good!!!  
★★★★★

sandip pandhare  
MongoDB In 2 Hours  
Good!!!  
★★★★★

**S** StudyNotion

**Company**  
About  
Careers  
Affiliates

**Support**  
Help Center

**Resources**  
Articles  
Blog  
Chart Sheet  
Code challenges  
Docs  
Projects  
Videos  
Workspaces

**Plans**  
Paid memberships  
For students  
Business solutions

**Community**  
Forums  
Chapters  
Events

**Subjects**  
AI  
Cloud Computing  
Code Foundations  
Computer Science  
Cybersecurity  
Data Analytics  
Data Science  
Data Visualization  
Developer Tools  
DevOps  
Game Development  
IT  
Machine Learning  
Math  
Mobile Development  
Web Design  
Web Development

**Languages**  
Bash  
C++  
C#  
Go  
HTML & CSS  
Java  
JavaScript  
Kotlin  
PHP  
Python  
R  
Ruby  
SQL  
Swift

**Career building**  
Career paths  
Career services  
Interview prep  
Professional certification  
-  
Full Catalog  
Beta Content

## 9. Signup Page:

**StudyNotion**

Home Catalog About Us Contact Us Search

Login Signup

**Join the millions learning to code with StudyNotion for free**

Build skills for today, tomorrow, and beyond.  
*Education to future-proof your career.*

**Student** **Instructor**

First Name \* Last Name \*

Enter first name Enter last name

Email Address \*

Enter email address

Create Password \* Confirm Password \*

Enter Password Confirm Password

**Create Account**



**StudyNotion**

| Resources  | Plans              | Subjects           | Languages  | Career building            |
|------------|--------------------|--------------------|------------|----------------------------|
| Articles   | Paid memberships   | AI                 | Bash       | Career paths               |
| About      | For students       | Cloud Computing    | C++        | Career services            |
| Careers    | Business solutions | Code Foundations   | C#         | Interview prep             |
| Affiliates | Code challenges    | Computer Science   | Go         | Professional certification |
| Docs       | Forums             | Cybersecurity      | HTML & CSS | -                          |
| Projects   | Chapters           | Data Analytics     | Java       | Full Catalog               |
| Videos     | Events             | Data Science       | JavaScript | Beta Content               |
| Workspaces |                    | Data Visualization | Kotlin     |                            |
|            |                    | Developer Tools    | PHP        |                            |
|            |                    | DevOps             | Python     |                            |
|            |                    | Game Development   | R          |                            |
|            |                    | IT                 | Ruby       |                            |
|            |                    | Machine Learning   | SQL        |                            |
|            |                    | Math               | Swift      |                            |
|            |                    | Mobile Development |            |                            |
|            |                    | Web Design         |            |                            |
|            |                    | Web Development    |            |                            |

Community

Support

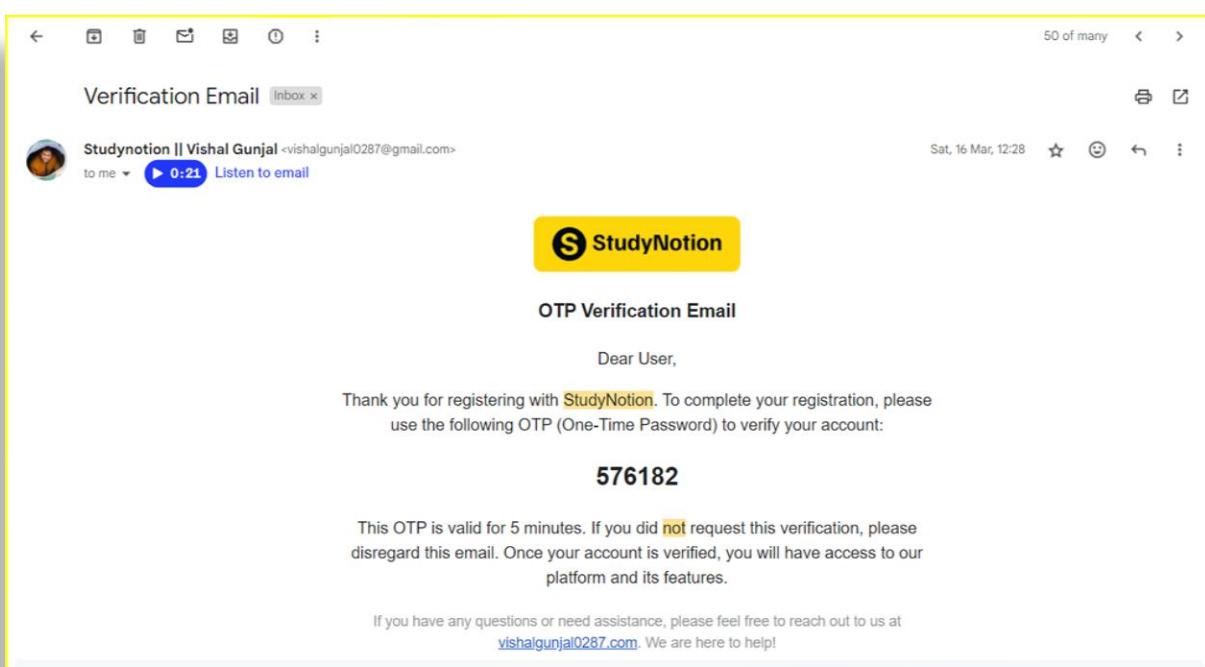
Help Center

Privacy Policy Cookie Policy Terms

Developed by Vishal D. Gunjal, © 2024 StudyNotion - Empowering Education  
A proficient S.Y MCA Student at MES's Institute of Management and Career Courses, Pune.

Alt+A

## 10. OTP Verification on E-Mail:



## 11. Password Update:

The screenshot shows a dark-themed web page for StudyNotion. At the top, there is a navigation bar with links for Home, Catalog, About Us, Contact Us, Log in, and Sign up. The main content area has a heading "Choose new password" and a sub-instruction "Almost done. Enter your new password and you're all set.". Below this are two input fields: "New Password" and "Confirm New Password", both with placeholder text "Enter Password" and "Confirm Password" respectively. A yellow "Reset Password" button is centered below the inputs. At the bottom left, there is a link "← Back To Login" and a keyboard shortcut "Alt+A".

## 12. Login Page:

The screenshot shows the StudyNotion login page. At the top, there is a navigation bar with links for Home, Catalog, About Us, Contact Us, a search bar, and buttons for Login and Signup. A yellow banner at the top left says "Welcome Back" and "Build skills for today, tomorrow, and tomorrow's tomorrow. Education to future-proof your career". It features two buttons: "Click here for Instructor Demo" and "Click here for Student Demo". Below the banner is a form for logging in, with fields for Email Address and Password, and a "Forgot Password" link. To the right of the form is a photo of a smiling woman sitting at a desk with a tablet. The main content area below the login form includes a sidebar with links for Company (About, Careers, Affiliates), Resources (Articles, Blog, Chart Sheet, Code challenges, Docs, Projects, Videos, Workspaces), and Support (Help Center). There is also a "Community" section with links for Forums, Chapters, and Events. The main content area lists Subjects (AI, Cloud Computing, Code Foundations, Computer Science, Cybersecurity, Data Analytics, Data Science, Data Visualization, Developer Tools, DevOps, Game Development, IT, Machine Learning, Math, Mobile Development, Web Design, Web Development) and Languages (Bash, C++, C#, Go, HTML & CSS, Java, JavaScript, Kotlin, PHP, Python, R, Ruby, SQL, Swift). A "Career building" section includes links for Career paths, Career services, Interview prep, Professional certification, Full Catalog, and Beta Content. At the bottom, there are links for Privacy Policy, Cookie Policy, Terms, and a note about the developer: "Developed by Vishal D. Gunjal, © 2024 StudyNotion - Empowering Education. A proficient 5.Y MCA Student at MES's Institute of Management and Career Courses, Pune."

## 13. Student Profile:

The screenshot shows the 'My Profile' section of the StudyNotion website. On the left, a sidebar menu includes 'My Profile' (selected), 'Enrolled Courses', 'Cart', 'Settings', and 'Logout'. The main content area displays a profile card for 'sandip pandhare' (demouser@gmail.com) with an 'Edit' button. Below it is an 'About' section with a placeholder 'Write Something about Yourself' and an 'Edit' button. The 'Personal Details' section contains fields for First Name ('sandip'), Last Name ('pandhare'), Email ('demouser@gmail.com'), Phone Number ('9022717020'), Gender ('Male'), and Date of Birth ('2002-05-09'), each with an 'Edit' button.

**StudyNotion**

**Company**

- About
- Careers
- Affiliates

**Resources**

- Articles
- Blog
- Chart Sheet
- Code challenges
- Docs
- Projects
- Videos
- Workspaces

**Community**

- Forums
- Chapters
- Events

**Support**

- Help Center

**Subjects**

- AI
- Cloud Computing
- Code Foundations
- Computer Science
- Cybersecurity
- Data Analytics
- Data Science
- Data Visualization
- Developer Tools
- Game Development
- IT
- Machine Learning
- Math
- Mobile Development
- Web Design
- Web Development

**Languages**

- Bash
- C++
- C#
- Go
- HTML & CSS
- Java
- JavaScript
- Kotlin
- PHP
- R
- Ruby
- SQL
- Swift

**Career building**

- Career paths
- Career services
- Interview prep
- Professional certification
- 
- Full Catalog
- Beta Content

Privacy Policy | Cookie Policy | Terms

Developed by Vishal D. Gunjal, © 2024 StudyNotion - Empowering Education  
A proficient S.Y MCA Student at MES's Institute of Management and Career Courses, Pune.

## 14. Enrolled Courses Page

**StudyNotion**

Home Catalog ▾ About Us Contact Us Search 

**Enrolled Courses**

| Course Name   | Progress         |
|---|------------------|
|  <b>DevOps Basics</b><br>Dev Ops Basics Description.   | Completed: 1 / 1 |
|  <b>Complete JAVA Language</b><br>Unlock the full potential of Java with our compreh...                  | Completed: 1 / 1 |
|  <b>MongoDB in 2 Hours</b><br>Master the MERN (MongoDB, Express.js, React.js, No....                     | Completed: 1 / 1 |
|  <b>MERN BOOTCAMP</b><br>MERN Bootcamp: Dive into full-stack web developmen...                           | Completed: 1 / 1 |
|  <b>Complete MERN</b><br>MERN Stack In 2 hours<br>MERN is a powerful tech stack for building web app.... | Completed: 2 / 2 |
|  <b>Data Science End to End</b><br>Data Science from start to finish. Dive into ev...                   | Completed: 0 / 2 |
|  <b>Basics of Cloud Computing</b><br>Introduction to essential cloud computing concepts....            | Completed: 0 / 2 |
|  <b>Cloud Fundamentals</b><br>Introduction to essential cloud computing concepts....                   | Completed: 0 / 2 |

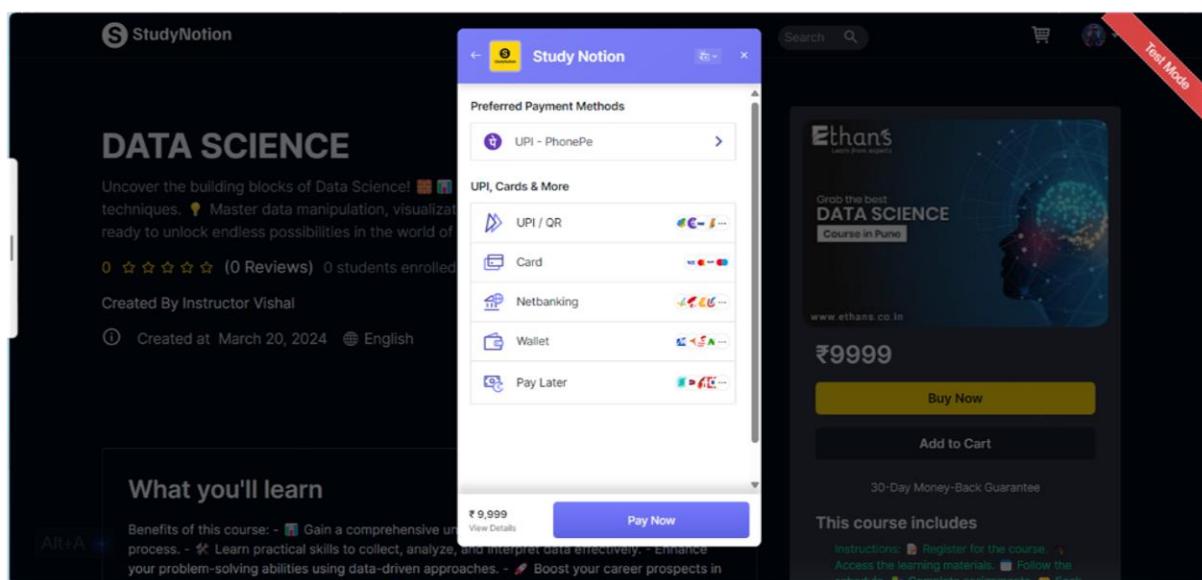
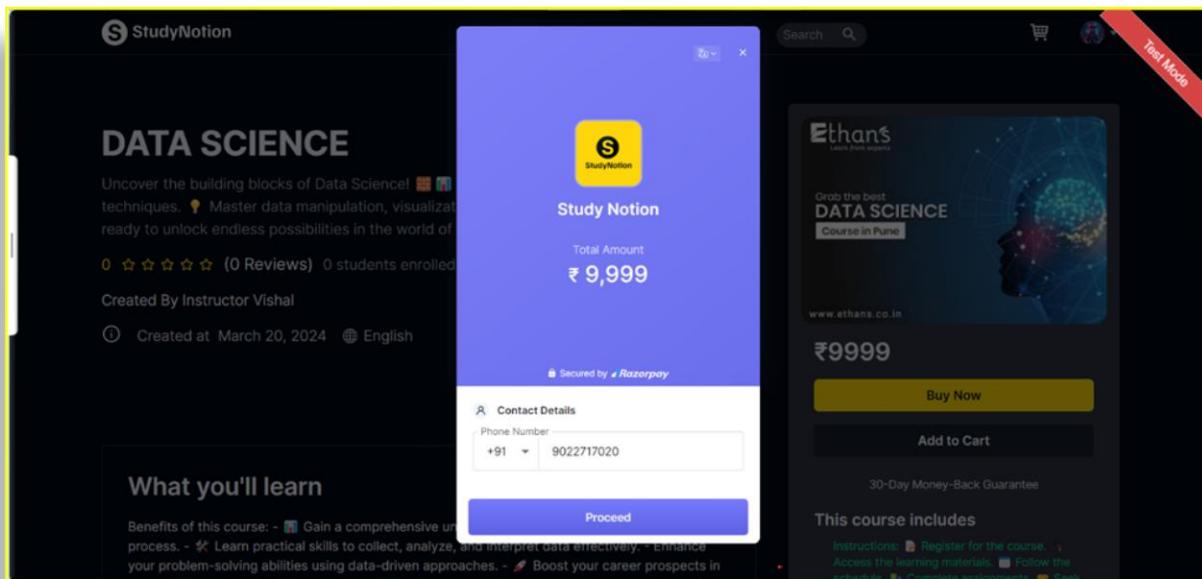
## 15. Student Cart:

The screenshot shows the StudyNotion website's cart page. The cart contains two courses:

- DATA SCIENCE** by Ethan's Data Science, priced at ₹ 9999. It has a yellow "Buy Now" button.
- Java Programming** by Code Java, priced at ₹ 4999. It also has a yellow "Buy Now" button.

The total amount shown in the cart summary is ₹ 14998.

## 16. Payment Creation:



The screenshot shows a Data Science course page on StudyNotion. The course title is "DATA SCIENCE". It includes a QR code for UPI payment, a form for UPI ID/Mobile Number, and a payment summary showing ₹9,999. To the right, there's a promotional box for Ethan's Data Science course at ₹9999.

**DATA SCIENCE**

Uncover the building blocks of Data Science! Master data manipulation, visualization, and machine learning techniques. Ready to unlock endless possibilities in the world of Data Science?

0 ★★★★★ (0 Reviews) 0 students enrolled

Created By Instructor Vishal

Created at March 20, 2024 English

**What you'll learn**

Benefits of this course: - Gain a comprehensive understanding of data science concepts. - Learn practical skills to collect, analyze, and interpret data effectively. - Enhance your problem-solving abilities using data-driven approaches. - Boost your career prospects in Data Science.

Pay With UPI QR

Scan the QR using any UPI app on your phone.  
QR Code is valid for 2:49 minutes

UPI ID/ Mobile Number  
Enter UPI ID/ Mobile Number  
9022717020@ybl

₹ 9,999 View Details Pay Now

**Ethan's** Learn from experts

Grab the best DATA SCIENCE Course in Pune

www.ethans.co.in ₹9999

Buy Now Add to Cart

30-Day Money-Back Guarantee

This course includes

Instructions: Register for the course. Access the learning materials. Follow the schedule. Complete assessments. Seek help.

The screenshot shows a successful payment confirmation from Study Notion. It displays a green success message with a checkmark icon, the amount ₹9,999, and the payment method UPI | pay\_NyNkdQccXuVS3x. Below this, there's a redirection message.

**DATA SCIENCE**

Uncover the building blocks of Data Science! Master data manipulation, visualization, and machine learning techniques. Ready to unlock endless possibilities in the world of Data Science?

0 ★★★★★ (0 Reviews) 0 students enrolled

Created By Instructor Vishal

Created at March 20, 2024 English

**What you'll learn**

Benefits of this course: - Gain a comprehensive understanding of data science concepts. - Learn practical skills to collect, analyze, and interpret data effectively. - Enhance your problem-solving abilities using data-driven approaches. - Boost your career prospects in Data Science.

Payment successful

₹ 9,999

Study Notion Apr 14, 2024 | 10:40 AM

UPI | pay\_NyNkdQccXuVS3x ⓘ  
Visit razorpay.com/support for queries

Redirecting in 3 seconds...

**Ethan's** Learn from experts

Grab the best DATA SCIENCE Course in Pune

www.ethans.co.in ₹9999

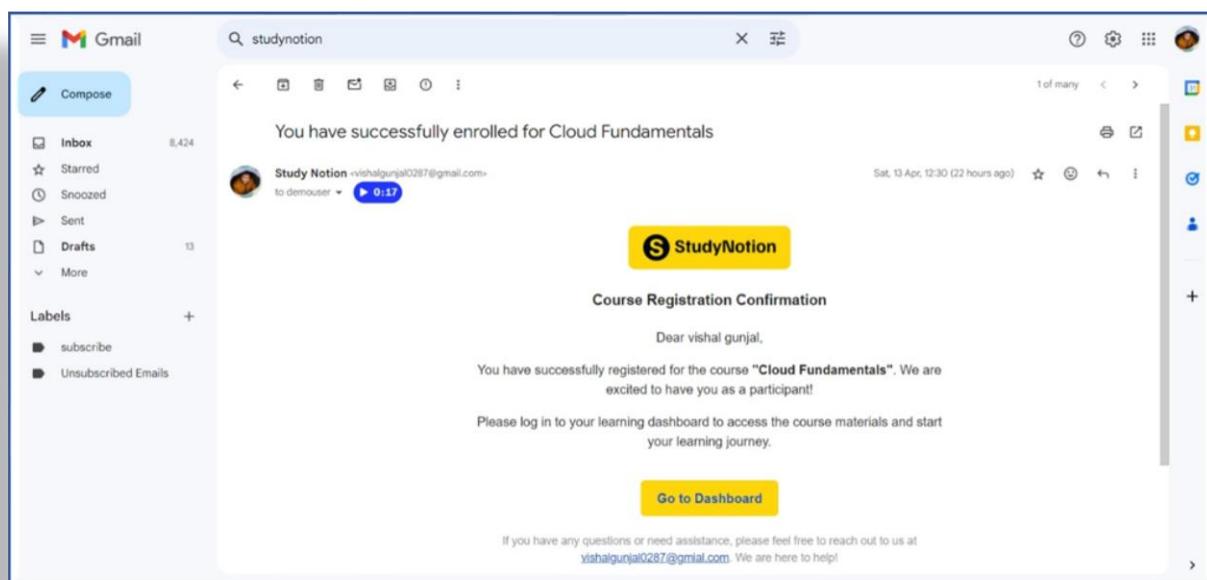
Buy Now Add to Cart

30-Day Money-Back Guarantee

This course includes

Instructions: Register for the course. Access the learning materials. Follow the schedule. Complete assessments. Seek help.

## 14. Course Registration Confirmation:



## 15. Enrolled Courses:

**StudyNotion**

Home Catalog About Us Contact Us Search

My Courses  
2 of 2 Lectures Completed

Pre-requisites  
Discussing Web Development in 2024

Starting the journey of MERN Stack

Lecture 1

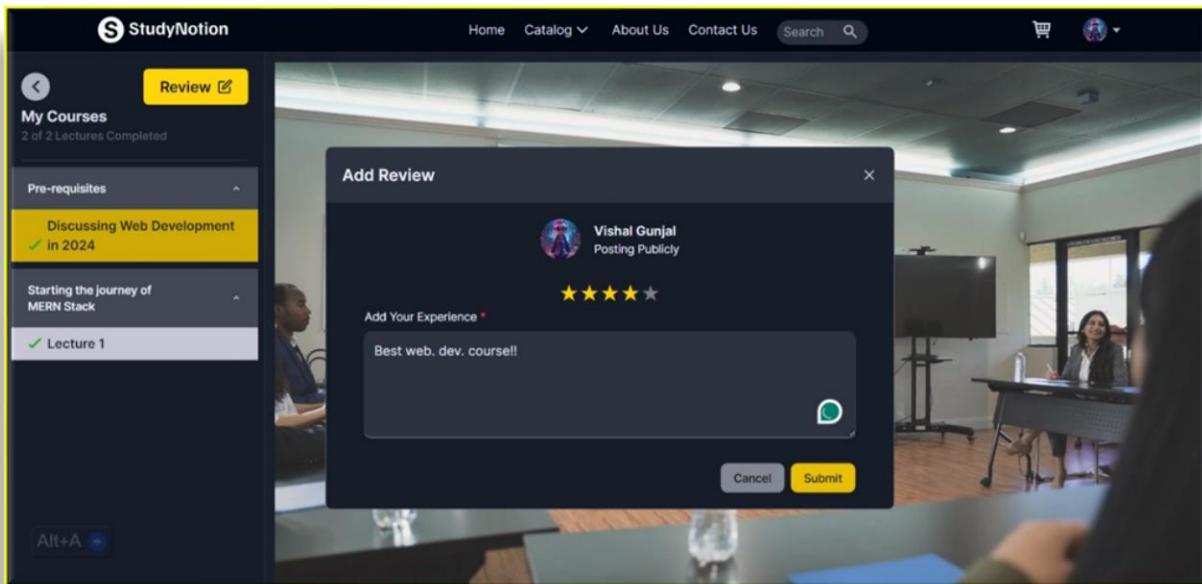
Discussing Web Development in 2024

| Resources   | Plans              | Subjects           | Languages  | Career building            |
|---|--------------------|--------------------|------------|----------------------------|
| Articles  | Paid memberships   | AI                 | Bash       | Career paths               |
| About   | For students       | Cloud Computing    | C++        | Career services            |
| Careers   | Business solutions | Code Foundations   | C#         | Interview prep             |
| Affiliates  | Code challenges    | Computer Science   | Go         | Professional certification |
| <a href="#">F</a> <a href="#">G</a> <a href="#">T</a> <a href="#">D</a> | Docs               | Cybersecurity      | HTML & CSS | -                          |
|   | Projects           | Data Analytics     | Java       | Full Catalog               |
|   | Videos             | Data Science       | JavaScript | Beta Content               |
|   | Workspaces         | Data Visualization | Kotlin     |                            |
| <b>Support</b>  | Forums             | Developer Tools    | PHP        |                            |
|   | Chapters           | DevOps             | Python     |                            |
|   | Events             | Game Development   | R          |                            |
|   |                    | IT                 | Ruby       |                            |
|   |                    | Machine Learning   | SQL        |                            |
|   |                    | Math               | Swift      |                            |
|   |                    | Mobile Development |            |                            |
|   |                    | Web Design         |            |                            |
|   |                    | Web Development    |            |                            |

Privacy Policy | Cookie Policy | Terms

Developed by Vishal D. Gunjal, © 2024 StudyNotion - Empowering Education  
A proficient S.Y MCA Student at MES's Institute of Management and Career Courses, Pune.

## 16. Student Add Review:



## 17. Student Profile Setting:

The screenshot shows the 'Edit Profile' page of the StudyNotion website. The left sidebar has 'My Profile', 'Enrolled Courses', 'Cart', and 'Logout'. The main area has a 'Settings' tab selected. The 'Profile Information' section contains fields for First Name (sandip), Last Name (pandhare), Date of Birth (09-05-2002), Gender (Male), Contact Number (9022717020), and About (Enter Bio Details). Below this are password fields: Old Password, New Password, and Confirm New Password, each with an 'Enter Password' placeholder and a 'Save' button. A 'Delete Account' dialog box is open at the bottom, asking if the user wants to delete their account, stating it's permanent and removes associated content, with a link to 'I want to delete my account.'

## 18. Instructor Profile Page

StudyNotion

Home Catalog About Us Contact Us Search VG

My Profile

Dashboard My Courses Add Course

Settings Logout

## My Profile

VG Vishal Gunjal demoinstructor@gmail.com Edit

About Write Something about Yourself Edit

Personal Details

| First Name | Last Name |
|------------|-----------|
| Vishal     | Gunjal    |

| Email                  | Phone Number       |
|------------------------|--------------------|
| doinstructor@gmail.com | Add Contact Number |

| Gender     | Date of Birth     |
|------------|-------------------|
| Add Gender | Add Date of Birth |

**Company**

- Articles
- Blog
- Chart Sheet
- Code challenges
- Docs
- Projects
- Videos
- Workspaces

**Community**

- Forums
- Chapters
- Events

**Support**

- Help Center

**Subjects**

- AI
- Cloud Computing
- Code Foundations
- Computer Science
- Cybersecurity
- Data Analytics
- Data Science
- Data Visualization
- Developer Tools
- DevOps
- Game Development
- IT
- Machine Learning
- Math
- Mobile Development
- Web Design
- Web Development

**Languages**

- Bash
- C++
- C#
- Go
- HTML & CSS
- Java
- JavaScript
- Kotlin
- PHP
- Python
- R
- Ruby
- SQL
- Swift

**Career building**

- Career paths
- Career services
- Interview prep
- Professional certification
- 
- Full Catalog
- Beta Content

Privacy Policy | Cookie Policy | Terms

Developed by Vishal D. Gunjal, © 2024 StudyNotion - Empowering Education  
A proficient S.Y MCA Student at MES's Institute of Management and Career Courses, Pune.

## 19. Instructor Dashboard Page

The screenshot shows the StudyNotion Instructor Dashboard. The top navigation bar includes links for Home, Catalog, About Us, Contact Us, a Search bar, and a user profile icon labeled 'VG'. On the left, a sidebar menu lists 'My Profile', 'Dashboard' (which is selected), 'My Courses', 'Add Course', 'Settings', and 'Logout'. The main content area starts with a greeting 'Hi Vishal' and a message 'Let's start something new'. It features a 'Visualize' section with a pie chart showing student distribution across three courses: DevOps Basics (orange), Complete JAVA Language (green), and Data Science Crash Course (blue). Below this is a 'Statistics' box displaying 'Total Courses: 3', 'Total Students: 7', and 'Total Earnings: ₹ 23329'. The 'Your Courses' section lists three courses: 'DevOps Basics' (3 students, ₹ 1111), 'Complete JAVA Language' (4 students, ₹ 4999), and 'Data Science Crash Course' (0 students, ₹ 7999). At the bottom, there is a footer with links for Privacy Policy, Cookie Policy, Terms, and a note about the developer: 'Developed by Vishal D. Gunjal, © 2024 StudyNotion - Empowering Education. A proficient S.Y MCA Student at MES's Institute of Management and Career Courses, Pune.'

## 20.Instructor Created Courses Page

The screenshot shows the 'MY Courses' section of the StudyNotion website. On the left is a sidebar with options: My Profile, Dashboard, My Courses (which is selected), Add Course, Settings, and Logout. The main area displays three courses:

- DevOps Basics**: Price ₹1111, Published.
- Complete JAVA Language**: Price ₹4999, Published.
- Data Science Crash Course**: Price ₹7999, Published.

Below the courses is a footer with links to Company, Resources, Plans, Subjects, Languages, and Career building sections. It also includes a Community section with forums, chapters, and events. The footer also contains links for Support and Social media (Facebook, Google+, Twitter, YouTube). At the bottom, there are links for Privacy Policy, Cookie Policy, Terms, and a note about the developer.

## 21.Instructor Add Courses Page

**StudyNotion**

- Home
- Catalog
- About Us
- Contact Us
- Search
- VG

**Add Course**

Course Information      Course Builder      Publishing Course

**Course Title\***  
Enter Course Title

**Course Short Description\***  
Enter Description

**Course Price\***  
Enter Course Price

**Course Category\***  
Choose a Category

**Tags\***  
Press Enter or , to add a tag

**Course Thumbnail\***  
Drag and drop an image, or click to [Browse](#) a file  
Aspect ratio 16:9   Recommended size 1024x576

**Benefits of the course\***  
Enter Benefits of the course

**Requirements/Instructions\***

**Add**

**Next**

**Course Upload Tips**

- Set the Course Price option or make it free.
- Standard size for the course thumbnail is 1024x576.
- Video section controls the course overview video.
- Course Builder is where you create and organize a course.
- Add Topics in the Course Builder section to create lessons, quizzes, and assignments.
- Information from the Additional Data section shows up on the course single page.
- Make Announcements to notify any important notes to all enrolled students at once.

**S StudyNotion**

**Company**

- Articles
- Blog
- Chart Sheet
- Code challenges
- Docs
- Projects
- Videos
- Workspaces

**Support**

- Help Center

**Resources**

- Paid memberships
- For students
- Business solutions

**Community**

- Forums
- Chapters
- Events

**Plans**

- AI
- Cloud Computing
- Code Foundations
- Computer Science
- Cybersecurity
- Data Science
- Data Visualization
- Developer Tools
- Game Development
- IT
- Machine Learning
- Math
- Mobile Development
- Web Design

**Subjects**

- Bash
- C++
- C#
- Go
- HTML & CSS
- Java
- JavaScript
- Kotlin
- PHP
- Python
- R
- Ruby
- SQL
- Swift

**Languages**

- Career paths
- Career services
- Interview prep
- Professional certification
- .
- Full Catalog
- Beta Content

**Career building**

Privacy Policy | Cookie Policy | Terms

Developed by Vishal D. Gunjal, © 2024 StudyNotion - Empowering Education  
A proficient S.Y MCA Student at MES's Institute of Management and Career Courses, Pune.

## 22.Instructor Profile Setting:

The screenshot shows the 'Edit Profile' page of the StudyNotion website. The top navigation bar includes links for Home, Catalog, About Us, Contact Us, and a search bar. A user icon with 'VG' is in the top right. The left sidebar has links for My Profile, Dashboard, My Courses, Add Course, Settings (which is selected), and Logout.

**Edit Profile**

**Profile Information**

|                      |                   |
|----------------------|-------------------|
| First Name           | Last Name         |
| Vishal               | Gunjal            |
| Date of Birth        | Gender            |
| dd-mm-yyyy           | Prefer not to say |
| Contact Number       | About             |
| Enter Contact Number | Enter Bio Details |

**Old Password \***  
Enter Password

**New Password \***  
Enter Password

**Confirm New Password \***  
Enter Password

**Delete Account**

Would you like to delete account?  
This account may contain Paid Courses. Deleting your account is permanent and will remove all the content associated with it.

I want to delete my account.

## **4.1 Algorithm:**

- **App.js**

Step 1 - Start

Step 2 – Input the Username and Password

Step 3 – Go to next

Step 4 – If Username and Password are correct then go to step 5

otherwise go step 2

Step 5 – Open the Home.jsx Page

Step 6 – If User forget password then go to step 7.

Step 7 – Go to forgot your password

Step 8 – If you are new user then click on signup otherwise go to step 2

Step 9 – Stop

- **Signup.jsx**

Step 1 – Start

Step 2 – Input name, Email, Address, Username, Password

Step 3 – Enter you received OTP on your register E-mail id

Step 4 – Click on Create Account.

Step 5 – If OTP is invalid then user go to Step 2

Step 7 – Stop

- **Login.jsx**

Step 1 – Start

Step 2 – Input Username and Password

Step 3 – Click on Login

Step 4 – Validate Username and Password

Step 5 – If validation fails, display error message and prompt user to re-enter credentials (Go back Step 2)

Step 6 – If validation succeeds, proceed to dashboard or main page

Step 7 - Stop

## 4.2 Code Snippets

### 1. Home Page:

```
import React, { useEffect, useState } from "react";
import { FaArrowRight } from "react-icons/fa";
import { Link } from "react-router-dom";
import CTAButton from "../Components/core/HomePage/Button";
import HighlightText from "../Components/core/HomePage/HighlightText";
import Banner from "../assets/Images/banner.mp4";
import CodeBlocks from "../Components/core/HomePage/CodeBlocks";
import TimelineSection from "../Components/core/HomePage/TimelineSection";
import LearningLanguageSection from
  "../Components/core/HomePage/LearningLanguageSection";
import InstructorSection from "../Components/core/HomePage/InstructorSection";
import ExploreMore from "../Components/core/HomePage/ExploreMore";
import { useDispatch } from "react-redux";
import { setProgress } from "../slices/loadingBarSlice";
import RatingSlider from "../Components/core/Ratings/RatingSlider";
import { motion } from "framer-motion";
import { fadeIn } from "../../Components/common/motionFrameVariants";

function Home() {
  const dispatch = useDispatch();
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    // Fetch data or perform any necessary initialization
    dispatch(setProgress(0)); // Example dispatch call
    setLoading(false);
  }, [dispatch]);
```

```

return (
  <div>
    {/* Section 1 */}
    <div className="relative mx-auto flex w-11/12 max-w-maxContent flex-col items-center justify-between gap-8 text-white">
      {/* Become an Instructor Button */}
      <Link to={"/signup"}>
        <div className="group mx-auto mt-16 w-fit rounded-full bg-richblack-800 p-1 font-bold text-richblack-200 drop-shadow-[0_1.5px_rgba(255,255,255,0.25)] transition-all duration-200 hover:scale-95 hover:drop-shadow-none">
          <div className="flex flex-row items-center gap-2 rounded-full px-10 py-[5px] transition-all duration-200 group-hover:bg-richblack-900">
            <p>Become an Instructor</p>
            <FaArrowRight />
          </div>
        </div>
      </Link>

      {/* Heading */}
      <motion.div
        variants={fadeIn("left", 0.1)}
        initial="hidden"
        whileInView={"show"}
        viewport={{ once: false, threshold: 0.1 }}
        className="text-center text-3xl lg:text-4xl font-semibold mt-7">
        >
        Empower Your Future with
        <HighlightText text={"Coding Skills"} />
      
```

```
</motion.div>

<motion.div
  variants={fadeIn("right", 0.1)}
  initial="hidden"
  whileInView={"show"}
  viewport={{ once: false, threshold: 0.1 }}
  className="mt-4 w-[90%] text-center text-base lg:text-lg font-bold text-richblack-300"
>
  With our online coding courses, you can learn at your own pace, from
  anywhere in the world, and get access to a wealth of resources,
  including hands-on projects, quizzes, and personalized feedback from
  instructors.
</motion.div>

 {/* CTA Buttons */}
<div className="mt-8 flex flex-row gap-7">
  <CTAButton active={true} linkto={"/signup"}>
    Learn More
  </CTAButton>
  <CTAButton active={false} linkto={"/login"}>
    Book a Demo
  </CTAButton>
  <CTAButton active={true} linkto={"/signup"}>
    Watch Live
  </CTAButton>
</div>
```

```
{/* Video */}

<div className="mx-3 my-7 shadow-[10px_-5px_50px_-5px] shadow-blue-200">

  <video

    className="shadow-[20px_20px_rgba(255,255,255)]"

    muted

    loop

    autoComplete

  >

  <source src={Banner} type="video/mp4" />

</video>

</div>
```

```
{/* Code Section 1 */}

<div>

  <CodeBlocks

    position={"lg:flex-row"}

    heading={

      <div className="text-3xl lg:text-4xl font-semibold">

        Unlock your

        <HighlightText text={"coding potential"} /> with our online

        courses.

      </div>

    }

    subheading={

      "Our courses are designed and taught by industry experts who have years of

      experience in coding and are passionate about sharing their knowledge with you."

    }

    ctabtn1={
```

```

        btnText: "Try it Yourself",
        link: "/signup",
        active: true,
    } }

ctabtn2={ {
    btnText: "Learn More",
    link: "/signup",
    active: true,
} }

codeColor={"text-yellow-25"}

codeblock={`<!DOCTYPE html>
<html lang="en">
<head>
<title>This is myPage</title>
</head>
<body>
<h1>
    <a href="/">Header</a></h1>
    <nav>
        <a href="/one">One</a>
        <a href="/two">Two</a>
        <a href="/three">Three</a>
    </nav>
</body>
`}

backgroundGradient={

    <div className="code-block1-grad codeblock1 absolute"></div>
}

/>

</div>

/* Most Popular

<div className=" mx-auto box-content w-full max-w-maxContentTab px- py-12 lg:max-w-maxContent">

    <h2 className="section_heading mb-6 md:text-3xl text-xl">
        Most Popular Courses
    </h2>

    <CourseSlider Courses={getCatalogaPageData?.selectedCourses} />

</div> */}

```

```
/* Code Section 2 */

<div>
  <CodeBlocks
    position={"lg:flex-row-reverse"}
    heading={
      <div className="w-[100%] text-4xl font-semibold lg:w-[50%]">
        Start
        <HighlightText text={"coding in seconds"} />
      </div>
    }
    subheading={
      "Go ahead, give it a try. Our hands-on learning environment means you'll be
      writing real code from your very first lesson."
    }
    ctabtn1={
      btnText: "Continue Lesson",
      link: "/signup",
      active: true,
    }
    ctabtn2={
      btnText: "Learn More",
      link: "/signup",
      active: false,
    }
    codeColor={"text-white"}
    codeblock={`import React from "react";\nimport CTAButton from\n"./Button";\nimport TypeAnimation from "react-type";\nimport { FaArrowRight } from\n"react-icons/fa";\n\nconst Home = () => {\n  return (\n    <div>Home</div>\n  )\n}\n\nexport default Home;`}
  </CodeBlocks>
</div>
```

```
backgroundGradient={<div className="codeblock2 absolute"></div>}

/>

</div>

{/* Explore Section */}

<ExploreMore />

</div>

{/* Section 2 */}

<div className="bg-pure-greys-5 text-richblack-700">
  <div className="homepage_bg h-[320px]">
    {/* Explore Full Category Section */}
    <div className="mx-auto flex w-11/12 max-w-maxContent flex-col items-center justify-between gap-8">
      <div className="lg:h-[150px]"></div>
      <div className="flex flex-row gap-7 text-white lg:mt-8">
        <CTAButton active={true} linkto={"/signup"}>
          <div className="flex items-center gap-2">
            Explore Full Catalog
            <FaArrowRight />
          </div>
        </CTAButton>
        <CTAButton active={false} linkto={"/login"}>
          Learn More
        </CTAButton>
      </div>
    </div>
  </div>
</div>
```

```
<div className="mx-auto flex w-11/12 max-w-maxContent flex-col items-center justify-between gap-8 ">

  {/* Job that is in Demand - Section 1 */}

  <div className="mb-10 mt-[100px] flex flex-col justify-between gap-7 lg:mt-20 lg:flex-row lg:gap-0">

    <div className="text-4xl font-semibold lg:w-[45%]">
      Get the skills you need for a{" "}
      <HighlightText text={"job that is in demand."} />
    </div>

    <div className="flex flex-col items-start gap-10 lg:w-[40%]">
      <div className="text-[16px]">
        The modern StudyNotion is the dictates its own terms. Today, to
        be a competitive specialist requires more than professional
        skills.
      </div>
      <CTAButton active={true} linkto={"/signup"}>
        <div className="">Learn More</div>
      </CTAButton>
    </div>
  </div>

  {/* Timeline Section - Section 2 */}

  <TimelineSection />

  {/* Learning Language Section - Section 3 */}

  <LearningLanguageSection />
</div>
</div>
```

```
{/* Section 3 */}

<div className="relative mx-auto my-20 flex w-11/12 max-w-maxContent flex-col items-center justify-between gap-8 bg-richblack-900 text-white">

  {/* Become an instructor section */}
  <InstructorSection />

</div>

  {/* Review Slider here */}

<div className=" mb-16 mt-3">

  <h2 className="text-center text-5xl md:text-4xl font-semibold mt-8 text-richblack-5 mb-5">

    <span className="bg-gradient-to-br from-[#833AB4] via-[#FD1D1D] to-[#FCB045] bg-clip-text text-4xl font-semibold text-transparent lg:w-[70%]">

      Reviews From Other Learners

    </span>
  </h2>
  <div>
    <RatingSlider />
  </div>
  </div>
</div>

);

}

export default Home;
```

## 2. App.jsx:

```
import "./App.css";  
//import React, { useEffect } from "react";  
import { Routes, Route } from "react-router-dom";  
import Home from "./pages/Home";  
import NavBar from "./Components/common/NavBar";  
import Footer from "./Components/common/Footer";  
import Login from "./pages/Login";  
import Signup from "./pages/Signup";  
import ForgotPassword from "./pages/ForgotPassword";  
import ResetPassword from "./pages/ResetPassword";  
import VerifyOtp from "./pages/VerifyOtp";  
import About from "./pages/About";  
import ContactUs from "./pages/ContactUs";  
import LoadingBar from "react-top-loading-bar";  
import { setProgress } from "./slices/loadingBarSlice";  
import { useSelector } from "react-redux";  
import { useDispatch } from "react-redux";  
import Dashboard from "./pages/Dashboard";  
import OpenRoute from "./Components/core/Auth/OpenRoute";  
import PrivateRoute from "./Components/core/Auth/PrivateRoute";  
import MyProfile from "./Components/core/Dashboard/MyProfile";  
import Setting from "./Components/core/Dashboard/Settings";  
import EnrollledCourses from "./Components/core/Dashboard/EnrolledCourses";  
import Cart from "./Components/core/Dashboard/Cart/index";  
import { ACCOUNT_TYPE } from "./utils/constants";  
import AddCourse from "./Components/core/Dashboard/AddCourse/index";  
import MyCourses from "./Components/core/Dashboard/MyCourses/MyCourses";  
import EditCourse from "./Components/core/Dashboard/EditCourse.jsx/EditCourse";
```

```

import Catalog from "./pages/Catalog";
import ScrollToTop from "./Components/ScrollToTop";
import CourseDetails from "./pages/CourseDetails";
import SearchCourse from "./pages/SearchCourse";
import ViewCourse from "./pages/ViewCourse";
import VideoDetails from "./Components/core/ViewCourse/VideoDetails";
import PurchaseHistory from "./Components/core/Dashboard/PurchaseHistory";
import InstructorDashboard from
"./Components/core/Dashboard/InstructorDashboard/InstructorDashboard";
import { RiWifiOffLine } from "react-icons/ri";
import AdminPannel from "./Components/core/Dashboard/AdminPannel";

function App() {
  console.log = function () {};
  const user = useSelector((state) => state.profile.user);
  const progress = useSelector((state) => state.loadingBar);
  const dispatch = useDispatch();

  // useEffect(() => {
  //   // Disable right-click context menu
  //   // document.addEventListener("contextmenu", function (event) {
  //   //   event.preventDefault();
  //   // });
  //   // }, []);

  return (
    <div className=" w-screen min-h-screen bg-richblack-900 flex flex-col font-inter">
      <LoadingBar
        color="#FFD60A"

```

```
height={1.4}

progress={progress}

onLoaderFinished={() => dispatch(setProgress(0))}

/>

<NavBar setProgress={setProgress}></NavBar>

{ !navigator.onLine && (

<div className="bg-red-500 flex text-white text-center p-2 bg-richblack-300 justify-
center gap-2 items-center">

<RiWifiOffLine size={22} />

Please check your internet connection.

<button

className="ml-2 bg-richblack-500 rounded-md p-1 px-2 text-white"

onClick={() => window.location.reload()}

>

Retry

</button>

</div>

)}

<ScrollToTop />

<Routes>

<Route path="/" element={<Home />} />

<Route path="/catalog/:catalog" element={<Catalog />} />

<Route

path="/login"

element={

<OpenRoute>

<Login />
```

```
</OpenRoute>
}

/>

<Route
  path="/signup"
  element={
    <OpenRoute>
      <Signup />
    </OpenRoute>
  }
/>

<Route path="/forgot-password" element={<ForgotPassword />} />

<Route path="/update-password/:id" element={<ResetPassword />} />

<Route path="/verify-email" element={<VerifyOtp />} />

<Route path="/about" element={<About />} />

<Route path="/contact" element={<ContactUs />} />

<Route path="/courses/:courseId" element={<CourseDetails />} />

<Route path="/search/:searchQuery" element={<SearchCourse />} />

<Route
  element={
```

```

<PrivateRoute>
  <Dashboard />
</PrivateRoute>
}

>

<Route path="dashboard/my-profile" element={<MyProfile />} />
<Route path="dashboard/settings" element={<Setting />} />
{user?.accountType === ACCOUNT_TYPE.STUDENT && (
  <>
    <Route path="dashboard/cart" element={<Cart />} />
    <Route
      path="dashboard/enrolled-courses"
      element={<EnrolldedCourses />}
    />
    <Route
      path="dashboard/purchase-history"
      element={<PurchaseHistory />}
    />
  </>
)
}

{user?.accountType === ACCOUNT_TYPE.INSTRUCTOR && (
  <>
    <Route path="dashboard/add-course" element={<AddCourse />} />
    <Route path="dashboard/my-courses" element={<MyCourses />} />
    <Route
      path="dashboard/edit-course/:courseId"
      element={<EditCourse />}
    />
    <Route

```

```
path="dashboard/instructor"
element={<InstructorDashboard />}
/>
</>
)}
{user?.accountType === ACCOUNT_TYPE.ADMIN && (
<>
<Route path="dashboard/admin-panel" element={<AdminPannel />} />
</>
)}
</Route>

<Route
element={
<PrivateRoute>
<ViewCourse />
</PrivateRoute>
}
>
{user?.accountType === ACCOUNT_TYPE.STUDENT && (
<>
<Route
path="/dashboard/enrolled-courses/view-course/:courseId/section/:sectionId/sub-
section/:subsectionId"
element={<VideoDetails />}
/>
</>
)}
</Route>
```

```
<Route path="/" element={<Home />} />
</Routes>
<Footer />
</div>
);
}

export default App;
```

Testing is the process of detecting errors. Testing performs a very critical role. Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

### **Objective of Testing:**

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say, Testing is a process of executing a program with intent of finding an error. A successful test is one that uncovers an as yet undiscovered error. A good test case is one that uncovers an as yet undiscovered error. A good testcase is one that has a high probability of finding an error, if it exists. The tests are inadequate to detect possibly present errors. The software more or less confirms to the quality and reliable standards. Or less confirms to the quality and reliable standards to a satisfactory extent.

## **5.1 Test Strategy**

### **Revision History**

| <b>Date</b> | <b>Version</b> | <b>Author</b> | <b>Description</b>                                  |
|-------------|----------------|---------------|---|
| 28/03/2024  | 1.0            | Vishal Gunjal | QA Test Strategy for StudyNotion An EdTech Platform |
|             |                |               |   |

### **Scope**

It defines parameters like

- ✓ Document will review by internal guide and external examiner of university.
- ✓ Document will approve by Internal guide.
- ✓ Interface is scope of testing
- ✓ Not in scope – Integration testing
- ✓ In Scope – Unit testing, Acceptance testing and functionality.

### **Test Approach It defines**

- ✓ Process of testing involves 6 steps
- ✓ levels are there in this document
- ✓ Roles and responsibilities of each team member

| <b>Tester</b> | <b>Module</b> | <b>Time</b> | <b>Responsibility</b> |
|---------------|---------------|-------------|-----------------------|
| Vishal Gunjal | All modules   | 10 hours    | All                   |
|               |               |             |                       |

- ✓ Types of Testing (Unit testing, Acceptance testing and Functionality etc.)
- ✓ Testing approach and automation tool if applicable Test Environment
- ✓ Backup of test data and restore strategy is copy of original code and document

### **Testing Tool**

- ✓ Automation and Test management tools needed for test execution
- ✓ Figure out number of open-source as well as commercial tools required and determine how many users are supported on it and plan accordingly

### **Release Control**

- ✓ Release management plan with appropriate version history that will make sure test execution for all modification in that release

### **Risk Analysis**

- ✓ It includes: Risk identification, Risk analysis, Risk evaluation
- ✓ Identify the possible threats then estimate their likely impacts if they were to happen.

### **Review and Approvals**

- ✓ All these activities are reviewed and sign off by the Guide.
- ✓ Summary of review changes should be traced at the beginning of the document along with approved date, name, and comment.

## **5.2 Unit Test Plan**

1. The Project will be divided into smaller units, with all modules organized chronologically within the document.
2. Each test case will be meticulously recorded with a detailed description.
3. Test cases will include the date and time of the test, tester's name, test results, and any comments.
4. The primary objective is to identify and rectify any major faults in the system promptly.

### **5.3 Acceptance Test Plan**

1. The Modules will be tested as one by one and will be used to find if any problem is occurring in important points of the project requirements.
2. The data here used would be large to create stress on system to observe its real time working efforts.
3. The most important process here is to find that if systems good enough from the client point of view.
4. All the results will be recorded for further testing and future enhancements.

## 5.4 Test Case

| Index  |                                      |
|--|--------------------------------------|
| <b>Document Name:</b> Test Plan Document     | <b>Document Version No.</b> TPD_V1.0 |
| <b>Author of the document:</b> Vishal Gunjal | <b>Date of Creation:</b> 28/03/2024  |
| <b>Approved By:</b> Dr. Shilpa Paralikar     | <b>Date of Approval:</b> 12/04/2024  |

|                               |  |
|-------------------------------|--|
| <b>Project Name:</b>          | StudyNotion An EdTech Platform   |
| <b>Test Plan ID:</b>          | TPD_V-1.0<br>[This document is modifiable according to the requirements & change plan.]  |
| <b>Introduction:</b>          | StudyNotion Web Application is the web base system which is helpful to user for purchasing courses from a particular area.   |
| <b>Test Objective:</b>        | 1) Test should cover functionalities mentioned in the Use case diagram for all modules & submodules.<br><br>2) To test correctness, completeness of User Interface, Functional, Database level test cases.<br>Test Plan, Test Scenarios & Test Cases should be written & submitted as per deadlines. |
| <b>Test Scope:</b>            | 1) End to End System Testing.  |
| <b>Features to be tested:</b> | User Interface, Usability Testing, System Testing<br><br>Functional Testing<br><br>3) Database Testing – DB Connectivity, Data Insert/Update/Retrieve operations.  |

|                                   |  |
|-----------------------------------|--|
|                                   | <p>Working of system with different customer and merchant username but one at a time should be tested for all functionalities in both the browsers.</p>  |
| <b>Features not to be tested:</b> | <p>1) Non-functional Scenarios like - Load, Stress, Performance related testing, Security Testing</p> <p>2) System should be tested with one user only. With multiple for concurrent users' system should not be tested.</p> <p>As this is already built in system, module integration testing should not be performed.</p>  |
| <b>Assumption:</b>                | <p>1) System should have database or SQL query through which database can be generated.</p> <p>2) Provision for default login username/password to the system should be provided. After building system, it should be logged in with default credentials.</p> <p>4) Smoke Test should be passed so that tester will start for further testing.</p> <p>Test Plan for creation &amp; execution of test cases &amp; defect reporting,</p> |

|  |                              |
|--|------------------------------|
|  | final submission, Vivas etc. |
|--|------------------------------|

|                       |  |
|-----------------------|--|
| <b>Risk Analysis:</b> | <p>Testing process may affect due to -</p> <p>Cracked or limited edition of software application setups installation.</p> <p>Antivirus application is not installed or not updated on daily basis.</p> <p><b>3)</b> Keeping Test Documents in USB PD only with no backup &amp; connecting same in IMCC LAB 1. (Word &amp; Excel Files will be converted to rcs&lt;=&gt;scr format which is screensaver format which makes file corrupt for forever. Every antivirus asks to delete this file.)</p> <p>Health problems - Injuries, Illness due to climate changes etc.</p> <p>Emergencies - Medical, Accidental, Others</p> |
| <b>Test Strategy:</b> | <p><b>1)</b> Creating project analysis document. (System specification, Business Scenario for system use)</p> <p><b>2)</b> Based on project analysis, draw Use Case Diagram, Activity Diagram, ER Diagram.</p> <p>Preparing test plan.</p> <p>Creating test scenarios.</p> <p>Writing test cases.</p> <p>Execution of test cases.</p> <p>Defect reporting for failed test cases.</p> <p>Test Report generation.</p> <p>Final Presentation (With Vivas)</p>   |

## Project Details

**Project Name:** StudyNotion An EdTech Platform

**Document Name:** Test Case Document

**Document Approved By:** Dr. Shilpa Paralikar

| <b>Test Case No.</b> | <b>Senario To</b>               | <b>Steps To Perform</b>          | <b>Expected Result</b>                           | <b>Actual Result</b>                                | <b>Pass / Fail</b> |
|----------------------|---------------------------------|----------------------------------|--|---|--------------------|
| TC001                | Front-end: Homepage display     | 1. Open StudyNotion homepage     | Homepage displays with brief intro and links     | Homepage displayed as expected                      | Pass               |
|                      |                                 |                                  | to course list and user details                  |   |                    |
| TC002                | Front-end: Course List display  | 1. Navigate to Course List page  | Course List displays with courses, descriptions, | Course List displayed with courses and descriptions | Pass               |
|                      |                                 |                                  | and ratings                                      | and ratings displayed as expected                   |                    |
| TC003                | Front-end: Wishlist display     | 1. Navigate to Wishlist page     | Wishlist displays with courses added by user     | Wishlist displayed with added courses               | Pass               |
| TC004                | Front-end: Course Checkout      | 1. Add a course to cart          | User proceeds to checkout page                   | User was able to proceed to checkout                | Pass               |
|                      |                                 | 2. Complete checkout process     | Course is purchased successfully                 | Course was successfully purchased                   | Pass               |
| TC005                | Front-end: User Profile Details | 1. Navigate to User Profile page | User details (name, email, etc.) display         | User details displayed as expected                  | Pass               |
| TC006                | Front-end: Edit User Profile    | 1. Navigate to Edit Profile page | User can edit their profile details              | User was able to edit profile details               | Pass               |

|           |   |  |   |   |      |
|-----------|---|--|---|---|------|
|           |   |  |   |   |      |
| TC007     | Back-end:<br>User<br>Authentication             | 1. Register a<br>new user  | User is registered<br>successfully        | User registered<br>successfully             | Pass |
|           |   | 2. Login with<br>registered<br>credentials                             | User logs in successfully                 | User logged in<br>successfully              | Pass |
| TC008     | Back-end:<br>Course<br>Creation                 | 1. Create a new<br>course  | Course is created<br>successfully         | Course created<br>successfully              | Pass |
|           |   | 2. Update<br>existing course   | Course is updated<br>successfully         | Course updated<br>successfully              | Pass |
| TC009     | Back-end:<br>Course<br>Deletion                 | 1. Delete an<br>existing course  | Course is deleted<br>successfully         | Course deleted<br>successfully              | Pass |
| TC0<br>10 | Back-end:<br>Payment<br>Integration             | 1. Add a<br>course to cart   | Course is added to cart<br>successfully   | Course added<br>to cart<br>successfully     | Pass |
|           |   | 2. Complete<br>checkout<br>process                                     | Payment is completed<br>successfully      | Payment<br>completed<br>successfully        | Pass |
| TC0<br>11 | Back-end:<br>Cloud-based<br>media<br>management | 1. Upload<br>media content<br>for a course                             | Media content is uploaded<br>successfully | Media content<br>uploaded<br>successfully   | Pass |
| TC0<br>12 | Back-end:<br>Markdown<br>formatting             | 1. Store course<br>content in<br>Markdown<br>format                    | Content is stored in<br>Markdown format   | Content stored<br>in Markdown<br>format     | Pass |
| TC0<br>13 | API: User<br>Registration                       | 1. Send POST<br>request to<br>/api/auth/signu<br>p with valid<br>creds | User is registered<br>successfully        | User registered<br>successfully             | Pass |
| TC0<br>14 | API: User<br>Login                              | 1. Send POST<br>request to<br>/api/auth/login<br>with valid<br>creds   | User logs in successfully                 | User logged in<br>successfully              | Pass |
| TC0<br>15 | API: Course<br>Retrieval                        | 1. Send GET<br>request to<br>/api/courses                              | List of courses is returned               | List of courses<br>returned<br>successfully | Pass |

|           |   |  |  |  |      |
|-----------|---|--|--|--|------|
|           |   |  |  |  |      |
| TC0<br>16 | API: Course Creation                    | 1. Send POST request to /api/courses with course details     | Course is created successfully                                     | Course created successfully            | Pass |
| TC0<br>17 | API: Course Update                      | 1. Send PUT request to /api/courses/:id with updated info    | Course is updated successfully                                     | Course updated successfully            | Pass |
| TC0<br>18 | API: Course Deletion                    | 1. Send DELETE request to /api/courses/:id                   | Course is deleted successfully                                     | Course deleted successfully            | Pass |
| TC0<br>19 | API: Course Rating                      | 1. Send POST request to /api/courses/:id/rate with rating    | Rating is added successfully                                       | Rating added successfully              | Pass |
| TC0<br>20 | API: User Password Reset                | 1. Send POST request to /api/auth/forgot-password with email | Password reset email is sent successfully                          | Password reset email sent successfully | Pass |
| TC0<br>21 | Front-end: Cart Checkout Error Handling | 1. Proceed to checkout without adding any course to the cart | Error message displayed prompting user to add a course to the cart | Error message displayed as expected    | Pass |
| TC0<br>22 | Front-end: Course Content Display       | 1. Navigate to a course's content page                       | Course content including videos, documents, etc. are displayed     | Course content displayed correctly     | Pass |
| TC0<br>23 | Front-end: User Details Update          | 1. Navigate to User Profile page                             | 2. Edit user details and save changes                              | User details updated successfully      | Pass |
| TC0<br>24 | Front-end: Course Rating Display        | 1. Navigate to a course page                                 | 2. Rate the course   | Course rating is displayed and updated | Pass |

|           |  |   |  |  |      |
|-----------|--|---|--|--|------|
|           |  |   |  |  |      |
| TC0<br>25 | Back-end:<br>User<br>Authentication<br>Failure   | 1. Attempt to log in with incorrect credentials                   | Error message displayed indicating authentication failure              | Authentication failure handled correctly | Pass |
| TC0<br>26 | Back-end:<br>Course<br>Creation<br>Validation    | 1. Attempt to create a course with missing or invalid details     | Error message displayed prompting user to provide valid course details | Validation error handled correctly       | Pass |
| TC0<br>27 | Back-end:<br>Course<br>Deletion<br>Authorization | 1. Attempt to delete a course without appropriate permissions     | Error message displayed indicating unauthorized deletion               | Authorization error handled correctly    | Pass |
| TC0<br>28 | API: Invalid<br>API Endpoint                     | 1. Send request to an invalid API endpoint                        | Error message returned indicating invalid endpoint                     | Invalid endpoint handled correctly       | Pass |
| TC0<br>29 | API:<br>Unauthorized<br>Access                   | 1. Send request with invalid or missing authentication token      | Error message returned indicating unauthorized access                  | Unauthorized access handled correctly    | Pass |
| TC0<br>30 | API: Rate<br>Course Error<br>Handling            | 1. Attempt to rate a course without providing a rating value      | Error message returned indicating missing rating value                 | Error handled correctly                  | Pass |
| TC0<br>31 | API: Update<br>Course<br>Unauthorized            | 1. Attempt to update a course without appropriate permissions     | Error message returned indicating unauthorized access                  | Unauthorized access handled correctly    | Pass |
| TC0<br>32 | API: Delete<br>Course                            | 1. Send DELETE request to /api/courses/:id with valid credentials | Course is deleted successfully   | Course deleted successfully              | Pass |

|           |   |  |  |  |      |
|-----------|---|--|--|--|------|
|           |   |  |  |  |      |
| TC0<br>33 | API: Rate Course Over Limit             | 1. Attempt to rate a course with a rating value over the limit       | Error message returned indicating rating value over limit            | Rating over limit handled correctly                  | Pass |
| TC0<br>34 | API: Retrieve Non-existent Course       | 1. Send GET request to /api/courses/:id with invalid course ID       | Error message returned indicating course not found                   | Non-existent course handled correctly                | Pass |
| TC0<br>35 | API: Update Non-existent Course         | 1. Send PUT request to /api/courses/:id with invalid course ID       | Error message returned indicating course not found                   | Non-existent course handled correctly                | Pass |
| TC0<br>36 | API: Delete Non-existent Course         | 1. Send DELETE request to /api/courses/:id with invalid course ID    | Error message returned indicating course not found                   | Non-existent course handled correctly                | Pass |
| TC0<br>37 | API: Invalid Request Body               | 1. Send request with invalid request body                            | Error message returned indicating invalid request body               | Invalid request body handled correctly               | Pass |
| TC0<br>38 | API: Invalid Authentication Token       | 1. Send request with invalid authentication token                    | Error message returned indicating invalid authentication token       | Invalid authentication token handled correctly       | Pass |
| TC0<br>39 | API: Invalid HTTP Method                | 1. Send request using an unsupported HTTP method                     | Error message returned indicating unsupported HTTP method            | Unsupported HTTP method handled correctly            | Pass |
| TC0<br>40 | API: Invalid Course ID in Rate Endpoint | 1. Send POST request to /api/courses/:id/rate with invalid course ID | Error message returned indicating invalid course ID in rate endpoint | Invalid course ID in rate endpoint handled correctly | Pass |

## **5.5 Defect Report:**

A defect in a software product, also referred to as a bug, error, or fault, occurs when the software deviates from the expected behaviour outlined in the software requirements. This can manifest as incorrect data, system hangs, unexpected errors, or missing/incorrect requirements.

A defect report is a document that provides concise details about identified defects, including the actions that trigger them and the expected results. These reports are typically generated by the Quality Assurance team or end-users (customers). Customers often detect defects and report them to the software development support team, as they extensively explore the application's features.

The purpose of defect reports is to assist developers in easily identifying and resolving defects. QA assigns defect reports to developers, who reproduce the defects based on the provided action steps. Subsequently, developers fix the defects to achieve the desired outcome specified in the report.

Defect reports are crucial as they streamline the defect resolution process and ensure that developers understand how the defects were detected and what actions triggered them.

Effective defect reports are short, organized, and directly address the information necessary for developers to identify and rectify the defects. They should provide clear instructions on how to reproduce the issue and specify the expected outcome.

It's common for QA teams to receive defect reports from clients that are either too brief to reproduce and resolve or overly verbose, making it difficult to ascertain the root cause of the issue. Therefore, creating defect reports with precision and clarity is essential for efficient defect resolution.

To address the limitations and further improve the StudyNotion platform, the following enhancement is proposed:

### **Implementation of Adaptive Learning Algorithms:**

**Overview:** Adaptive learning algorithms leverage artificial intelligence (AI) and machine learning (ML) techniques to personalize the learning experience for each user based on their individual learning preferences, strengths, and weaknesses.

#### **How It Works:**

- The platform collects data on user interactions, including course preferences, learning styles, and performance metrics.
- Adaptive algorithms analyze this data to identify patterns and trends, allowing the platform to tailor course recommendations, content delivery, and assessment methods to each user's unique needs.
- As users engage with the platform, the algorithms continuously adapt and refine their recommendations to optimize learning outcomes.

#### **Benefits:**

- **Personalized Learning:** Adaptive algorithms enable StudyNotion to deliver customized learning experiences tailored to each user's specific requirements, maximizing engagement and knowledge retention.
- **Targeted Support:** By identifying areas of difficulty or gaps in understanding, the platform can provide targeted interventions, such as additional resources or adaptive feedback, to support learners in overcoming challenges.
- **Enhanced Engagement:** Personalized content and adaptive assessments enhance user engagement by aligning course materials with individual interests, learning styles, and skill levels.

- **Improved Learning Outcomes:** By providing learners with relevant and timely feedback, adaptive algorithms empower users to track their progress, set goals, and achieve mastery in their chosen subjects.

### **Implementation Plan:**

- **Conduct Research:** Collaborate with educational experts and data scientists to research and develop adaptive learning models tailored to StudyNotion's user base and content offerings.
- **Data Collection and Analysis:** Implement data collection mechanisms to gather user interaction data and performance metrics. Utilize machine learning techniques to analyse this data and train adaptive algorithms.
- **Integration:** Integrate adaptive learning features seamlessly into the StudyNotion platform, ensuring compatibility with existing functionalities and user interfaces.
- **Testing and Iteration:** Conduct thorough testing and evaluation of the adaptive learning functionality to assess its effectiveness, usability, and impact on user engagement and learning outcomes. Iterate based on user feedback and performance metrics.
- **Rollout and Monitoring:** Gradually roll out the adaptive learning feature to users, closely monitoring its performance and collecting feedback for further refinement and optimization.

### **Expected Outcomes:**

- **Enhanced User Experience:** Adaptive learning algorithms will elevate the StudyNotion experience by providing personalized support and guidance tailored to each user's unique learning journey.
- **Improved Learning Efficacy:** By adapting content and assessments to individual needs, StudyNotion will empower users to achieve their learning goals more effectively, resulting in improved academic outcomes and user satisfaction.

- **Competitive Advantage:** The implementation of adaptive learning technology will differentiate StudyNotion from competitors, positioning it as a leader in personalized education solutions.

Overall, the implementation of adaptive learning algorithms represents a significant opportunity for StudyNotion to enhance its platform and better serve the diverse needs of its user community. By leveraging the power of AI and machine learning, StudyNotion can revolutionize the online learning experience and foster a culture of continuous improvement and personalized growth.

## **Proposed Enhancement for StudyNotion:**

The current StudyNotion platform offers a valuable service for both students and instructors. However, it lacks a system for instructors to monetize their content creation efforts. Implementing a content creator monetization system would incentivize instructors to create high-quality content and contribute to the platform's growth.

### **Proposed Functionality:**

- Subscription Model:**

Instructors can offer courses with a monthly or yearly subscription fee.

Students can subscribe to individual courses or purchase an all-access pass for a discounted rate.

The platform would handle secure payment processing through a payment gateway integration (e.g., Stripe, PayPal).

- Pay-Per-Course Model:**

Instructors can set a one-time purchase price for their courses.

Students can purchase individual courses on a la carte basis.

- Revenue Sharing:**

StudyNotion would take a commission fee on all transactions (e.g., 15-30%).

The remaining revenue would be distributed to the instructor after deducting any processing fees.

### **Benefits:**

- Increased Instructor Engagement:**

Monetization provides a financial incentive for instructors to create high-quality content and maintain their courses.

- **Improved Content Quality:**

Instructors are more likely to invest time and resources in creating in-depth and valuable courses if they can earn from them.

- **Attracting New Instructors:**

A monetization system makes the platform more attractive to potential instructors, leading to a wider variety of courses.

- **Platform Sustainability:**

Revenue generated from course purchases can be used to support platform maintenance, upgrades, and future development.

### **Implementation Considerations:**

- **User Interface Design:**

Integrate a seamless user interface for instructors to manage pricing and subscriptions.

Provide students with a clear way to browse and purchase courses.

- **Security:**

Ensure secure payment processing and data protection for both instructors and students.

- **Tax Implications:**

Research and implement a system that complies with relevant tax regulations for instructors receiving payments.

A content creator monetization system would significantly benefit the StudyNotion platform by attracting and retaining talented instructors, expanding course offerings, and fostering a sustainable business model. By carefully considering the implementation details, StudyNotion can create a win-win situation for both instructors and students.

The StudyNotion project presents a promising ed-tech platform built using the MERN stack (**ReactJS**, **NodeJS**, **ExpressJS**, **MongoDB**). It offers a seamless learning experience for students and a platform for instructors to showcase their expertise.

The technical aspects of the project are well-documented, outlining the system architecture, front-end and back-end functionalities, **API design**, **deployment process**, and potential future enhancements.

The proposed content creator monetization system, including subscription and pay-per-course models, offers a valuable addition to StudyNotion. It incentivizes instructors, improves content quality, attracts new instructors, and fosters platform sustainability. However, limitations like initial user adoption, content pricing, platform competition, and payment processing challenges need to be addressed.

By carefully considering these limitations and implementing the system thoughtfully, StudyNotion can create a win-win situation for both instructors and students. The platform can attract and retain talented instructors, expand course offerings, and establish a sustainable business model, ultimately enhancing the educational experience for its users.

## **Further Considerations:**

- This document serves as a starting point. As the project progresses, additional details and functionalities can be incorporated.
- Conducting user research and gathering feedback from both students and instructors will be crucial for refining the platform's features and monetization strategy.
- Staying updated on emerging technologies in the ed-tech landscape can open doors for further innovation and improvement of the StudyNotion platform.

By continuing development with a focus on user needs and strategic implementation, StudyNotion has the potential to become a valuable resource for learners and educators worldwide.