

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**A PROJECT REPORT ON**

**CODENEXUS: YOUR ULTIMATE PROGRAMMING  
HUB**

SUBMITTED TOWARDS THE  
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

**BACHELOR OF ENGINEERING (Computer Engineering)**

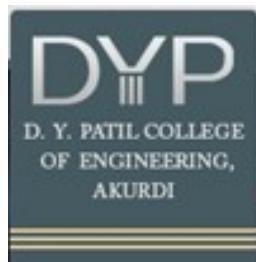
**BY**

Abhijeet Karande  
Chandrakant Shelke  
Shreyash Lohar  
Abhishek Jogdand

Exam No: B400080486  
Exam No: B400080541  
Exam No: B400080493  
Exam No: B400080478

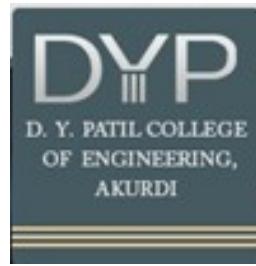
**Under The Guidance of**

Mrs. Dr. Shanthi Guru



**DEPARTMENT OF COMPUTER ENGINEERING**

**D. Y. PATIL COLLEGE OF ENGINEERING, AKURDI,  
PUNE-44**



**D. Y. PATIL COLLEGE OF ENGINEERING, AKURDI,  
PUNE-44  
DEPARTMENT OF COMPUTER ENGINEERING**

**CERTIFICATE**

This is to certify that the Project Entitled

**CODENEXUS: YOUR ULTIMATE PROGRAMMING  
HUB**

Submitted by

|                    |                     |
|--------------------|---------------------|
| Abhijeet Karande   | Exam No: B400080486 |
| Chandrakant Shelke | Exam No: B400080541 |
| Shreyash Lohar     | Exam No: B400080493 |
| Abhishek Jogdand   | Exam No: B400080478 |

is a bonafide work carried out by Students under the supervision of Mrs. Dr. Shanthi Guru and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Mrs. Dr. Shanthi Guru  
Internal Guide  
Dept. of Computer Engg.

Dr. Madhuri A. Potey  
H.O.D  
Dept. of Computer Engg.

Principal  
D. Y. PATIL COLLEGE OF ENGINEERING, AKURDI, PUNE- 44

Signature of Internal Examiner

Signature of External Examiner

## **PROJECT APPROVAL SHEET**

**A Project Titled**

**CODENEXUS: YOUR ULTIMATE PROGRAMMING  
HUB**

Is successfully completed by

Abhijeet Karande  
Chandrakant Shelke  
Shreyash Lohar  
Abhishek Jogdand

Exam No: B400080486  
Exam No: B400080541  
Exam No: B400080493  
Exam No: B400080478

at

**DEPARTMENT OF COMPUTER ENGINEERING  
D. Y. PATIL COLLEGE OF ENGINEERING, AKURDI, PUNE- 44**

**SAVITRIBAI PHULE PUNE UNIVERSITY  
ACADEMIC YEAR 2024-2025**

Mrs. Dr. Shanthi Guru  
Guide  
Dept. of Computer Engg.

Dr. Mrs. M. A. Potey  
H.O.D.  
Dept. of Computer Engg.

## Abstract

CodeNexus is an advanced educational platform built to support both students and administrators in the field of programming education. Designed with a user-friendly interface, it empowers students to engage deeply with programming content by providing access to a diverse library of educational videos. Alongside learning, students can store personal notes directly linked to these videos, ensuring that their study materials are centralized and well-organized. The platform also promotes active learning through quizzes designed for each topic, allowing students to assess their understanding as they progress. Upon mastering topics, students can take certification exams, earning credentials that validate their skills and knowledge. In addition, students have a direct line of communication with administrators via the Contact Us section, fostering a supportive learning environment.

The platform's Manage Programming Languages section enables administrators to easily add, update, or remove programming languages and their learning topics. This flexibility ensures that Codenexus stays current with the ever-evolving landscape of programming technologies. Moreover, the Manage Videos section provides a space where administrators can curate educational content, including video links and descriptions, ensuring that students have access to the latest and most relevant learning materials. A unique feature of Codenexus is its comprehensive certification management system, which allows administrators to create and manage certification exams. They can add, update, or remove exam questions and answers, tailoring the exams to the learning objectives of each programming language. This ensures that certification reflects real-world skills, giving students confidence in their abilities as they enter the job market. The platform's communication system also benefits administrators by giving them the ability to view and respond to student queries, ensuring that any issues or feedback are addressed promptly and effectively.

Ultimately, Codenexus bridges the gap between learners and educators, providing a dynamic and customizable platform for programming education. Its blend of instructional content, self-assessment tools, certification pathways, and administrative flexibility makes it an ideal solution for both educational institutions and independent learners. By streamlining the learning experience and ensuring the platform remains up-to-date, Codenexus positions itself as the ultimate hub for anyone looking to master programming in a structured, supportive environment.

## Acknowledgments

*It gives us great pleasure in presenting the preliminary project report on  
**'CODENEXUS: YOUR ULTIMATE PROGRAMMING HUB'.***

*I would like to take this opportunity to thank my internal guide **Dr. Mrs. Shanthi Guru** for giving me all the help and guidance I needed. I am grateful to them for their kind support. Their valuable suggestions were extremely helpful.*

*I am also grateful to **Dr. Mrs. M. A. Potey**, Head of Computer Engineering Department, D. Y. Patil College of Engineering for his indispensable support, suggestions.*

Chandrakant Shelke  
Abhijeet Karande  
Abhishek Jogdand  
Shreyash Lohar  
(B.E. Computer Engineering)

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Synopsis</b>  | <b>1</b>  |
| 1.1      | Project Title .....  | 1         |
| 1.2      | Project Option.....  | 1         |
| 1.3      | Internal Guide.....  | 1         |
| 1.4      | Technical Keywords (As per ACM Keywords).....                      | 2         |
| 1.5      | Problem Statement.....   | 3         |
| 1.6      | Abstract.....  | 3         |
| 1.7      | Goals and Objectives .....   | 3         |
| 1.8      | Relevant mathematics associated with the Project .....             | 4         |
| 1.9      | Names of Conferences / Journals where papers can be published..... | 5         |
| 1.10     | Review of Conference/Journal Papers supporting Project idea.....   | 5         |
| 1.11     | Plan of Project Execution .....                                    | 7         |
| <b>2</b> | <b>Technical Keywords</b>  | <b>9</b>  |
| 2.1      | Area of Project.....   | 9         |
| 2.2      | Technical Keywords .....   | 9         |
| <b>3</b> | <b>Introduction</b>  | <b>10</b> |
| 3.1      | Project Idea .....   | 10        |
| 3.2      | Motivation of the Project .....                                    | 10        |
| 3.3      | Literature Survey .....  | 10        |
| <b>4</b> | <b>Problem Definition and scope</b>                                | <b>12</b> |
| 4.1      | Problem Statement.....   | 12        |
| 4.1.1    | Goals and objectives .....   | 12        |
| 4.1.2    | Statement of scope.....  | 12        |
| 4.2      | Major Constraints.....   | 13        |
| 4.3      | Methodologies of Problem solving and efficiency issues .....       | 14        |
| 4.4      | Outcome.....   | 15        |
| 4.5      | Applications .....   | 15        |
| 4.6      | Hardware Resources Required .....                                  | 16        |
| 4.7      | Software Resources Required.....                                   | 16        |

|  |           |
|--|-----------|
| <b>5 Project Plan</b>  | <b>17</b> |
| 5.1 Project Estimates .....                                    | 17        |
| 5.1.1 Reconciled Estimates.....                                | 17        |
| 5.1.2 Project Resources.....                                   | 17        |
| 5.2 Risk Management w.r.t. NP Hard analysis.....               | 18        |
| 5.2.1 Risk Identification.....                                 | 18        |
| 5.2.2 Risk Analysis.....                                       | 18        |
| 5.2.3 Overview of Risk Mitigation, Monitoring, Management..... | 18        |
| 5.3 Project Schedule .....                                     | 19        |
| 5.3.1 Project task set.....                                    | 19        |
| 5.3.2 Task network.....  | 21        |
| 5.3.3 Timeline Chart.....                                      | 22        |
| 5.4 Team Organization.....                                     | 23        |
| 5.4.1 Team structure .....                                     | 23        |
| 5.4.2 Management reporting and communication.....              | 23        |
| <b>6 Software requirement specification</b>                    | <b>24</b> |
| 6.1 Introduction.....  | 24        |
| 6.1.1 Purpose and Scope of Document.....                       | 24        |
| 6.1.2 Overview of responsibilities of Developer.....           | 24        |
| 6.2 Usage Scenario .....                                       | 25        |
| 6.2.1 User profiles.....                                       | 25        |
| 6.2.2 Use-cases.....   | 25        |
| 6.2.3 Use Case View.....                                       | 28        |
| 6.3 Data Model and Description.....                            | 28        |
| 6.3.1 Data Description .....                                   | 28        |
| 6.3.2 Data objects and Relationships .....                     | 29        |
| 6.4 Functional Model and Description .....                     | 30        |
| 6.4.1 Data Flow Diagram.....                                   | 30        |
| 6.4.2 Activity Diagram .....                                   | 31        |
| 6.4.3 Non-Functional Requirements .....                        | 32        |
| 6.4.4 State Diagram .....                                      | 33        |
| 6.4.5 Design Constraints.....                                  | 33        |
| 6.4.6 Software Interface Description .....                     | 34        |
| <b>7 Detailed Design Document using Appendix A and B</b>       | <b>35</b> |
| 7.1 Introduction.....  | 35        |
| 7.2 Architectural Design.....                                  | 35        |
| 7.3 Data design (using Appendices A and B).....                | 36        |
| 7.3.1 Internal software data structure .....                   | 36        |

|   |                                       |           |
|---|---------------------------------------|-----------|
| 7.3.2   | Global data structure.....            | 36        |
| 7.3.3   | Temporary data structure .....        | 37        |
| 7.3.4   | Database description.....             | 38        |
| 7.4   | Component Design .....                | 39        |
| 7.4.1   | Class Diagram.....                    | 41        |
| <b>8</b>  | <b>Project Implementation</b>         | <b>42</b> |
| 8.1   | Introduction.....                     | 42        |
| 8.2   | Tools and Technologies Used .....     | 42        |
| 8.3   | Methodologies/Algorithm Details ..... | 43        |
| 8.3.1   | Algorithm 1/Pseudo Code .....         | 44        |
| 8.3.2   | Algorithm 2/Pseudo Code .....         | 45        |
| <b>9</b>  | <b>Software Testing</b>               | <b>46</b> |
| 9.1   | Type of Test Cases Used.....          | 46        |
| 9.2   | Test Cases .....                      | 47        |
| <b>10</b>   | <b>Results</b>                        | <b>48</b> |
| 10.1  | Screen shots .....                    | 53        |
| 10.2  | Outputs.....                          | 58        |
| <b>11</b>   | <b>Deployment and Maintenance</b>     | <b>59</b> |
| 11.1  | Installation and un-installation..... | 59        |
| 11.2  | User help .....                       | 60        |
| <b>12</b>   | <b>Conclusion and Future Scope</b>    | <b>61</b> |
| <b>Appendix References</b>  |                                       | <b>63</b> |
| <b>Appendix Laboratory assignments on Project Analysis of Algorithmic Design</b>                    |                                       | <b>64</b> |
| <b>Appendix Laboratory assignments on Project Quality and Reliability Testing of Project Design</b> |                                       | <b>66</b> |
| <b>Appendix Project Planner</b>   |                                       | <b>69</b> |
| <b>Appendix Reviewers Comments of Paper Submitted</b>   |                                       | <b>71</b> |
| <b>Appendix Plagiarism Report</b>   |                                       | <b>78</b> |
| <b>Appendix Information of Project Group Members</b>  |                                       | <b>79</b> |

# List of Figures

| Sr. No | Figure                        | Pg. No. |
|--------|-------------------------------|---------|
| 5.3.3  | Time Line Chart-----          | 22      |
| 6.2.3  | Use Case Diagram-----         | 28      |
| 6.4.2  | Activity Diagram-----         | 31      |
| 6.4.4  | State Transition Diagram----- | 33      |
| 7.2.1  | Deployment Diagram-----       | 35      |
| 7.3.3  | Architecture diagram -----    | 37      |
| 7.4.1  | DFD Diagram-----              | 40      |
| 7.4.2  | Class Diagram-----            | 41      |

# List of Tables

| Sr.No   | Table                             | Page No. |
|---------|-----------------------------------|----------|
| 1.10    | Literature Survey.....            | 5        |
| 4.6     | Hardware Requirements.....        | 16       |
| 5.2.2.1 | Risk Table.....                   | 18       |
| 5.2.2.2 | Risk Probability Definitions..... | 18       |
| 5.3.1.1 | Risk Impact Definitions.....      | 19       |
| 5.3.1.2 | Risk 1.....                       | 19       |
| 5.3.1.3 | Risk 2.....                       | 20       |
| 5.3.1.4 | Risk 3.....                       | 20       |
| 5.4     | Team Structure.....               | 23       |
| 6.2.3   | Use Cases.....                    | 28       |
| 9.2.1   | Unit Test Cases.....              | 46       |
| 9.2.2   | Integration Test Cases.....       | 47       |
| 9.2.3   | System Test Cases .....           | 47       |
|         | IDEA Matrix.....                  | 64       |
|         | Task Breakdown.....               | 69       |

# Synopsis

## 1.1 Project Title

CodeNexus : Your Ultimate Programming Hub.

## 1.2 Project Option

Internal Project

## 1.3 Internal Guide

Dr. Mrs. Shanthi Guru

## 1.4 Technical Keywords

### 1. H. Information Systems

#### (a) H.3 Information Storage and Retrieval

##### I. H.3.3 Information Search and Retrieval

- Content-based Filtering
- Personalized Recommendations

#### (b) H.3.5 Online Information Services

- Web-based Services
- Digital Libraries
- E-learning Platforms

### 2. H.5 Information Interfaces and Presentation (e.g., HCI)

#### (a) H.5.2 User Interfaces

- Web-based Interaction
- Personalization
- User Experience (UX)
- Accessibility Design

**(b) H.5.3 Group and Organization Interfaces**

- Collaborative Learning
- Asynchronous Learning Systems

**3. K.3 Computers and Education****(a) K.3.1 Computer Uses in Education**

- Distance Learning
- Computer-assisted Instruction (CAI)
- Certification Systems
- Interactive Learning Environments

**4. K.6 Management of Computing and Information Systems****(a) K.6.5 Security and Protection**

- i. Authentication (JWT)
- ii. Secure Password Hashing (bcrypt)
- iii. Role-Based Access Control (RBAC)
- iv. Data Privacy Management

**5. I. Computing Methodologies****(a) I.2.6 Learning**

- i. Machine Learning (if future recommendation features planned)
- ii. Adaptive Learning Systems

**6. K.4 Computers and Society****(a) K.4.3 Organizational Impacts**

- i. Educational Institutions
- ii. Digital Transformation in Education

**7. D. Software****(a) D.2 Software Engineering**

- i. Software Maintenance
- ii. Web Application Development

## 1.5 Problem Statement

How can CodeNexus develop a unified, scalable e-learning platform that consolidates essential programming resources such as AI-driven support, interactive content, video tutorials, and personalized quizzes, while dynamically adapting to individual learning paths. The platform must maintain high engagement, efficient resource management, and seamless integration of new features, ensuring optimal performance and a user-friendly interface for programmers at all skill levels.

## 1.6 Abstract

CodeNexus aims to establish a comprehensive e-learning platform tailored to the needs of both novice and experienced programmers. This platform merges cutting-edge AI-driven technologies with interactive learning content to offer a seamless, user-friendly educational experience. Key features include an AI-powered chatbot for real-time support, custom note storage for personalized learning, video tutorials, quizzes, and certification opportunities. Moreover, CodeNexus integrates robust administrative tools that allow for efficient content management and continuous user support. The goal is to create a holistic and engaging learning environment that enhances educational efficiency while fostering skill development through personalized, on-demand resources. Through these efforts, CodeNexus addresses the increasing demand for dynamic, scalable, and accessible online programming education platforms. The project's detailed roadmap and system architecture ensure the development of a scalable solution that meets the diverse needs of learners and administrators alike, fostering a continuous and enriched learning journey .

## 1.7 Goals and Objectives

### **Implement a Structured Learning Path for Users**

Develop a learning environment that provides users with guided access to programming languages through video tutorials, quizzes, and certification exams. Enable students to progress step-by-step and keep track of their learning through personalized note-taking features.

### **Enhance User Experience with Interactive Assessments**

Offer topic-specific quizzes and certification exams that allow students to evaluate their understanding and receive real-time feedback on their performance. Incorporate automatic grading and result tracking to help users monitor their learning journey effectively.

### **Provide Seamless Administrative Control and Content Management**

Allow administrators to manage programming languages, video content, quizzes, and exams, ensuring they can add, update, or delete educational material as needed. Provide tools for managing user accounts and responding to student queries, ensuring a well-maintained and supportive learning environment.

### **Facilitate User-Admin Communication and Support**

Develop a direct communication system between users and administrators through the Contact Us section. Ensure that student concerns, inquiries, and feedback are handled efficiently, fostering a supportive educational experience.

## 1.8 Relevant mathematics associated with the Project

### System Description:

#### Input:

- **User Data:** This includes information about the user's programming preferences, learning history, performance metrics, and interaction data within the platform. These inputs are gathered from user activity logs, quizzes, and code submissions.
- **Content Data:** The platform processes data from learning resources such as video tutorials, coding exercises, and educational content to extract features like topic, difficulty level, and teaching style.

#### Output:

- **Personalized Learning Path:** Based on the user and content data, the system recommends customized learning paths and coding exercises. These recommendations are made using hybrid filtering techniques (collaborative and content-based filtering) to match users with relevant educational materials.

#### Data Structures:

- **Matrices:** User performance and interaction data are stored in matrices that represent the user's learning history and content interactions. This could include a user-content interaction matrix where each entry represents the time spent, quiz results, or code submissions.
- **Feature Vectors:** Used to store extracted features from user data (e.g., skill level, preferences) and content data (e.g., topic coverage, difficulty). These vectors represent important attributes needed for recommending appropriate content.
- **Content Database:** Stores structured data about the educational materials such as videos, quizzes, and coding challenges, along with metadata like topic, duration, and user ratings.

#### Classes:

- **User:** Stores information related to the user's profile, including learning preferences, skill level, quiz scores, and interaction history.
- **Content Analysis:** Handles feature extraction from the educational materials, processing attributes such as complexity, teaching method, and user feedback.
- **Content Database:** Manages storage and retrieval of educational content for recommendation.

## 1.9 Name of Conference/Journal where Paper is Published

1. C.Shelke, S.Lohar, A.Jogdand, A.Karande, Dr.Shanthi Kumarguru, "CodeNexus : Your Ultimate Programming Hub" in Proc. NCFCAI 2024 - National Conference on Frontier in Computing and Artificial Intelligence, ISTE Approved, 2024.

## 1.10 Review of Conference/Journal Papers supporting Project idea

Table 1.10 Literature Survey

| Sr. No | Title  | Author(s)                          | Date | Methodology  |
|--------|--|------------------------------------|------|--|
| 1.     | Distance Learning System Learning Programming              | Mr. B. A. Ali, Dr. R. J. Anand     | 2020 | Focuses on the design and development of a distance learning system. Methodology includes user-centered design, and technologies used are web-based platforms. Algorithms include matching learning styles with content. |
| 2.     | Research Paper on E-Learning Application design features   | Dr.Veeramanic-kam Murugappan       | 2016 | Discusses the impact of online learning on student performance. Methodology includes surveys and data analysis. Uses statistical algorithms for performance evaluation.  |
| 3.     | E-Learning Based Web Programming Course                    | Dony Novaliendry,Asrul Hudda, etc. | 2021 | Covers the creation of an e-learning platform for teaching web programming. Methodology includes blended learning and interactive modules. Algorithms for adaptive learning are used.<br>.                               |
| 4.     | A Web Based Learning Environment for Programming Languages | Mr.Mustafa Yagci                   | 2017 | Aimed to capture Students Positive and Negative Feedback regarding blended learning environment.Responses collected from the students were analyzed using qualitative analysis and Overall feedback was on positive side |

| Sr. No | Title  | Author(s)   | Date | Methodology   |
|--------|--|---|------|---|
| 5.     | E-Learning in Higher Education:Focus Groups and survey among Students                                      | Lucija Majcen   | 2016 | Conducted two focus groups to explore students perception, attitudes, and experiences with e-learning. Administered a survey to 104 students in central Europe to assess their readiness for e-learning and willingness to pay  |
| 6.     | E-Learning Effectiveness in Higher Education   | Monika Takru, Tapan Kumar Jha                               | 2019 | The study uses a quantitative approach, surveying students in higher education to assess the effectiveness of e-learning across gender, course, and city. Data is analyzed statistically to identify patterns and differences in how e-learning impacts various student groups. |
| 7.     | E-Learning in Higher Education: Focus Groups and Survey Among Students in Central Europe                   | Marko Tuševljak, Lucija Majcen, et al.                      | 2016 | Mixed-method research using focus groups and surveys to assess student perception and readiness for e-learning in higher education in Central Europe (EJ1131833).   |
| 8.     | "Educational Website (E-Learning)"   | Dimple Patil, Siddhi Rane, Vanshika Waghela, Sandeep Mishra | 2022 | MERN stack implementation for a scalable e-learning platform with features like course hosting and student access (IRJET-V9I5235)   |
| 9.     | Research on E-Learning Application Design Features Using Cloud Computing and Software Engineering Approach | M.R.M. Veeramanickam, M. Mohanapriya                        | 2016 | Application design model using cloud-based architecture and Software Engineering principles to enhance e-learning efficiency (ICICES2016).  |
| 10.    | "Developing Virtual Web as Learning Media for Web Programming in SMK"                                      | Dedi Sorongan, Yuliana Mose, Reonaldy A Berikang            | 2022 | Development and evaluation using DDD-E model for web programming e-learning, with testing phases for usability and effectiveness (PDF-4)  |

## 1.11 Plan of Project Execution

### Phase 1: Project Initiation and Requirement Gathering (Week 1-2)

- Define the project objectives and establish the scope for Codenexus.
- Collect and document functional and non-functional requirements from stakeholders.
- Identify necessary tools, technologies, and resources needed for development (e.g., database, programming language, framework).
- **Milestone:** Completion of project proposal and requirement documentation.

### Phase 2: System Design and Architecture (Week 3-4)

- Develop system architecture diagrams, including components for user management, content delivery, and assessments.
- Design database schemas for user data, programming language content, quiz results, and certification tracking.
- Design the frontend and backend structure for managing educational content, video integration, and quizzes.
- **Milestone:** Finalization of system design and architecture.

### Phase 3: Backend Development (Week 5-7)

- Set up the database to manage user data, programming language topics, and video links.
- Implement the backend features, including user authentication, note-taking functionality, and video access.
- Develop APIs for quiz management, certification exams, and user progress tracking.
- **Milestone:** Completion of backend functionalities and API integration.

### Phase 4: Quiz and Certification Module Development (Week 8-9)

- Develop the quiz system with real-time feedback for users based on their performance in programming topics.
- Implement the certification module for exams, allowing users to take tests and receive certificates upon passing.
- Integrate auto-grading for quiz and exam answers, along with result tracking.
- **Milestone:** Completion of quiz and certification module.

### Phase 5: Admin Panel and Content Management (Week 10-12)

- Build an admin panel for managing programming languages, video content, quizzes, and user accounts.
- Enable admins to add, update, or delete programming languages and related learning materials.
- Integrate functionality for managing quizzes, certification questions, and responses.
- **Milestone:** Completion of admin panel and content management system.

### Phase 6: Frontend Development and UI (Week 13-14)

- Design and develop the frontend interface for both students and administrators, ensuring a user-friendly experience.
- Implement features such as video access, note-taking, and quiz interaction for students.
- Design the admin interface for managing content and users.
- **Milestone:** Completion of frontend UI and interaction features.

#### **Phase 7: Testing and Quality Assurance (Week 15-16)**

- Conduct unit, integration, and system testing across all modules (user features, admin panel, quizzes, certifications).
- Perform stress testing on user management and content delivery systems to ensure scalability.
- Debug any issues found during testing to ensure reliable functionality.
- **Milestone:** Successful completion of testing phases.

#### **Phase 8: Deployment and User Feedback (Week 17-18)**

- Deploy the platform on the chosen platform (e.g., web).
- Gather feedback from a test group of users to refine the user experience and identify any further improvements.
- Monitor the system's performance and adjust features based on feedback.
- **Milestone:** Deployment and collection of user feedback.

#### **Phase 9: Final Documentation and Presentation (Week 19)**

- Prepare comprehensive final documentation covering system design, functionality, and testing outcomes.
- Create a detailed presentation and demonstration of Codenexus for stakeholders.
- **Milestone:** Completion of final documentation and project presentation.

#### **Phase 10: Project Handover (Week 20)**

- Finalize all project deliverables, including source code, documentation, and deployment instructions.
- Handover the project for future maintenance, updates, or enhancements.
- **Milestone:** Project handover completed and ready for future use.

# Technical Keywords

## 2.1 Area of Project

**E-Learning Platforms:** Codenexus is centered on providing an interactive platform for learning programming languages through a structured approach, offering students access to video tutorials, coding assessments, quizzes, and certification exams. The platform helps users track their progress and supports personalized learning experiences with tailored educational content..

## 2.2 Technical Keywords

### 1. Computing Methodologies

#### 1.1 Software Development

1.1.1 Learning Management Systems

### 2. Educational Platforms

#### 2.1 Applications

- 2.1.1.1 Student Performance Tracking
- 2.1.1.2 Automated Quiz Evaluation

#### 2.2 User Experience

- 2.2.1 Content Delivery Systems
- 2.2.2 Personalized Learning Paths

### 3. Information Systems

#### 3.1 Information Interfaces and Presentation

3.1.1 User Interfaces (UI/UX)

#### 3.2 Information Storage and Retrieval

- 3.2.1 Course and Video Content Search
- 3.2.2 Student Progress Data Management

### 4. Applied Computing

#### 4.1 Computer Application

- 4.1.1.1 E-Learning Platforms
- 4.1.1.3 Distance Learning Tools

# Introduction

## 3.1 Project Idea

The primary goal of CodeNexus is to create an all-in-one platform that enhances programming education by offering real-time code evaluation, personalized learning paths, and collaboration tools. CodeNexus aims to provide a seamless, engaging, and effective learning experience for students and educators alike, bridging the gap between theoretical concepts and practical coding skills.

## 3.2 Motivation of the Project

The motivation behind Codenexus stems from the growing need to bridge the gap between theoretical programming education and practical, hands-on experience. In traditional learning environments, students often struggle to effectively apply programming concepts, while instructors face challenges in providing timely feedback and assessing student progress. The lack of interactive, real-world coding experiences hinders students' ability to grasp complex topics, leaving gaps in their understanding. Additionally, online education platforms often fall short in offering personalized learning paths and secure assessment systems, which are crucial for helping students succeed in a self-paced, remote environment.

As the demand for online learning continues to rise, Codenexus aims to address these challenges by providing an engaging, secure, and comprehensive platform for programming education. It offers personalized learning experiences through video tutorials, quizzes, and hands-on coding assessments, allowing students to track their progress and apply what they've learned in real time. For instructors, the platform simplifies the process of content management, assessment, and feedback, making it easier to deliver high-quality education. Ultimately, Codenexus seeks to create an environment where both students and educators can thrive in an ever-evolving digital learning landscape.

## 3.3 Literature Survey

1. **Ali, B. A., and Anand, R. J. (2020):** "Distance Learning System for Learning Programming." This paper focuses on the design and development of a distance learning system for programming using a user-centered design approach. Web-based platforms serve as the primary technology, with algorithms that match learning styles to content to enhance engagement.

2. **Murugappan, V. (2016)**: "Research on E-Learning Application Design Features." This study examines the effects of online learning on student performance, employing surveys and data analysis. Statistical algorithms are used to evaluate performance, providing insights into how e-learning affects outcomes.
3. **Novaliendry, D., Hudda, A., et al. (2021)**: "E-Learning-Based Web Programming Course." This research covers the development of an e-learning platform for teaching web programming, using blended learning methods and interactive modules. Adaptive learning algorithms enhance the customization of content delivery.
4. **Yagci, M. (2017)**: "A Web-Based Learning Environment for Programming Languages." This paper investigates student feedback on a blended learning environment for programming languages. Using qualitative analysis, the study finds overall positive responses from students regarding their experience.
5. **Majcen, L. (2016)**: "E-Learning in Higher Education: Focus Groups and Surveys Among Students." This research explores students' perceptions, attitudes, and experiences with e-learning through focus groups and surveys, gathering data from 104 students in Central Europe to assess e-learning readiness and willingness to invest.
6. **Takru, M., and Jha, T. K. (2019)**: "E-Learning Effectiveness in Higher Education." This paper uses a quantitative approach to survey students in higher education, analyzing statistical data to understand e-learning effectiveness across gender, course, and city. It identifies patterns in how different groups experience e-learning.
7. **Tuševljak, M., Majcen, L., et al. (2016)**: "E-Learning in Higher Education: Focus Groups and Survey Among Students in Central Europe." This study uses a mixed-method approach, combining focus groups and surveys to assess student perception and readiness for e-learning in higher education across Central Europe (EJ1131833)
8. **Patil, D., Rane, S., Waghela, V., and Mishra, S. (2022)**: "Educational Website (E-Learning)." This paper presents a scalable e-learning platform implemented using the MERN stack, featuring course hosting capabilities and student access management (IRJET-V9I5235).
9. **Veeramanickam, M. R. M., and Mohanapriya, M. (2016)**: "Research on E-Learning Application Design Features Using Cloud Computing and Software Engineering Approach." This paper introduces an e-learning application design model based on cloud architecture and Software Engineering principles to improve system efficiency (ICICES2016).
10. **Sorongan, D., Mose, Y., and Berikang, R. A. (2022)**: "Developing Virtual Web as Learning Media for Web Programming in SMK." This study details the development and evaluation of a web programming e-learning platform using the DDD-E model, with phases to test usability and effectiveness (PDF-4).

# Problem Definition and Scope

## 4.1 Problem Statement

In the current landscape of programming education, both traditional classroom settings and many online platforms often fail to deliver a comprehensive, practical learning experience. Students commonly face challenges such as delayed or insufficient feedback on coding assignments, which hampers their ability to correct mistakes and learn effectively. Additionally, the absence of personalized learning paths means that learners with varying skill levels must follow the same content, leading to either boredom or confusion, depending on their proficiency. Many platforms also lack real-time code evaluation systems, limiting students' exposure to practical problem-solving and instant validation of their work—an essential component of mastering programming skills..

### 4.1.1 Goals and Objectives

- To create a comprehensive platform that delivers interactive, real-time learning experiences for programming students, with features like video tutorials, quizzes, and certification exams tailored to individual progress.
- To provide a seamless content management system for administrators, enabling them to efficiently manage programming languages, educational content, quizzes, and certification exams without technical complexity.
- To implement personalized learning paths by tracking user performance and offering real-time feedback, enhancing student engagement and comprehension through gamified quizzes and assessments.
- To ensure a secure and scalable platform using cloud-based infrastructure, offering reliable user management, content delivery, and examination systems that maintain the integrity of assessments and user data.
- To deliver an intuitive and user-friendly interface for both students and administrators, ensuring that the platform is accessible, easy to navigate, and provides a smooth educational experience.

### 4.1.2 Statement of scope

The objective of CodeNexus is to provide a robust e-learning platform that enables learners to develop and enhance their programming skills through real-time coding evaluations, personalized learning modules, and secure assessment systems. The platform aims to address key challenges in online education, such as instant feedback, collaboration, academic integrity, and scalability, creating a seamless and secure learning experience.

#### 1. Input:

- **User Data:** Input from students including login credentials, programming language preferences, quiz answers, and progress tracking data.

- **Admin Data:** Input from administrators such as programming language content, video links, quiz questions, and certification exam data.

## 2. Bounds on Input:

- **User Data:** Requires valid user login credentials and appropriate answers for quizzes and certification exams. Incorrect formats or incomplete data may result in input rejection.
- **Admin Data:** Admin inputs must follow specific formats, such as video URLs and structured quiz/exam formats. Incorrectly formatted data will not be accepted.

## 3. Input Validation:

- **User Data:** The system verifies that user input is valid, ensuring proper authentication, completion of required fields in quizzes, and accurate tracking of progress.
- **Admin Data:** The system checks for proper formatting of programming language content, video links, and quiz structures before accepting the data.

## 4. Input Dependency:

- **User Data:** The system's output depends on accurate user input, including quiz results and exam submissions. A lack of input or incomplete input could hinder accurate progress tracking and certification generation.
- **Admin Data:** The functionality of the platform depends on accurate content provided by administrators, including properly structured lessons, quizzes, and exams.

## 5. I/O State Diagram:

- The system continuously processes user and admin inputs, managing content delivery (videos, quizzes) and outputting user progress, certifications, and updated content for both students and administrators.

## 6. Major Inputs and Outputs:

- **Inputs:** User login credentials, quiz answers, programming preferences, and admin inputs for content and exam data.
- **Outputs:** Personalized learning paths, quiz results, certificates, and progress reports for users; updated content and user management data for administrators.

## 4.2 Major Constraints

### 1. Hardware Constraints:

- The platform should function smoothly across a variety of devices, including desktops, laptops, and tablets. Devices with low performance or outdated hardware may experience delays when accessing video tutorials or handling coding challenges.

## 2. Real-Time Processing:

- Codenexus requires real-time processing for quizzes, assessments, and progress tracking. The platform must ensure that quiz results and certification generation happen without noticeable delays, providing a smooth and responsive experience for users.

## 3. Network and Latency Conditions:

- The quality of the user experience may be affected by unstable internet connections, especially during video streaming and real-time assessments. A poor network connection could result in interruptions in content delivery or difficulty in submitting quizzes and assessments.

## 4. Computational Constraints:

- The system's backend must efficiently handle multiple simultaneous users, especially when managing video content, real-time quiz assessments, and certification exams. Overloading the server with too many concurrent requests can result in slower responses and diminished performance.

## 5. Privacy and Security:

- User data privacy is critical. Codenexus will not store sensitive information beyond necessary user credentials and progress data. Secure handling of quiz and exam data is essential to maintain the integrity of assessments and certification processes. Strict encryption protocols should be in place to protect user information.

## 4.3 Methodologies of Problem solving and efficiency issues

### Content Management for Programming Languages:

- **Method:** The system uses a robust database structure to store programming languages, related tutorials, and quiz questions. Administrators can manage the content efficiently through a well-organized backend interface. Updates to programming languages and content are made via CRUD (Create, Read, Update, and Delete) operations, allowing flexible content management.
- **Efficiency:** As the number of programming languages and content increases, database queries can slow down. To address this, database indexing and query optimization techniques (e.g., caching frequently accessed content) will be implemented to ensure fast retrieval of data.

### Quiz and Assessment Management:

- **Method:** The platform leverages an automated quiz system that validates and scores user responses in real time. A well-defined question bank stores quiz questions, and

- results are calculated immediately after submission. The system uses both multiple-choice questions and code-based assessments.
- **Efficiency:** To avoid delays during large-scale assessments, especially when multiple users submit quizzes simultaneously, load balancing and parallel processing techniques will be used. Efficient algorithms will be employed to ensure quick evaluation, even for complex code-based quizzes.

#### **Real-Time User Management and Certification:**

- **Method:** The platform manages user accounts, quiz scores, and certification records in real time. As users complete quizzes and exams, the system verifies and generates certificates instantly. A dynamic user management system allows administrators to monitor progress and address issues on the fly.
- **Efficiency:** Handling a large number of user certifications and records in real time can lead to system delays. To mitigate this, batch processing for certification generation and optimized data storage methods, such as cloud-based solutions, will be utilized to ensure smooth and timely operations.

## **4.4 Planned Outcome**

The planned outcome of the CodeNexus project is to deliver a fully functional and user-friendly web-based platform that revolutionizes programming education through a combination of real-time code evaluation, personalized learning, and secure assessments. The platform will enable students to write and test code directly within the system, receiving instant feedback that reinforces learning and corrects mistakes as they occur. It will also incorporate adaptive learning paths that adjust to each user's performance, ensuring that content is tailored to individual skill levels and learning pace.

## **4.5 Applications**

### **1. Online Programming Education Platforms**

CodeNexus can be used by educational institutions and e-learning providers to offer structured programming courses with real-time code evaluation, personalized learning paths, and automatic assessments, enhancing student engagement and learning outcomes.

### **2. Skill Development and Certification Portals**

The platform can serve as a training and certification tool for individuals looking to upskill in programming. With secure assessments and instant feedback, learners can practice and earn certifications based on demonstrated coding proficiency.

### **3. Technical Training for Companies and Bootcamps**

Organizations and coding bootcamps can use CodeNexus to train employees or participants in specific programming languages or technologies, track their progress, and evaluate their readiness through practical coding tests.

#### 4. Coding Practice and Competitive Preparation

CodeNexus can act as a practice hub for students preparing for technical interviews or competitive programming contests, offering a wide range of coding challenges, time-bound tasks, and feedback-driven learning.

### 1.12 Hardware Resources Required

Table 4.6 Hardware Requirements

| Sr. No. | Parameter             | Minimum Requirement                             | Justification  |
|---------|-----------------------|---|--|
| 1.      | Computer/Device       | Desktop, laptop, or tablet with internet access | A standard computing device is necessary for accessing the platform, watching tutorials, and taking quizzes. |
| 2.      | Internet Connection   | High-speed internet (10 Mbps or higher)         | A stable connection is essential for streaming videos, real-time quizzes, and Content uploads/downloads.     |
| 3.      | Processing Unit (CPU) | Multi-core CPU (2.0 GHz or higher)              | Required to handle simultaneous tasks like video streaming, quiz processing, And content management.         |
| 4.      | Memory (RAM)          | 4GB RAM or higher                               | Supports smooth navigation and allows users to watch videos and Take quizzes without performance issues.     |
| 5.      | Storage               | 128GB SSD or higher                             | Fast storage for quick loading of content, caching progress, and Storing quiz results.                       |

### 1.13 Software Resources Required

#### 1. Languages and Frameworks

HTML/CSS/Bootstrap : Frontend structure and styling

JavaScript : Frontend interactivity (quizzes, dynamic forms)

PHP : Backend logic, CRUD operations, session handling

MySQL : Database management for users, quizzes, notes

Python (Flask) : AI features (chatbot, summarizer using OpenAI)

#### 2. Libraries & APIs

Chart.js : Data visualization (quiz results, progress)

Flask (Python) : Lightweight API server for AI-related endpoints

Flask-CORS : To enable cross-origin requests from frontend

OpenAI API : Chatbot, text summarizer, or other AI features

PHPMailer (optional) : Email/contact form handling in PHP

#### 3. Hosting & Deployment (Local)

Localhost (via XAMPP) : Backend & frontend hosting during dev  
Python local server : Hosting Flask microservice for AI  
phpMyAdmin : Managing MySQL database

#### 4. Optional Tools (Future Enhancements)

Netlify/GitHub Pages : If you decide to move frontend online  
Firebase Authentication : For modern login/OTP/email auth  
Docker : To containerize backend + AI APIs (if needed)  
Expo (if mobile) : For future mobile version of CodeNexus

## Project Plan

### 5.1 Project Estimates

The project follows a Waterfall model, where each phase—requirements gathering, design, implementation, testing, deployment, and maintenance—is completed before moving to the next. The following sections provide the reconciled estimates for the project.

#### 5.1.1 Reconciled Estimates

- **Cost Estimate**
  - Paper publication cost: Rs. 1000/-
  - Documentation: Rs. 2000/-
  - Total: Rs. 3000/-
- **Time Estimates**

Phase 1 - Requirements and Design: 4 weeks.

Phase 2 - Model Development: 6 weeks.

Phase 3 - System Integration and Testing: 5 weeks.

Phase 4 - Deployment and Maintenance: 2 weeks.

Total Estimated Time: 17 weeks.

#### 5.1.2 Project Resources

- **People:** Backend and frontend developers, database administrators, UI/UX designers, and project managers.
- **Hardware:** Cloud servers (e.g., AWS/Google Cloud) for hosting, and local development machines with Intel Core i3, 40GB hard disk, and 2GB RAM.
- **Software:** React.js (frontend), Php, Express, Nodejs (backend), SQL for databases, and HTML/.CSS for web design
- **Tools:** Git for version control, Jira/Trello for task management, Jenkins for CI/CD, Docker for containerization, and PostgreSQL/MySQL for database management.
- **Other Resources:** Web browsers (Chrome, Safari, Edge), coding libraries, quiz frameworks, and external APIs for user authentication.

## 5.2 Risk Management w.r.t. NP Hard analysis

This section identifies project risks and strategies to mitigate them.

### 5.2.1 Risk Identification

Risks were identified by reviewing the scope, requirements, and schedule, and answering the following questions:

- Are end-users involved and enthusiastic about the product?
- Are requirements clearly defined and stable?
- Does the team have the necessary skills and size?

### 5.2.2 Risk Analysis

The identified risks for the project, along with their probability and impact, are outlined below:

Table 5.2.2.1 Risk Table

| ID | Risk Description                      | Probability | Impact   |         |         |
|----|---------------------------------------|-------------|----------|---------|---------|
|    |                                       |             | Schedule | Quality | Overall |
| 1  | Lack of data (quizzes, etc.)          | Medium      | High     | High    | High    |
| 2  | Integration issues with external APIs | Low         | High     | Medium  | High    |

Table 5.2.2.2 Risk Probability definitions

| Probability | Value                        | Description |
|-------------|------------------------------|-------------|
| High        | Probability of occurrence is | > 75%       |
| Medium      | Probability of occurrence is | 26 – 75%    |
| Low         | Probability of occurrence is | < 25%       |

### 5.2.3 Overview of Risk Mitigation, Monitoring, Management

- **Risk 1: Server overload due to increasing users** – As the platform grows, it may struggle to handle high traffic, leading to slow performance or crashes. **Mitigation:** Implement load balancing and leverage scalable cloud infrastructure like AWS or Google Cloud to ensure the platform can grow efficiently.

- **Risk 2: Integration issues with third-party APIs** – APIs used for authentication or quizzes could face compatibility issues or downtime. **Mitigation:** Identify backup APIs and conduct rigorous, frequent testing to minimize disruptions and ensure consistent functionality.

- **Risk 3: Data security vulnerabilities** – Storing user data (e.g., notes, quiz results) involves privacy risks. **Mitigation:** Use data encryption, regular security audits, and adhere to data protection standards to safeguard user information and maintain platform integrity.

## 5.3 Project Schedule

### 5.3.1 Project task set

The major tasks in the project are as follows:

- Task 1: Gathering requirements and preparing the dataset.
- Task 2: Developing an AI-driven interactive learning model.
- Task 3: Optimizing and training the adaptive quiz and certification module.
- Task 4: Integrating external APIs (e.g., Coursera, Udemy) for additional learning resources.
- Task 5: Testing, debugging, and deployment of the e-learning platform

Table 5.3.1.1 Risk Impact definitions

| Impact    | Value   | Description   |
|-----------|---------|---|
| Very high | > 10%   | Schedule impact or Unacceptable quality   |
| High      | 5 – 10% | Schedule impact or Some parts of the project have low quality   |
| Medium    | < 5%    | Schedule impact or barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated |

Table 5.3.1.2 Risk 1

|                  |                         |
|------------------|-------------------------|
| Risk ID          | 1                       |
| Risk Description | Lack of sufficient data |
| Category         | Dataset Availability.   |

|             |  |
|-------------|--|
| Source      | During dataset collection phase.           |
| Probability | Medium                                     |
| Impact      | High (5 – 10%)                             |
| Response    | Mitigate                                   |
| Strategy    | Expand data sources, use data augmentation |
| Risk Status | Identified                                 |

Table 5.3.1.3 Risk 2

|                  |  |
|------------------|--|
| Risk ID          | 2  |
| Risk Description | Requirement inconsistencies                  |
| Category         | Technical Requirements                       |
| Source           | Hardware requirements during model training. |
| Probability      | Low  |
| Impact           | High (5-10%)                                 |
| Response         | Mitigate                                     |
| Strategy         | Ensure access to high-performance hardware   |
| Risk Status      | Identified                                   |

Table 5.3.1.4 Risk 3

|                  |   |
|------------------|---|
| Risk ID          | 3   |
| Risk Description | Data privacy concerns                               |
| Category         | Security  |
| Source           | During testing phase and use of external APIs.      |
| Probability      | Medium  |
| Impact           | Very High (>10%)                                    |
| Response         | Mitigate  |
| Strategy         | Implement encryption, anonymization, and compliance |
| Risk Status      | Identified  |

### 5.3.2 Task network

- **User Interaction Layer**
  - **User Registration/Login**
    - Task: User authentication
    - Dependencies: Frontend and backend integration, database for user data
  - **Dashboard Access**
    - Task: Display user-specific content
    - Dependencies: Data fetching from backend, UI/UX design for customization
- **Core Programming Hub Features**
  - **Code Editor**
    - Task: Provide coding environment for users
    - Dependencies: Integration with code execution engines, syntax highlighting
  - **Coding Assistance**
    - Task: Real-time code suggestions
    - Dependencies: Machine learning model integration for suggestions, backend communication
- **Backend Integration**
  - **API Development**
    - Task: Create RESTful APIs for frontend-backend communication
    - Dependencies: Flask/Django, API security measures
  - **Database Management**
    - Task: Manage user data, project files, emotion states
    - Dependencies: MongoDB or SQL databases, data schemas
- **Testing and Quality Assurance**
  - **Unit Testing**

- Task: Test individual components and features
- Dependencies: Automated testing tools (e.g., Jest, PyTest)
- **Integration Testing**
  - Task: Ensure seamless integration between frontend, backend, and ML models
  - Dependencies: Test environments, CI/CD pipelines (e.g., Jenkins)
- **User Acceptance Testing (UAT)**
  - Task: Test system from the end-user perspective
  - Dependencies: User feedback collection, real-world testing environments
- **Deployment and Maintenance**
  - **Cloud Deployment**
    - Task: Deploy the system on cloud servers
    - Dependencies: AWS EC2, load balancing, scalability planning
  - **Monitoring and Updates**
    - Task: Monitor performance, handle bug fixes, release updates
    - Dependencies: Monitoring tools (e.g., New Relic), logging mechanisms

### 5.3.3 Timeline Chart

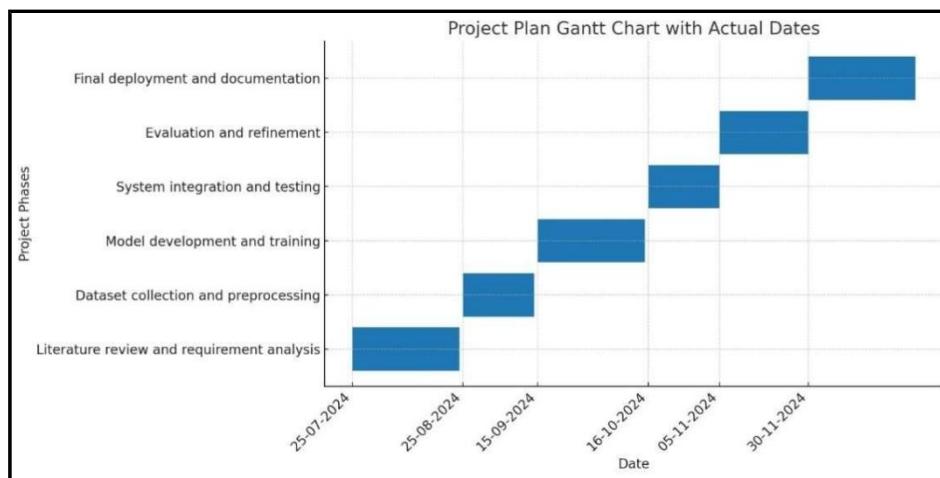


Figure 5.3.3 Timeline Chart

## 5.4 Team Organization

Table 5.4 Team Structure

| Role                               | Team member                          | Responsibilities  |
|------------------------------------|--------------------------------------|---|
| <b>Project Management</b>          | Chandrakant Shelke, Abhijeet Karande | Manages project milestones, timelines, and team coordination. Ensures successful implementation of system components.                                 |
| <b>Backend Development</b>         | Chandrakant Shelke, Shreyash Lohar   | Develop and maintain the backend using Node.js and Express. Ensure scalability and security of the application. Implement server-side logic and APIs. |
| <b>Frontend Development</b>        | Abhishek Jogdand, Shreyash Lohar     | Design and implement the user interface (UI) with a focus on user experience. Ensure smooth communication between frontend and backend                |
| <b>API Integration and Testing</b> | Abhishek Jogdand, Shreyash Lohar     | Develop and integrate APIs. Conduct system testing, bug fixing, and ensure efficient communication between different components of the system         |
| <b>Documentation and Support</b>   | Abhishek Jogdand, Abhijeet Karande   | Create and maintain project documentation. Provide technical support and troubleshooting for the system.  |
| <b>Database Management</b>         | Chandrakant Shelke, Abhijeet Karande | Design and manage the database schema. Implement data storage, retrieval, and optimization. Ensure data integrity and security.                       |

### 5.4.1 Management reporting and communication

The project team will maintain a structured approach to management reporting and communication throughout the duration of the CodeNexus project. The team will report to the project guide, Mrs. Dr. Shanthi Guru, on a weekly basis, providing updates on progress, challenges, and milestones achieved.

In addition, the team will hold weekly meetings to facilitate internal communication, allowing members to discuss ongoing tasks, share insights, and address any blockers. This collaborative environment will ensure that all team members are aligned with project goals and can effectively contribute to the project's success. Regular updates and open channels of communication will foster transparency and accountability within the team.

# Software requirement specification

## 6.1 Introduction

The CodeNexus E-Learning Platform is a unified, scalable system designed to provide personalized programming education. It consolidates essential resources such as interactive content, video tutorials, personalized quizzes, and certification opportunities. The platform dynamically adapts to individual learning paths to deliver an engaging and customized learning experience for users.

### 6.1.1 Purpose and Scope of Document

The Software Requirements Specification (SRS) outlines the functional and non-functional requirements for the CodeNexus E-Learning Platform. This document is a guide for developers, project managers, and stakeholders, detailing the objectives, features, and constraints of the system. It covers the platform's architecture, user interactions, data handling processes, and performance expectations.

### 6.1.2 Overview of responsibilities of Developer

The developer is responsible for managing the entire development lifecycle of the CodeNexus E-Learning Platform. Key responsibilities include:

- **Requirement Analysis:** Collaborating with stakeholders to define system features, educational content, and user expectations.
- **System Design:** Creating an architecture that integrates interactive learning resources, adaptive quizzes, and external content APIs, ensuring scalability and responsiveness.
- **Content and Resource Integration:** Developing and integrating various educational resources such as video tutorials, quizzes, and certification modules into the platform.
- **Database Design:** Structuring and managing databases to store user progress, quiz results, and personalized learning paths, ensuring efficient data retrieval and storage.
- **User Interface Development:** Designing an intuitive, user-friendly interface that enhances the learning experience and allows users to navigate easily between lessons, resources, and quizzes.
- **Testing and Validation:** Conducting comprehensive testing to ensure system performance, user engagement, and content accuracy meet the required standards.
- **Deployment and Maintenance:** Overseeing the launch of the platform, managing post-deployment issues, and ensuring continuous updates and improvements based on user feedback.

- **Documentation:** Preparing detailed documentation for developers, administrators, and end-users, explaining the platform's usage, features, and technical details.

## 6.2 Usage Scenario

The CodeNexus E-Learning Platform interacts with users by providing personalized programming content based on their progress, learning paths, and preferences. Different profiles are involved in the system, each with specific interactions.

### 6.2.1 User profiles

Various categories of users interact with the platform:

- General User:
  - Description: Learners seeking personalized programming education, quizzes, and certifications based on their skills and progress.
  - Responsibilities: Access learning materials, complete quizzes, track progress, and provide feedback on course recommendations.
- Administrator:
  - Description: Oversees the management and smooth operation of the platform, ensuring proper functionality and user satisfaction.
  - Responsibilities: Monitors system performance, manages user accounts, updates content, and oversees quiz and certification modules.
- Content Provider:
  - Description: Contributes educational content such as tutorials, quizzes, and learning resources.
  - Responsibilities: Uploads and manages educational resources, ensures accuracy and relevance of the content, and tracks performance of uploaded materials.

### 6.2.2 Use-cases

All use-cases for the software are presented. Description of all main Use cases using use case template is to be provided.

Table 6.2.2: Use Cases

| Sr.no | Test CaseName | Description                                      | Expected Result           | Actual Result    | Remark |
|-------|---------------|--|---------------------------|------------------|--------|
| 1.    | Student Login | Enter Username and Password for login to website | Login should beSuccessful | Login Successful | Pass   |

|    |                      |  |   |   |      |
|----|----------------------|--|---|---|------|
| 2. | Admin Login          | Check response when valid name and password is entered               | Doctor Login should be Successful   | Admin Login Successful  | Pass |
| 3. | Open Service section | To check whether the section is responding or not.                   | Successfully open and show content from database  | Opened section and list down available courses with their photo | Pass |
| 4. | Notes display        | To check whether predefined notes are opening or not                 | Open a note specified in database for that topic respected to subject                     | Display notes of topic which is selected from list              | Pass |
| 5. | Run compiler         | To check whether compiler / interpreter is working                   | Respond to user written code and generate the output                                      | Generate output according to the programmer                     | Pass |
| 6. | Video play           | To check whether the assigned video is available and playing or not  | Video gets opened in iframe and starts playing on selecting the topic respected to course | Video played successfully                                       | Pass |
| 7. | Notes Operation      | To check whether the insertion and retrieval function working or not | Display notes from database & update when update button is clicked                        | Successfully retrieved and updated                              | Pass |

|     |                             |  |   |   |      |
|-----|-----------------------------|--|---|---|------|
| 8.  | StartQuiz                   | Check whether the quiz is starting or not                                      | Open a quiz inside frame respected to topic selected particular to programming language | Quiz started froselected topic                              | Pass |
| 9.  | Quiz Validation             | Check the validation (insert score, reattempt quiz, avoid repetition of quiz ) | Proceed for further process if condition is true otherwise show error                   | Prepossessing for true condition, Error for false condition | Pass |
| 10. | Certificate Generation      | Generate the certificate for student based on their marks obtain               | Calculate their score & generate certificate as per that result                         | Result generated successfull y                              | Pass |
| 11. | Send Email                  | Send email from contact us section for enquiry purpose                         | Notification formail send   | Mail send successfull y                                     | Pass |
| 12. | Manage Programming Language | Insert, update, delete Content of language section from admin                  | Given operation has been successfully performed   | Operation performed successfull y                           | Pass |
| 13. | Manage Video Section        | Insert, update, delete Content of Video section from admin                     | Given operation has been successfully performed   | Operation performed successfull y                           | Pass |

|     |                |  |                                 |                                     |      |
|-----|----------------|--|---------------------------------|-------------------------------------|------|
| 14. | Delete Student | Delete user from admin login and data related to that user | User deleted a and data deleted | User and Data deleted successfull y | Pass |
|-----|----------------|--|---------------------------------|-------------------------------------|------|

### 6.2.3 Use Case View

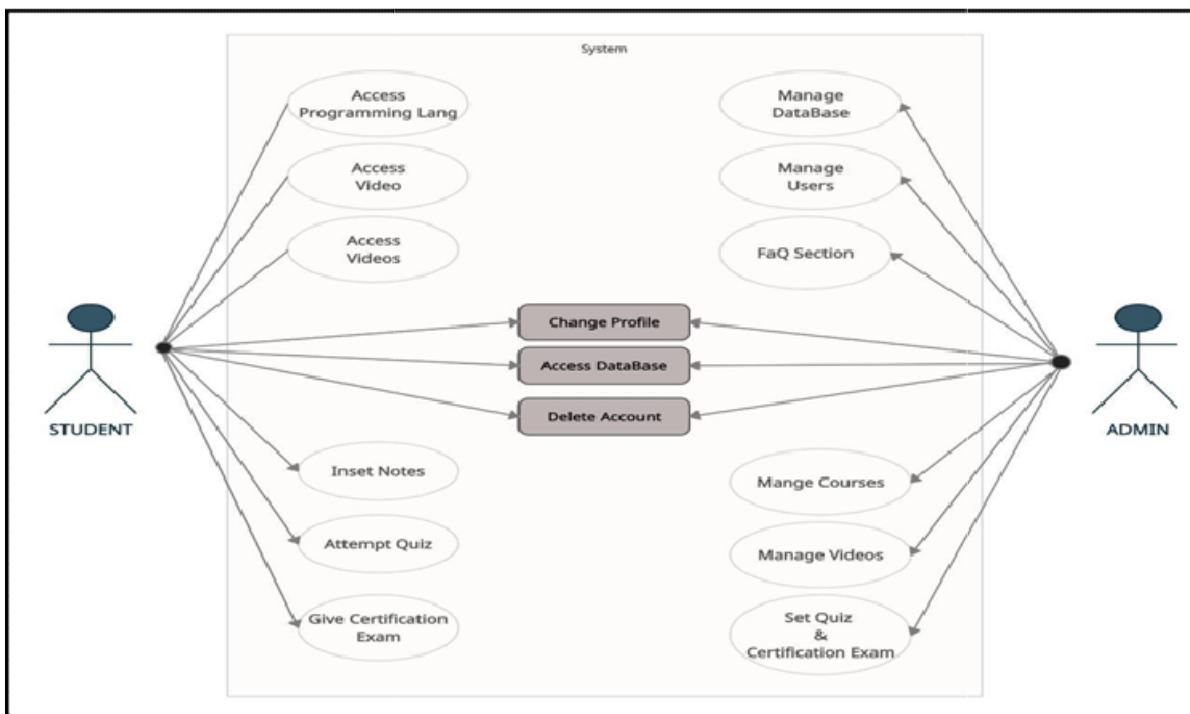


Figure 6.2.3: Use case diagram

## 6.3 Data Model and Description

### 6.3.1 Data Description

The CodeNexus E-Learning Platform manages various data objects essential for its operation. These data objects include user information, learning progress, course metadata, quiz results, and platform configurations. Each data object is structured to facilitate efficient storage, retrieval, and manipulation to support personalized learning paths and a seamless user experience.

### Key Data Objects:

- **User Profile:**
  - **Attributes:** User ID, Username, Email, Subscription Status, Learning Preferences, Profile Settings
- **Learning Progress:**
  - **Attributes:** Progress ID, User ID, Course ID, Completed Sections, Time Spent, Quiz Results, Timestamp
- **Course Metadata:**
  - **Attributes:** Course ID, Title, Author, Language, Topic, Difficulty Level, Resources (e.g., video, quizzes, tutorials)
- **Quiz Results:**
  - **Attributes:** Quiz ID, User ID, Course ID, Score, Questions Attempted, Time Taken, Timestamp
- **Notes:**
  - **Attributes:** Note ID, User ID, Course ID, Section, Content, Timestamp
  - **Attributes:** Playlist ID, User ID, Song IDs, Playlist Name, Creation Date, Shared Status

#### 6.3.2 Data objects and Relationships

The relationships among data objects are structured to ensure data integrity and support the system's functionalities. An **Entity-Relationship Diagram (ERD)** outlines these relationships, highlighting how different entities interact within the platform.

##### Entities and Relationships:

- **User Profile** is connected to **Learning Progress**, **Quiz Results**, and **Notes** through the User ID.
- **Learning Progress** is linked to **Course Metadata**, tracking user advancement in specific courses through the Course ID.
- **Quiz Results** are associated with both **User Profile** and **Course Metadata**, recording individual quiz outcomes per course.
- **Notes** are tied to specific **User Profiles** and **Course Metadata**, allowing users to save and organize personal notes related to course content.

## 6.4 Functional Model and Description

### 6.4.1 Data Flow Diagram

#### 6.4.1.1 Level 0 Data Flow Diagram

The Level 0 Data Flow Diagram provides a high-level overview of the system, showcasing the primary processes, data stores, and external entities involved.

**Flow:**

1. Users interact with the system, providing input on their learning preferences and progress, receiving personalized learning paths and resources.
2. Content Providers upload and manage educational materials.
3. The system processes inputs from users and recommends learning materials, quizzes, and certification paths based on their progress and preferences.
4. Feedback from users is collected to improve content recommendations and platform performance.

#### 6.4.1.2 Level 1 Data Flow Diagram

The Level 1 Data Flow Diagram breaks down the main processes identified in Level 0 into more detailed subprocesses and data flows.

**Flow:**

1. Users log in and set their preferences or continue from where they left off.
2. The system collects and analyses user progress and quiz results to customize learning paths.
3. Based on the collected data, the system recommends relevant courses, tutorials, and quizzes that match the user's learning needs and preferences.
4. Users interact with the recommended learning resources and complete quizzes, providing feedback that is processed to enhance the recommendations.

Content providers manage and upload new educational materials, ensuring that the platform's database remains current and diverse.

### 6.4.2 Activity Diagram:

The Activity Diagram represents the sequential flow of activities within the CodeNexus platform that directs the activity flow for the user and admin panel.

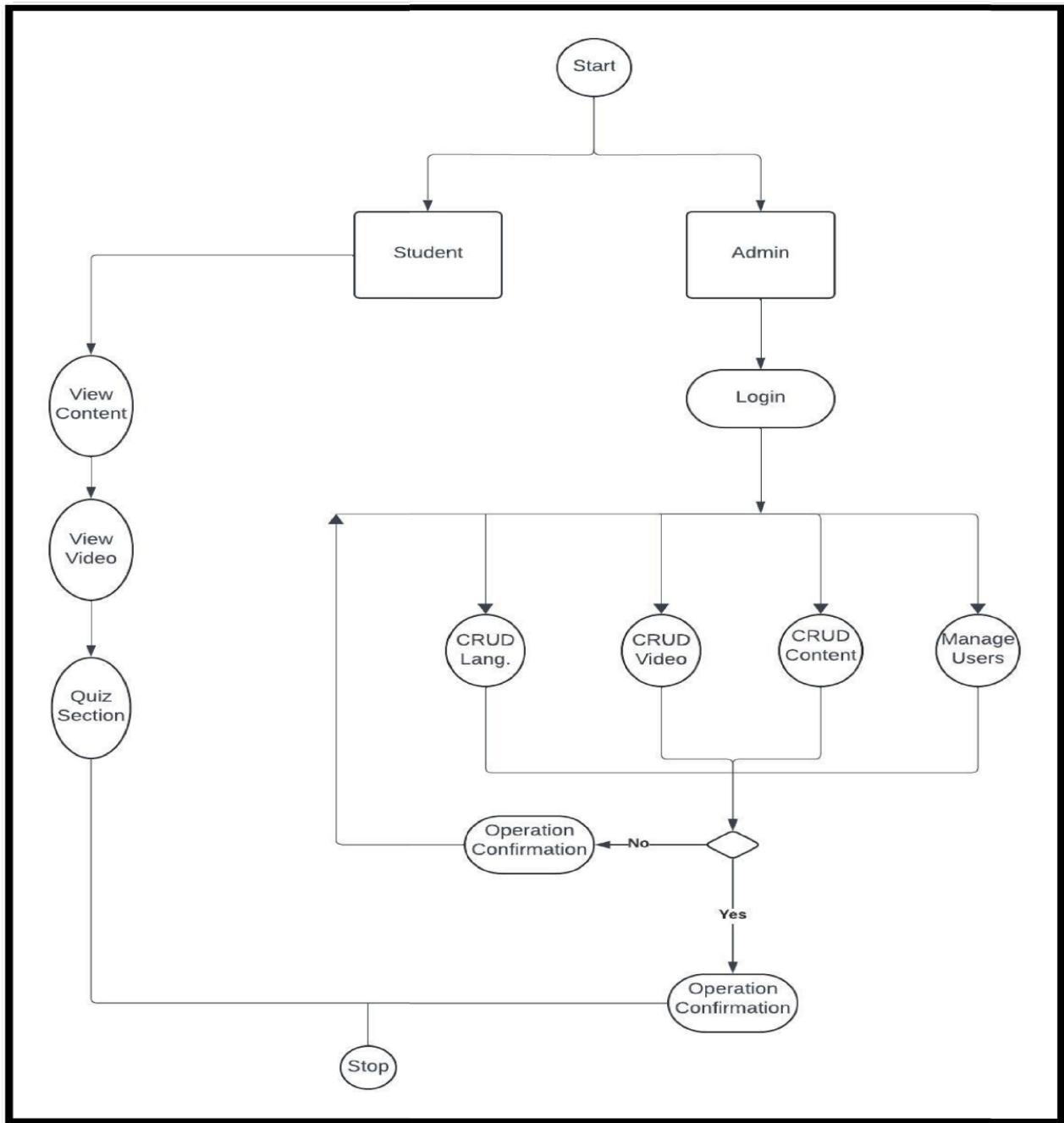


Fig.6.4.2 Activity diagram

### 6.4.3 Non-Functional Requirements:

- **Interface Requirements:**
  - **UI:** Responsive across devices.
  - **API:** Secure for external integrations.
- **Performance Requirements**
  - **Throughput:** Handles multiple users
  - **Availability:** 99.9%
- **Software quality attributes**
  - **Reliability:** Minimal down time.
  - **Usability:** User-friendly and adaptable.
  - **Modifiability:** Easy updates and features.

### 6.4.4 State Diagram:

A state diagram, sometimes known as a state machine diagram, is a type of behavioral diagram in the Unified Modeling Language (UML) that shows transitions between various objects.

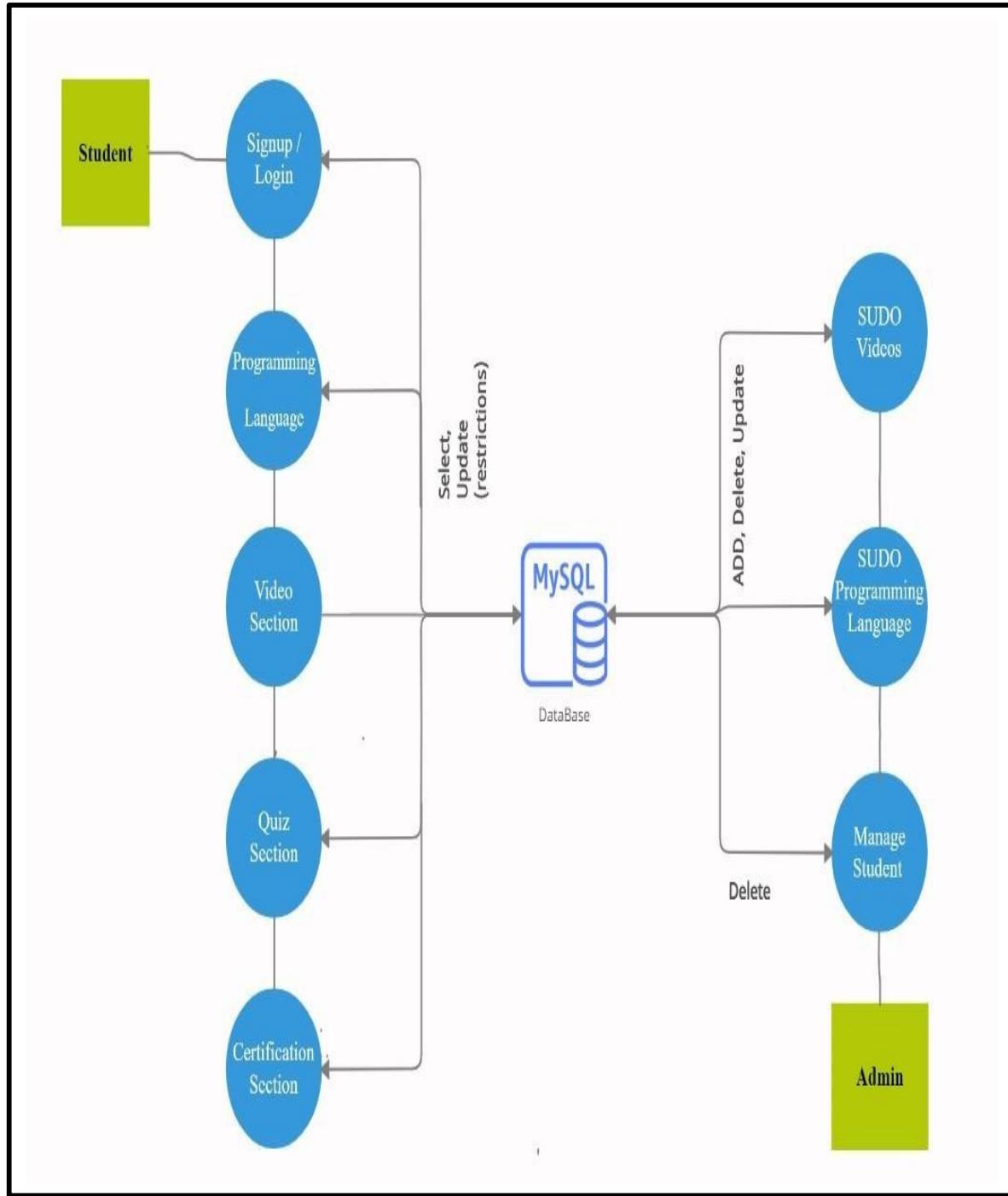


Figure 6.4.4: State Transition Diagram

#### 6.4.5 Design Constraints

Design constraints are limitations or restrictions that impact the system's architecture and implementation. These constraints ensure the system aligns with technical, operational, and business requirements.

**Key Design Constraints:**

- **Technology Stack:** Use of React for frontend and Node.js or php for backend services.
- **Resource Constraints:** Optimize computational resources to manage costs, ensuring scalability as user traffic increases.
- **Integration Standards:** Follow standardized APIs for external system integration, ensuring compatibility with third-party educational resources.
- **Performance Limits:** Ensure course recommendations and data retrieval are completed within defined response times.
- **Security:** Implement encryption, secure authentication, and conduct regular security audits to protect user data.
- **UI Guidelines:** Adhere to accessibility best practices to ensure ease of use for all learners.
- **Dependency Management:** Minimize reliance on third-party services for better reliability and control.

#### 6.4.6 Software Interface Description

**User Interfaces:**

- **Web Interface:**  
Full functionality for coding, compiling, and managing projects via web browsers.

**API Interfaces:**

- **Public APIs:**  
For third-party integration to access project and compilation functionalities.
- **Internal APIs:**  
Facilitate communication between frontend, backend, and database modules.

**External System Interfaces:**

- **Code Compilation Services:**  
Interfaces with external compilers to execute user code securely.

**Database Interfaces:**

- **DBMS (Relational/NoSQL):**  
Manages user data, project metadata, and version history.

**Interface Requirements:**

- **Consistency:**  
Uniform design and API standards.
- **Performance:**  
Ensure fast response times with minimal latency.
- **Documentation:**  
Clear API documentation for easy integration..

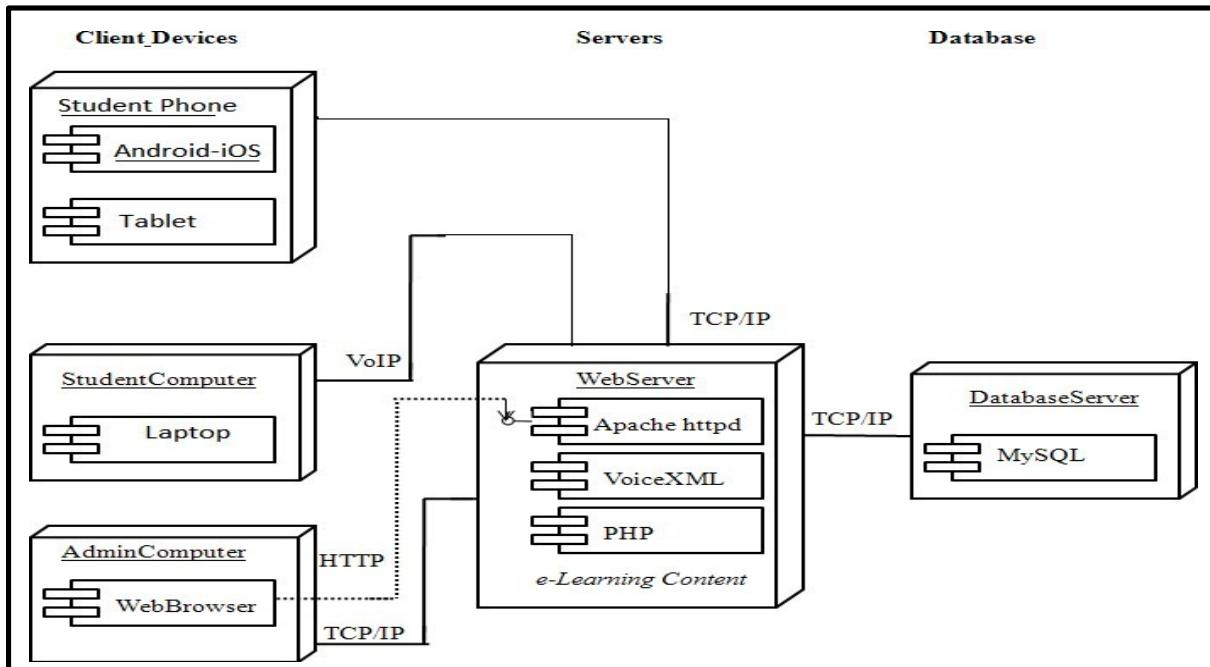
# Detailed Design Document using Appendix A and B

## 7.1 Introduction

A Codenexus aims to create a cutting-edge e-learning platform tailored to both beginners and advanced programmers, addressing the rising demand for accessible, scalable, and dynamic online programming education. This platform integrates advanced AI-driven features with interactive learning content to enhance user engagement and efficiency in skill-building. The core objective of Codenexus is to offer a cohesive, personalized educational experience that blends real-time support, curated learning resources, and administrative tools for effective content management. Through this approach, Codenexus fosters an adaptive and supportive learning environment..

## 7.2 Architectural Design

The platform's architecture consists of several interlocking modules, including AI-Powered Chatbot Assistance, Personalized Learning Content, Interactive Learning Tools, and Administrative Dashboard. Together, these modules process user inputs to deliver relevant support, customized educational content, and effective content management, thereby creating a seamless, comprehensive user experience.



7.2.1 : Deployment Diagram

## 7.3 Data design (using Appendices A and B)

Data structures are designed to support real-time interaction and personalized learning, involving both internal and global structures to streamline user experience..

### 7.3.1 Internal software data structure

- **User Interaction Data:** Tracks user queries and responses in real-time to facilitate AI chatbot interactions.
  - Format: Text data and interaction history
  - Structure: UserInteraction {String query; String response; Timestamp time;}
- **Learning Progress Data:** Maintains user progress in courses, quizzes, and certifications.
  - Format: Course identifiers and progress metrics
  - Structure: LearningProgress {String courseID; float completionPercentage; List<String> quizScores;}

### 7.3.2 Global data structure

**User Profile Data:** Contains user preferences and historical data.

```
struct UserProfile {
    String userID;
    String username;
    List<String> completedCourses;
    Map<String, Integer> quizScores;
    String certificationStatus;
}
```

**Content Metadata:** Metadata for tutorials, quizzes, and certification materials.

```
struct ContentMetadata {
    String contentID;
    String title;
    String contentType; // e.g., "tutorial", "quiz", "certification"
    String difficultyLevel;
    List<String> tags;
}
```

### 7.3.3 Temporary data structure

**Session Data:** Temporary data capturing the current user session and learning activities.

Structure:

```
struct SessionData {
    String sessionID;
    List<String> activeModules;
    String currentQuiz;
}
```

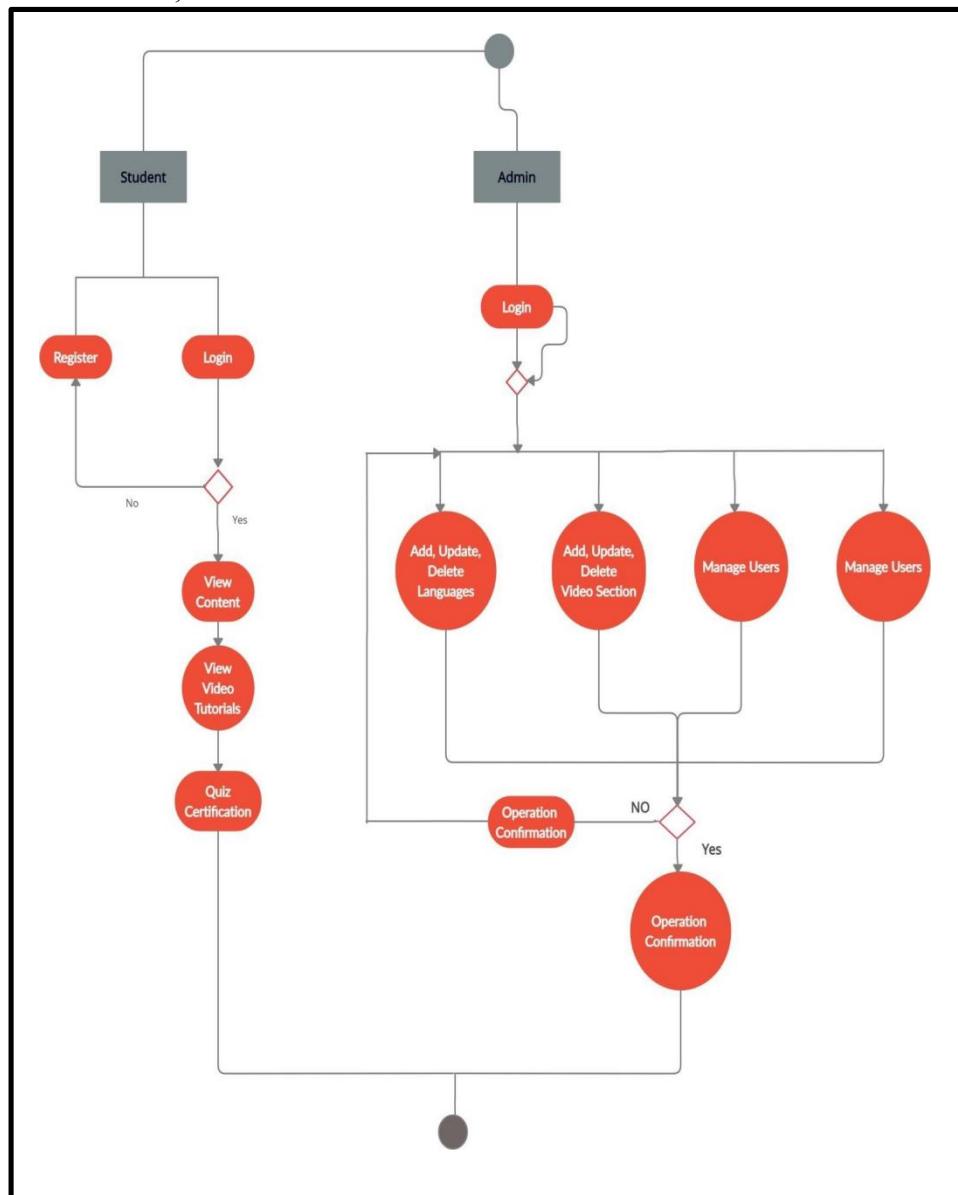


Figure 7.3.3: Architecture Diagram

### 7.3.4 Database description

Codenexus uses a relational database to securely store content metadata, user profiles, and progress data. This persistent storage solution facilitates a personalized learning journey by recording user progress, quiz results, and certification achievements.

#### 1. User Table:

- **Purpose:** Contains metadata for all tutorials, quizzes, and certifications.
- **Fields:**
  - **userID:** Primary key, unique identifier for each user.
  - **username:** The user's display name.
  - **email:** User's email address.
  - **completeCourses:** List of courses previously completed.

#### 2. Content Table:

- **Purpose:** Contains metadata for all tutorials, quizzes, and certifications.
- **Fields:**
  - contentID, title, contentType, difficultyLevel, tags

#### 3. Progress Table:

- **Purpose:** Tracks individual user progress..
- **Fields:**
  - progressID, userID, contentID, completionPercentage, quizScores

#### 4. Certification Table:

- **Purpose:** Stores Manages certification status and records.
- **Fields:**
  - certificationID, userID, certificationStatus, issueDate

## 7.4 Component Design

The component design consists of four main parts: **AI Chatbot Assistance**, **Personalized Learning Module**, **Interactive Learning Tools**, and **Administrative Dashboard**. Each component interacts with the database to provide tailored support, maintain progress, and deliver engaging educational resources.**Facial Emotion Detection Component:**

- **Purpose:** To provide real-time responses to user queries.

- **Key Methods:**

- `processQuery(query)`: Analyzes user questions and returns relevant responses.\
- `storeInteraction(query, response)`: Logs each interaction for future insights.

**Personalized Learning Component:**

- **Purpose:** Delivers tailored content based on user profile and learning progress.
- **Key Methods:**
  - `fetchContentByLevel(difficultyLevel)`: Retrieves content based on the user's proficiency level.
  - `trackProgress(userID, contentID)`: Updates user's progress data.

**Interactive Learning Tools Component:**

- **Purpose:** Offers quizzes and interactive elements to reinforce learning.
- **Key Methods:**
  - `generateQuiz(contentID)`: Produces quizzes aligned with the content.

**Administrative Dashboard Component:**

- **Purpose:** Provides educators with tools for content and user management.
- **Key Methods:**
  - `uploadContent(contentData)`: Allows administrators to upload new learning resources.
  - `manageUserProfiles(userID)`: Facilitates user support and progress tracking.

**updateUserProfile (userID, newUser):**

- **Input:**
  - `userID`: The unique identifier of the user.
  - `newRecommendation`: The latest user recommended.
- **Process:**
  - Updates the user's profile by adding the newly recommended changes to their history.
  - Adjusts the user's history, allowing the system to refine predictions based on evolving emotional patterns over time.
- **Output:**
  - An updated user profile that considers both new recommendations and the user's ongoing interactions with the system.

### 7.4.1 DFD Diagrams

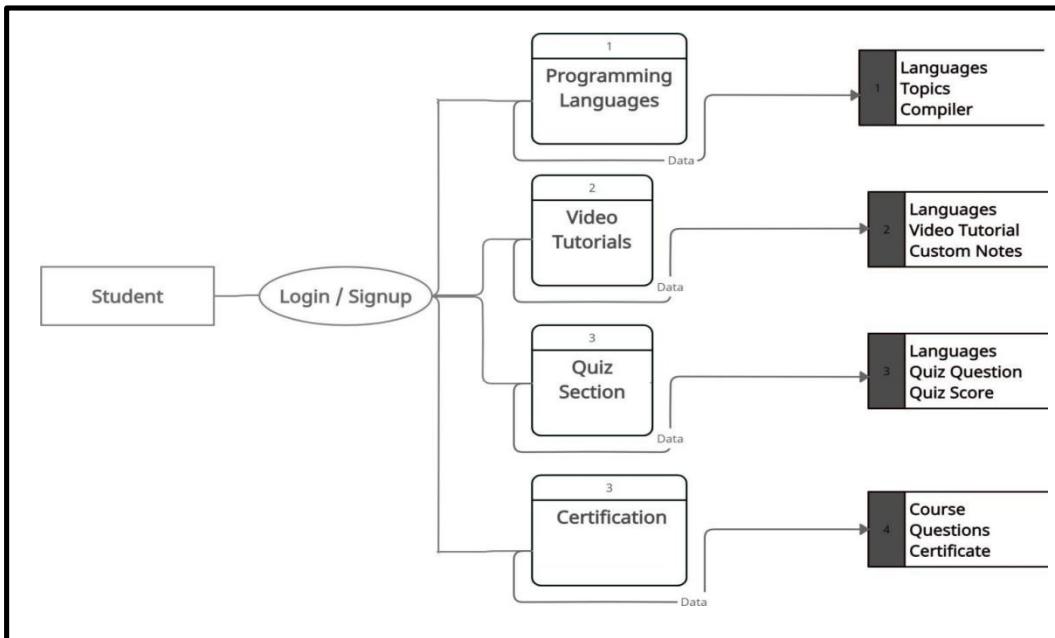


Fig. 7.4.1.1 Student Data Flow Diagram

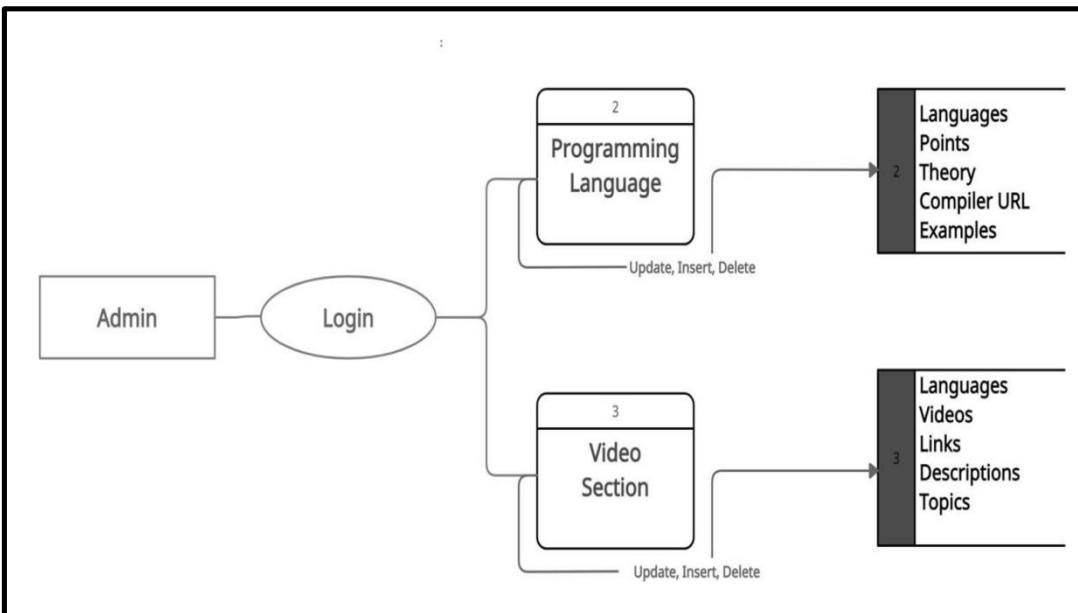


Fig. 7.4.1.2 Admin Data Flow Diagram

### 7.4.2 Class Diagram

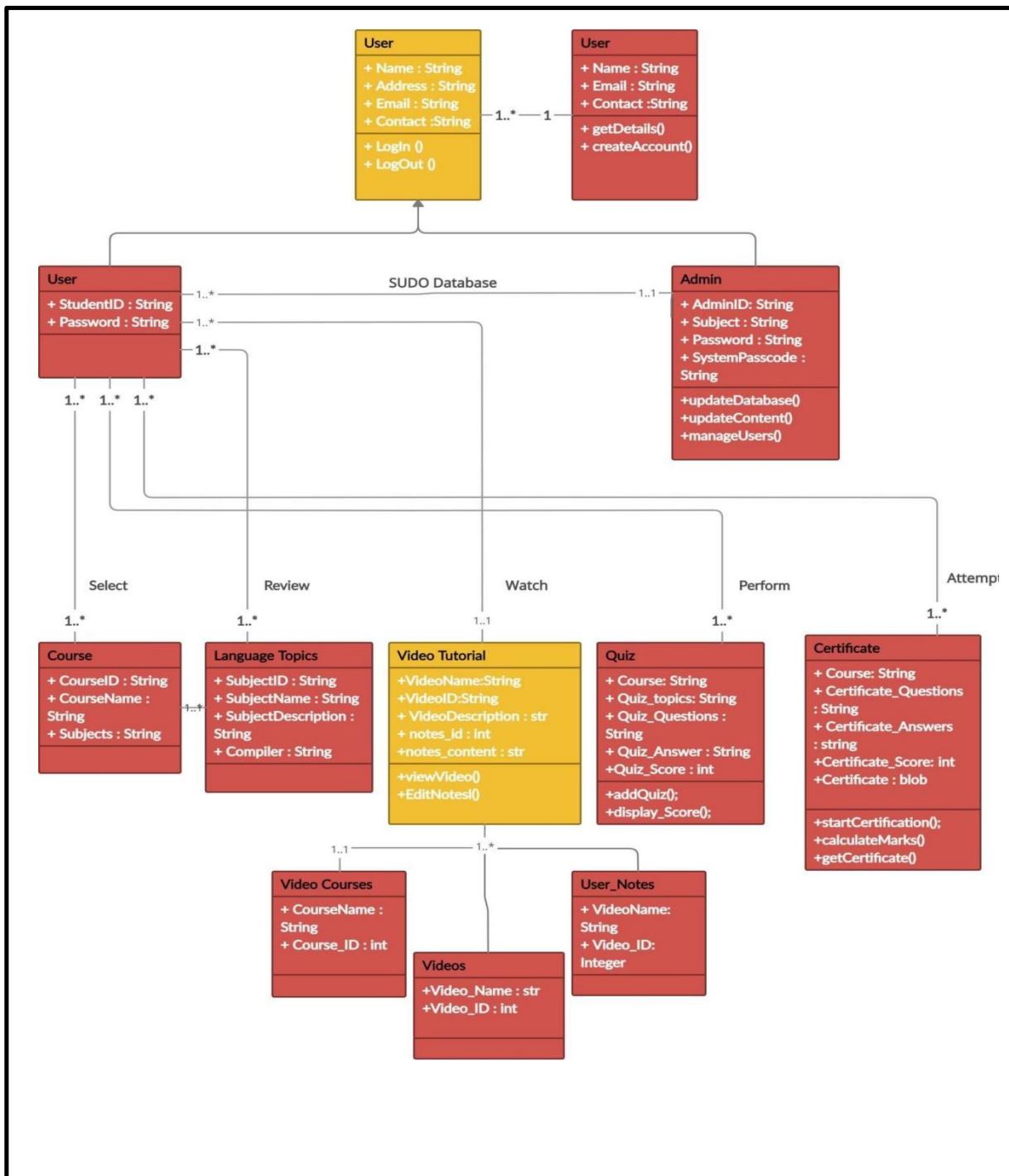


Figure 7.4.2.1 Class Diagram

# Project Implementation

## 8.1 Introduction

Codenexus leverages an integrated system of user-responsive learning modules and administrative tools to deliver a comprehensive programming education experience. By combining AI-driven support with dynamic learning resources, Codenexus strives to create an efficient and adaptive learning platform.

The platform's Manage Programming Languages section enables administrators to easily add, update, or remove programming languages and their learning topics. This flexibility ensures that Codenexus stays current with the ever-evolving landscape of programming technologies. Moreover, the Manage Videos section provides a space where administrators can curate educational content, including video links and descriptions, ensuring that students have access to the latest and most relevant learning materials. A unique feature of Codenexus is its comprehensive certification management system, which allows administrators to create and manage certification exams. They can add, update, or remove exam questions and answers, tailoring the exams to the learning objectives of each programming language. This ensures that certification reflects real-world skills, giving students confidence in their abilities as they enter the job market. The platform's communication system also benefits administrators by giving them the ability to view and respond to student queries, ensuring that any issues or feedback are addressed promptly and effectively.

## 8.2 Tools and Technologies Used

- **Programming Languages:**
  - React for Frontend.
  - JavaScript for real-time user interface interaction.
- **Libraries and Frameworks:**
  - React for responsive UI
  - Node.js and Express for backend functionality
  - TensorFlow and Scikit-learn for AI-driven personalization
- **Database:**
  - MySQL for user and content data storage

| <b>1) Student Module Workflow Algorithm</b>  | <b>2) Admin Module Workflow Algorithm</b>   |
|--|---|
| <p><b>•Algorithm: Student Access</b></p> <p>1. Start</p> <p>2. <b>Login:</b></p> <p>1. Input: Student credentials (Username and Password).</p> <p>2. Verify credentials against the database.</p> <p>3. If valid, proceed; if not, return error message.</p> <p>3. <b>Access Learning Material:</b></p> <p>1. Display available courses, videos, and notes based on the student's chosen topic.</p> <p>2. Search for videos/notes based on the selected programming language or topic.</p> <p>3. Retrieve the corresponding video or notes from the database.</p> <p>4. <b>Quizzes:</b></p> <p>1. Allow students to select a quiz after completing a topic.</p> <p>2. Retrieve quiz questions from the database.</p> <p>3. Student inputs answers and submits the quiz.</p> <p>5. <b>Score Calculation:</b></p> <p>1. Compare submitted answers with correct answers in the database.</p> <p>2. Calculate the score and store the result.</p> <p>6. <b>Certification:</b> 1. If score meets/exceeds the threshold, generate a certificate.</p> | <p><b>•Algorithm: Admin Management</b></p> <p>Start</p> <p>2. <b>Login:</b></p> <p>1. Input: Admin credentials (Username and Password).</p> <p>2. Verify credentials against the database.</p> <p>3. If valid, proceed to the admin dashboard.</p> <p>3. <b>Manage Users:</b></p> <p>1. Add, update, or delete user profiles.</p> <p>2. If adding a new user, input user data (username, password, role).</p> <p>3. Store/update/delete user data in the database.</p> <p>4. <b>Manage Content:</b></p> <p>1. Add, update, or delete programming languages, videos, and quizzes.</p> <p>2. For each content update:</p> <p>1. Input the new content details (topic, video description, quiz questions).</p> <p>2. Validate the data before submission.</p> <p>3. Store the new content in the appropriate database tables.</p> <p>5. <b>Review Student Feedback/Queries:</b></p> <p>1. Retrieve student queries from the contact form database.</p> <p>2. Respond to feedback or resolve issues.</p> <p>6. <b>End</b></p> |

|   |   |
|---|---|
| <p><b>3) Quiz and Certification Module Workflow Algorithm</b></p> <ul style="list-style-type: none"> <li><b>Algorithm: Quiz Submission and Certificate Generation</b></li> </ul> <p><b>1. Start</b></p> <p><b>2. Quiz Access:</b></p> <ol style="list-style-type: none"> <li>1. Student selects a quiz related to a specific course or topic.</li> <li>2. Retrieve quiz questions from the database.</li> </ol> <p><b>3. Quiz Submission:</b></p> <ol style="list-style-type: none"> <li>1. Student submits answers.</li> <li>2. Compare answers to the correct answers stored in the database.</li> </ol> <p><b>4. Score Calculation:</b></p> <ol style="list-style-type: none"> <li>1. For each correct answer, increase the score.</li> <li>2. Store the final score in the database.</li> </ol> <p><b>5. Certificate Generation:</b></p> <ol style="list-style-type: none"> <li>1. If the score exceeds the threshold, generate a certificate.</li> <li>2. Generate a certificate with the student's name and course details.</li> </ol> <p><b>6. End</b></p> | <p><b>4) Video Retrieval and Management Workflow Algorithm</b></p> <p><b>1. Start</b></p> <p><b>2. Admin Management:</b></p> <ol style="list-style-type: none"> <li>1. Admin can add, update, or delete videos.</li> <li>2. Input the video link, description, and related topic.</li> <li>3. Validate video data (format and size).</li> <li>4. Store video metadata in the database.</li> </ol> <p><b>3. Student Access:</b></p> <ol style="list-style-type: none"> <li>1. Student selects a course.</li> <li>2. Retrieve and display related videos from the database.</li> <li>3. Stream the video using the embedded player.</li> </ol> <p><b>4. End</b></p> |
|---|---|

|  |  |
|--|--|
| <p><b>5) Notes Management Module Workflow Algorithm</b></p> <p>• <b>Algorithm: Notes Creation and Retrieval</b></p> <p><b>1. Start 2. Create/Update</b></p> <p><b>Notes:</b></p> <p>1. Student writes or edits notes related to a topic.</p> <p>2. Input the notes and save them in the notes database.</p> <p><b>3. Save Notes:</b></p> <p>1. Ensure notes are saved with the correct timestamp and topic tags.</p> <p><b>4. Retrieve Notes:</b></p> <p>1. When requested, fetch the saved notes from the database based on user and topic.</p> <p>2. Display the notes in the student's dashboard.</p> <p><b>5. Delete Notes:</b></p> <p>1. Allow students to delete notes.</p> <p>2. Confirm deletion and remove the corresponding note entry from the database.</p> <p><b>6. End</b></p> | <p><b>6) AI Chatbot Algorithm Workflow Algorithm</b></p> <ol style="list-style-type: none"> <li>1. <b>Start Chatbot Process</b> <ul style="list-style-type: none"> <li>o Initialize and prepare the chatbot for user interaction.</li> </ul> </li> <li>2. <b>Initialize Chatbot Components</b> <ul style="list-style-type: none"> <li>o Load any pre-trained models, knowledge base, or connect APIs that support response generation.</li> </ul> </li> <li>3. <b>Listen for User Input</b> <ul style="list-style-type: none"> <li>o Wait for the user to enter a query, question, or message.</li> </ul> </li> <li>4. <b>Capture and Process User Query</b> <ul style="list-style-type: none"> <li>o Capture the user's input and store it for processing.</li> </ul> </li> <li>5. <b>Analyze Input with NLP</b> <ul style="list-style-type: none"> <li>o Tokenize the text, identify keywords, and use Natural Language Processing (NLP) to understand intent.</li> </ul> </li> <li>6. <b>Identify User Intent</b> <ul style="list-style-type: none"> <li>o Match the user's input to pre-defined intents, using either a machine learning model or rules-based approach.</li> </ul> </li> <li>7. <b>Fetch or Generate Response</b> <ul style="list-style-type: none"> <li>o If the input matches an existing intent, retrieve the appropriate response from the knowledge base.</li> <li>o If no match is found, provide a fallback response or ask the user for clarification.</li> </ul> </li> <li>8. <b>Convert to Natural Language</b> <ul style="list-style-type: none"> <li>o Use natural language generation to format the response in a readable, conversational style.</li> </ul> </li> <li>9. <b>Display Response to User</b> <ul style="list-style-type: none"> <li>o Send the response to the user via the chat interface.</li> </ul> </li> <li>10. <b>Continue or End Session</b> <ul style="list-style-type: none"> <li>Check if the user has more queries or has requested to end the session.</li> <li>• Repeat the process for new inputs, or terminate if the user opts to exit.</li> </ul> </li> </ol> |
|--|--|

# Software Testing

## 9.1 Types of Testing Used

For CodeNexus, the following types of testing were applied to ensure robust performance:

- **Unit Testing:**

Unit testing involves testing individual components or functions of the application in isolation to ensure that each part performs as expected. For CodeNexus, unit tests are written for core functionalities like user authentication, code submission handling, and evaluation logic..

- **Integration Testing:**

Integration testing checks how different modules or components of the system interact with each other. In CodeNexus, it ensures that the frontend properly communicates with backend APIs—for example, when a user submits code and receives evaluation results in real time.

## 9.2 Test Cases

Table 9.2.1 Unit Test Cases

| Test Case ID | Test Description                         | Input  | Expected Output                                   | Status           |
|--------------|--|--|---|------------------|
| UT-01        | Test code syntax error detection         | Code snippet with syntax error                   | Correct syntax error message and suggestions      | Passed           |
| UT-02        | Test code optimization suggestion        | Suboptimal code snippet (e.g., inefficient loop) | Optimized version of the code and recommendations | Passed           |
| UT-03        | Test programming language identification | Code snippet in various languages                | Correct identification of programming language    | Passed           |
| UT-04        | Test personalized code                   | User coding history and current code input       | List of relevant code suggestion                  | Partially Passed |
| UT-05        | Test learning resource suggestion        | Detected coding error and user learning history  | List of relevant learning resources or tutorials  | Passed           |

Table 9.2.2 Integration Test Cases

| Test Case ID | Test Description                       | Input   | Expected Output   | Status           |
|--------------|--|---|---|------------------|
| IT-01        | Test code analysis                     | Code snippet with errors and user coding history  | Unified Analysis With Personalized code recommendations   | Passed           |
| IT-02        | Test end-to-end coding support process | Real-time Code Submission and user coding history | Optimized code suggestions, error detection and resources | Partially Passed |

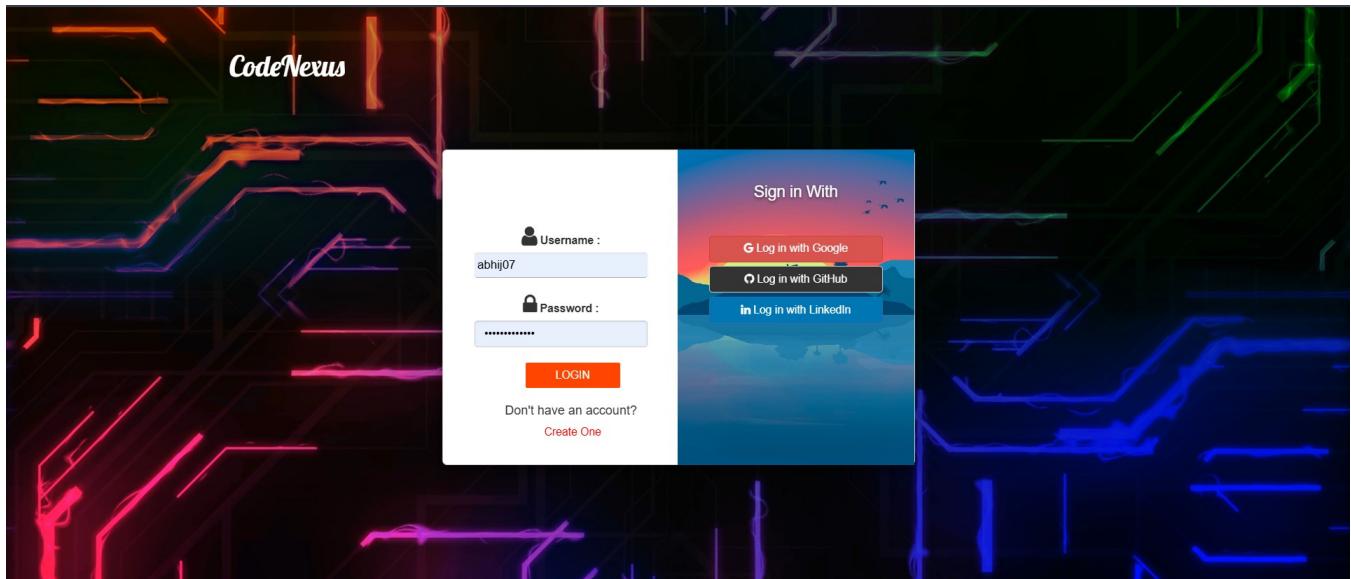
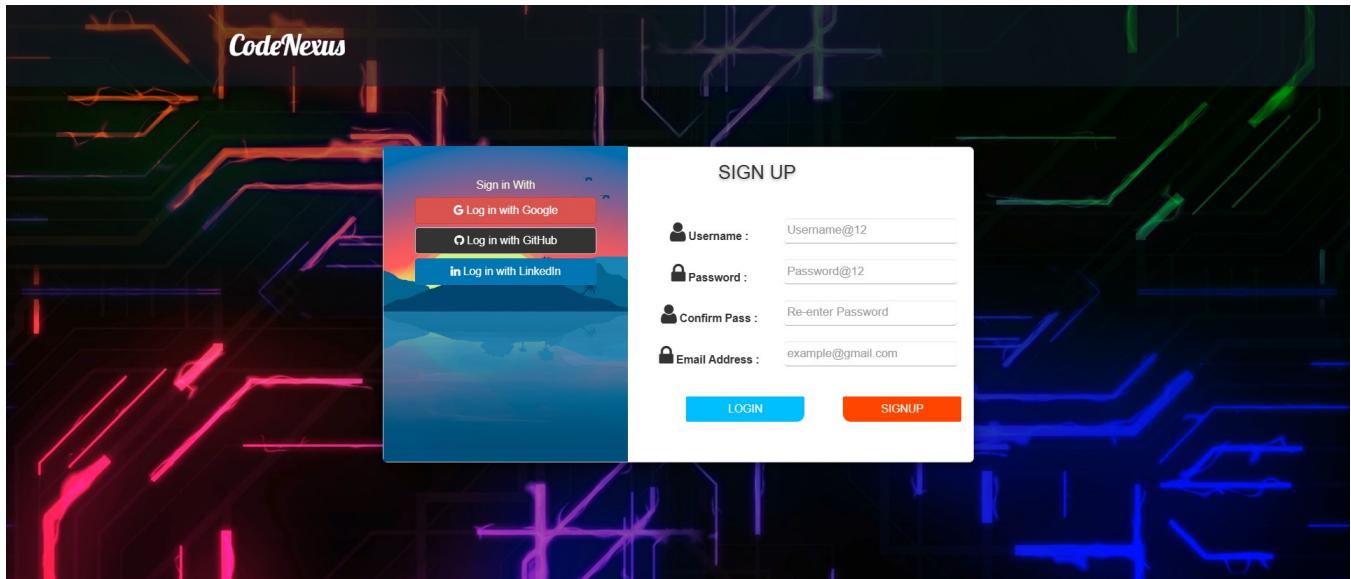
Table 9.2.3 System Test Cases

| Test Case ID | Test Description                | Input                                  | Expected Output   | Status |
|--------------|---------------------------------|--|---|--------|
| ST-01        | Test overall system performance | Multiple users code submissions        | Quick and accurate code analysis                        | Passed |
| ST-02        | Test user acceptanc             | User coding input and interaction data | Personalized and Contextually relevant code suggestions | Passed |

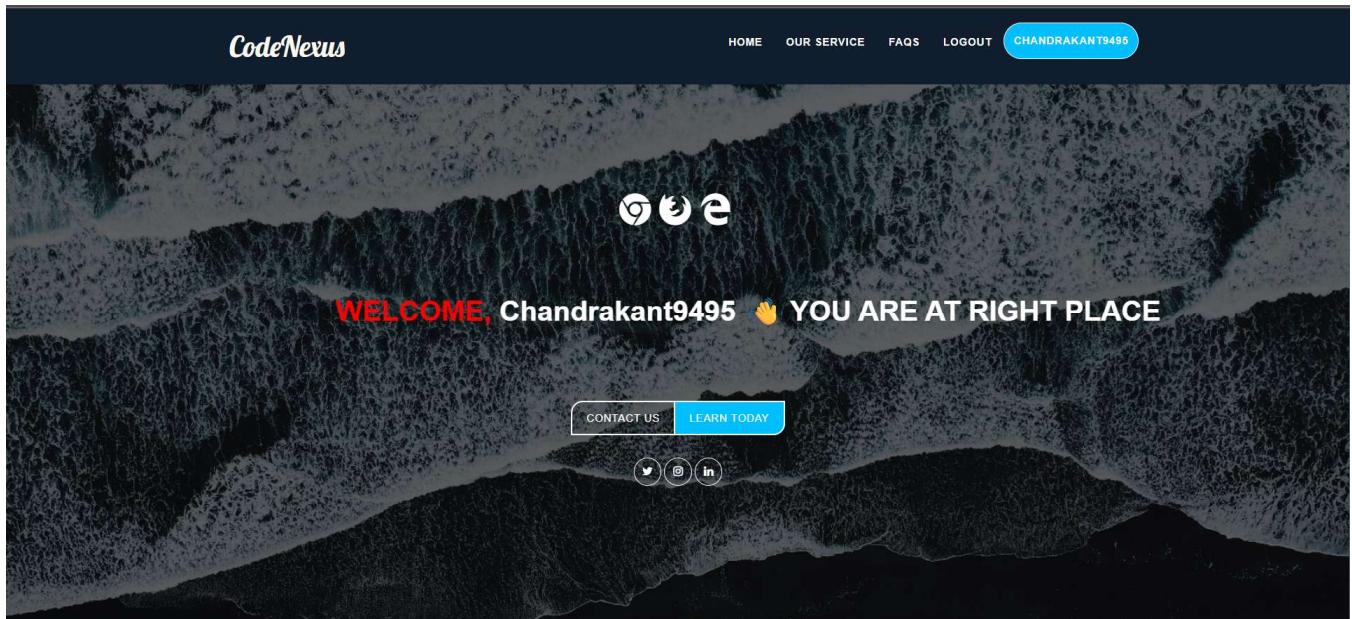
# Results

## 10.1 ScreenShots

### LOGIN PAGE :



## HOME PAGE :



## SEVICES PROVIDED BY CODENEXUS :

**SERVICES**

Services provided by us !  
Why to wait? Get started today !!

**PROGRAMMING**

Here you will find all the tutorials related to programming languages like JAVA, PYTHON, ANDROID etc

**VIDEO TUTORIALS**

Here you will find all the videos tutorials related to programming languages like JAVA, PYTHON, ANDROID etc

**MY NOTES**

Here you will find all notes of particular watched video of programming languages like JAVA, PYTHON, ANDROID etc

**LANGUAGE TOPICS QUIZ**

Complete all the topics quiz of particular language to appear for certificate exam

**CERTIFICATE**

Get your certificate of the language which you completed language topics quiz

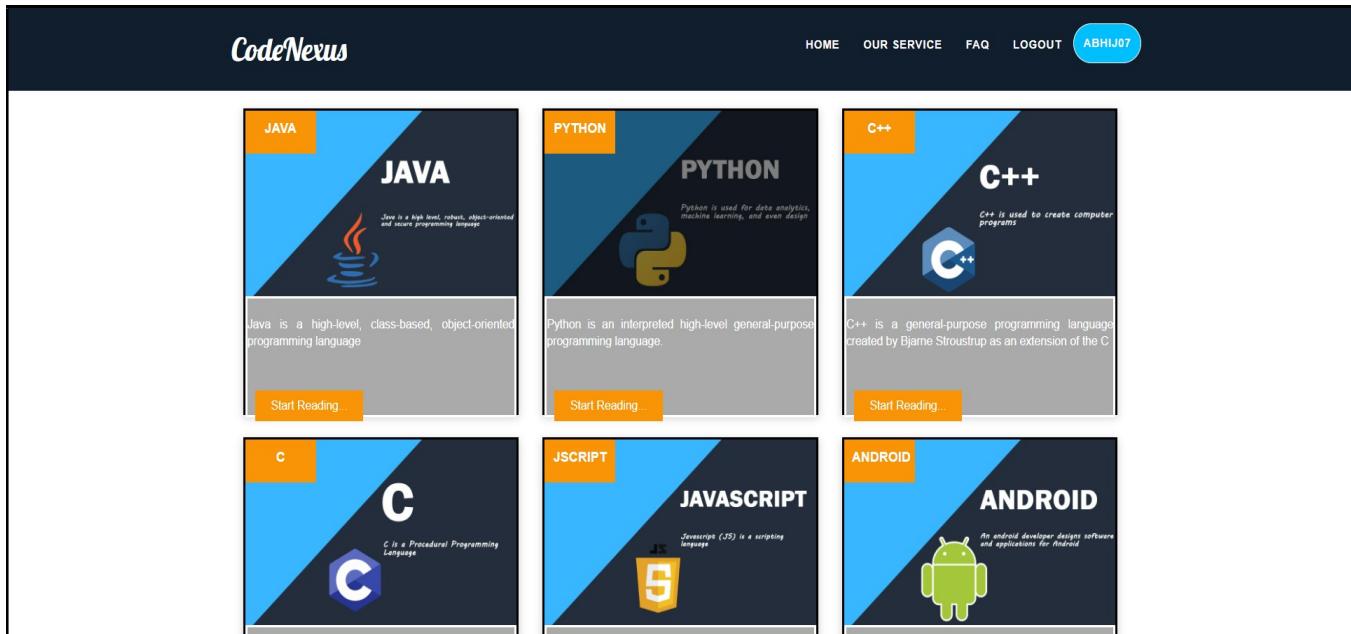
## FAQS PAGE :

The screenshot shows the 'GENERAL FAQs' section of the CodeNexus website. At the top, there is a navigation bar with links for HOME, OUR SERVICE, FAQS, LOGOUT, and a user profile labeled ABHIJ07. Below the navigation, the title 'GENERAL FAQs' is displayed, followed by a sub-instruction: 'Most common questions here! Have more doubts? Contact us now!!'. To the right of the text is a large red 3D-style 'FAQ' text icon with a white stick figure standing behind it. On the left, there is a dark blue sidebar containing four collapsed faq items, each with a minus sign icon and a question: 'What is our website is about ?', 'Which courses I will learn from this site ?', 'Can I view my notes again ?', and 'How can I get certified for a course ?'. A plus sign icon is located at the bottom right of this sidebar.

## CONTACT US :

The screenshot shows the 'CONTACT US' section of the CodeNexus website. At the top, there is a navigation bar with links for HOME, OUR SERVICE, FAQS, LOGOUT, and a user profile labeled CHANDRAKANT9496. Below the navigation, the title 'CONTACT US' is displayed, followed by a sub-instruction: 'Any Issue? Drop us below! We will contact you soon!'. To the left of the form, there is a green envelope icon with several light gray dashed lines trailing behind it. To the right of the icon, the heading 'Get in touch' is centered above a form. The form consists of four input fields: 'Name' (text), 'Email' (text), 'Subject' (text), and a large 'Message' area (text). Below the message area is a green button labeled 'Send Email →'. The entire contact form is set against a white background with a thin black border around the main content area.

## PROGRAMMING SECTION :



**Java Overview**

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be **Write Once, Run Anywhere**.

Java is –

- **Object Oriented** – In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** – Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web

Java is a high-level, object-oriented programming language developed by Sun Microsystems in 1995. It is platform-independent, which means we can write code once and run it anywhere using the Java Virtual Machine (JVM). Java is mostly used for building desktop applications, web applications, Android apps, and enterprise systems.

Key Features of Java

Platform Independent: Java is famous for its Write Once, Run Anywhere (WORA) feature. This means we can write our Java code once, and it will run on any device or operating system without changing anything.

Object-Oriented: Java follows the object-oriented programming. This makes code clean and reusable.

Security: Java does not support pointers, it includes built-in protections to keep our programs secure from common problems like memory leakage.

Multithreading: Java programs can do many things at the same time using multiple threads. This is useful for handling complex tasks like processing transactions.

Just-In-Time (JIT) Compiler: Java uses a JIT compiler. It improves performance by converting the bytecode into machine readable code at the time of execution.

## VIDEO TUTORIAL AND PERSONALIZED NOTE SECTION :

The screenshot shows the CodeNexus homepage with a dark header bar. The header includes the 'CodeNexus' logo, navigation links for 'HOME', 'OUR SERVICE', 'FAQS', 'LOGOUT', and a user ID 'ABHIJ07'. Below the header is a large 'WELCOME TO' message. The main content area displays a 2x3 grid of video thumbnail cards:

- Introduction to Java**: An image of a coffee cup with steam, with the text 'Introduction To Java' below it. A 'Watch Video' button is at the bottom.
- CONSTRUCTOR IN JAVA**: An image of a computer monitor displaying code, with the text 'Java Constructor' below it. A 'Watch Video' button is at the bottom.
- JAVA NON-PRIMITIVE DATA TYPES**: An image of a coffee cup with steam, with the text 'Java DataType' below it. A 'Watch Video' button is at the bottom.
- Variables in Java (Practice)**: An image of a code editor window showing Java code, with the text 'Variables in Java (Practice)' below it.
- Access Modifiers with example**: An image of a coffee cup with steam, with the text 'Access Modifiers with example' below it. A 'CODE DECODE' logo is visible.
- Operators in Java**: An image showing various hexagonal icons representing different Java operators like Logical Operators, Assignment Operators, Relational Operators, and Shift Operators.

The screenshot shows a detailed Java tutorial page on the CodeNexus website. The top navigation bar is identical to the previous screenshot. The main content features a large video thumbnail on the left titled 'What is Java? - Java Programming Tutorial'. The thumbnail shows a coffee cup icon and the text 'What is Java? (Java Tutorial)'. Below the thumbnail is a 'Watch on YouTube' button. To the right of the thumbnail is a rich text editor interface with a toolbar at the top and a content area below. The content area contains a paragraph about Java's history, features, and benefits, followed by a summary of its key features and performance improvements through JIT compilation.

## MYNOTES SECTION :



Welcome To Java Notes Section

**INTRODUCTION TO JAVA :-**

Java is a high-level, class-based, object-oriented programming language developed by Sun Microsystems in 1995. It is platform independent, which means we can write code in Java once and run it anywhere using the Java Virtual Machine (JVM). Java is mostly used for building desktop applications, web applications, Android apps, and enterprise systems. Key Features of Java Platform Independent Java is known for its object-oriented programming. This means we can write Java code once, and it will run on any device or operating system without changing anything. Objects share memory and communicate with each other through message passing. Multithreading makes Java programs can do many things at the same time using multiple threads. This is useful for handling multiple users simultaneously. Just-In-Time (JIT) Compiler: Java uses a JIT compiler. It improves performance by converting the bytecode into machine readable code at the time of execution.

**JAVA CONSTRUCTOR :-**

Constructor in Java

**JAVA DATATYPE :-**

(Code) JAVA NON-PRIMITIVE DATA TYPES

## LANGUAGES TOPIC QUIZ SECTION :

**PROGRAMMING LANGUAGES QUIZ**

Following are the available programming languages Quiz  
Start learning today

- JAVA**: Java is a high-level, robust, object-oriented and secure programming language.  Start Quiz
- PYTHON**: Python is used for data analytics, machine learning, and even design.  Start Quiz
- C++**: C++ is used to create computer programs.  Start Quiz
- C**: C is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C language.  Start Quiz
- JAVASCRIPT**: JavaScript (JS) is a scripting language.  Start Quiz
- ANDROID**: An android developer designs software.  Start Quiz

**QUIZ**

- Introduction
- Constructor
- Datatype
- Modifiers
- Operators
- Decision Making
- Looping

**ThinkTank Quiz** Time Left 14

2. Which component is used to compile, debug and execute the java programs?

JRE

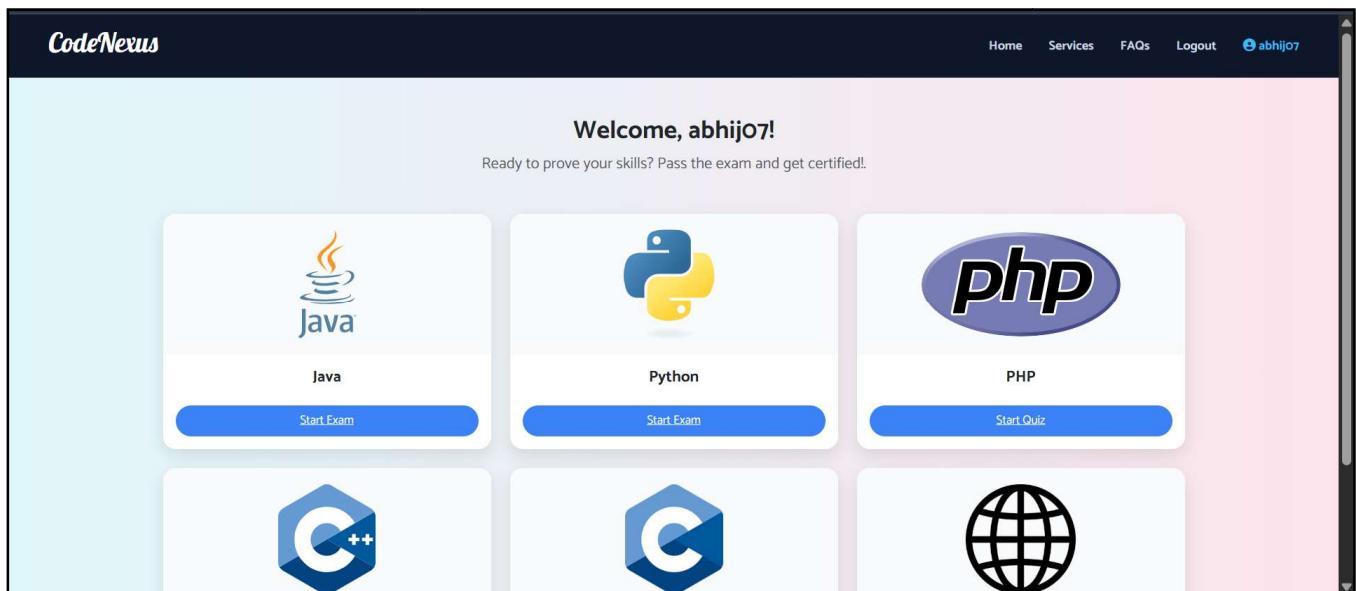
JIT

JVM (X)

JDK (✓)

2 of 5 Questions **Next Que**

## CERTIFICATION SECTION :



The screenshot shows the Java Certification quiz page. The navigation bar includes links for HOME, OUR SERVICE, FAQS, LOGOUT, and a user profile labeled ABHIJ07. The main content area is titled "Java Certification" and contains the following questions:

1. Which of the following option leads to the portability and security of Java?
  - Bytecode is executed by JVM
  - The applet makes the Java code secure and portable
  - Use of exception handling
  - Dynamic binding between objects
2. Which of the following is not a Java features?
  - Dynamic
  - Use of pointers
  - Architecture Neutral
  - Portable
3. \_\_\_\_\_ is used to find and fix bugs in the Java programs.
  - JDK



## ADMIN MODULE :

**CodeNexus**

**Home**

**DASHBOARD**

**MANAGE COURSES**

**MANAGE VIDEOS**

No. Of Courses: 6

No. Of Videos: 4

**List of Users**

Note : User will be directly deleted along with its all data clicking onto Remove

| <b>Id</b> | <b>Username</b> | <b>Password</b> | <b>Email-id</b>             | <b>Mob No</b> | <b>Delete</b>          |
|-----------|-----------------|-----------------|-----------------------------|---------------|------------------------|
| 1         | admin           | admin           |                             | 0             | <a href="#">Remove</a> |
| 15        | abhiji07        | Abhishek@7449   | abhijogdand4@gmail.com      | 0             | <a href="#">Remove</a> |
| 18        | chandrakant45   | Chandrakant@45  | chandrakantshelke@gmail.com | 0             | <a href="#">Remove</a> |

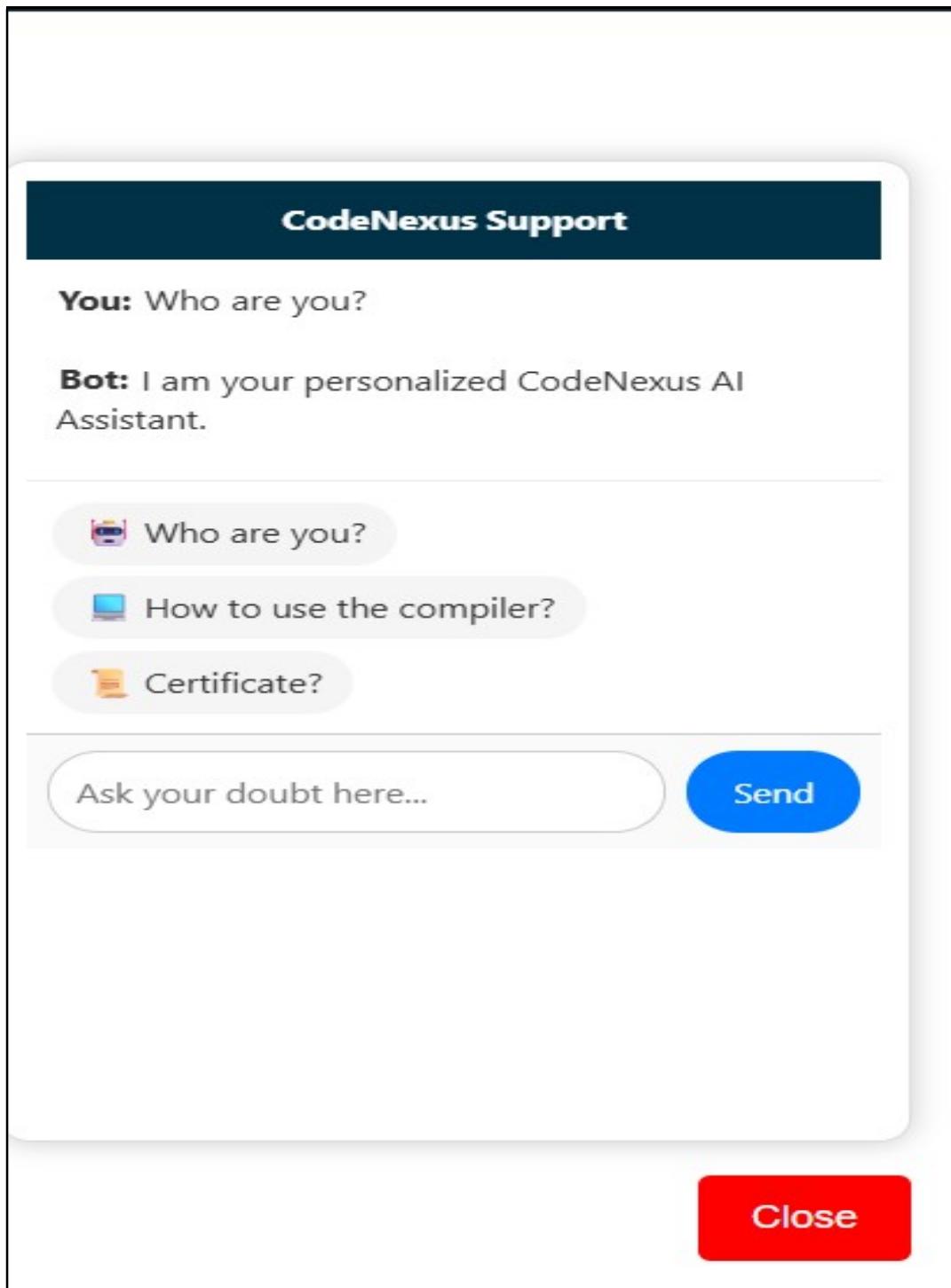
## MANAGE COURSE SECTION :

The screenshot shows the 'Manage Course' section of the CodeNexus platform. On the left, there's a sidebar with links for 'Welcome Admin', 'Manage Courses', 'Manage Videos', and 'Logout'. The main area has tabs for 'Home' and 'Manage Course', with 'Manage Course' being active. A central modal window titled 'ADD NEW COURSE' contains fields for 'Course Name', 'Course Description', and 'Course Image', along with a 'Choose File' button and a 'Submit' button.

## MANAGE VIDEO SECTION :

The screenshot shows the 'Manage Video' section of the CodeNexus platform. The sidebar includes 'Welcome Admin', 'Manage Courses', 'Manage Videos', and 'Logout'. The main content area displays a grid of video thumbnails with titles like 'Introduction to Java', 'CONSTRUCTOR IN JAVA', 'NON-PRIMITIVE DATA TYPES', 'Variables in Java (Practice)', 'Access Modifiers with example', and 'Operators in Java'. Each thumbnail includes a preview image, the video title, a brief description, and 'watch' and 'delete' buttons.

## CODENEXUS SUPPORT CHATBOT :



# Deployment and Maintenance

## 12.1 Installation and Uninstallation

### Frontend (HTML, CSS, JavaScript, Bootstrap)

#### 1. Locate the frontend folder:

Make sure your HTML, CSS, JS, and Bootstrap files are in a project folder like:  
C:\CodeNexus\

#### 2. Run locally:

- o Open index.html directly in a browser, or
- o Use **Live Server** extension in VS Code for auto-reloading.

#### 3. Deploy using IIS/XAMPP/Apache (optional):

- o For **XAMPP**: Move files to C:\xampp\htdocs\CodeNexus\
- o Start Apache via XAMPP control panel.
- o Access the site: <http://localhost/CodeNexus>

---

### Backend (PHP + MySQL)

#### 1. Locate backend folder:

E.g., C:\CodeNexus\

#### 2. Move to XAMPP's web root:

Copy backend files (like index.php, login.php, etc.) into  
C:\xampp\htdocs\CodeNexus\

#### 3. Start local server:

- o Open **XAMPP Control Panel**
- o Start **Apache** and **MySQL**

#### 4. Set up the database:

- o Open <http://localhost/phpmyadmin>
- o Create a new database: codenexus
- o Import your .sql file

#### 5. Update database credentials in PHP files:

```
->$conn = mysqli_connect("localhost", "root", "", "codenexus");
```

#### 6. Access via browser:

<http://localhost/CodeNexus/index.php>

---

### AI-Based Features (Python Flask Microservice using OpenAI API)

#### 1. Prepare Python environment:

- o Go to folder like C:\CodeNexus\ai\_features\
- o Create virtual environment:

```
->python -m venv venv
```

```
->venv\Scripts\activate # For Windows
```

- o Install dependencies:

```
->pip install flask openai flask-cors
```

#### 2. Start AI server:

```
->python app.py
```

#### 3. Connect frontend or PHP via HTTP request to Flask service

(e.g., <http://localhost:5000/summarize>)

## Uninstallation

- **Frontend:**
  - Delete the frontend folder from your PC.
  - Remove files from XAMPP's htdocs if used.
- **Backend:**
  - Stop Apache and MySQL from XAMPP.
  - Delete backend PHP files.
  - Drop database from phpMyAdmin.
- **AI Python Service:**
  - Deactivate virtual environment:

->venv\Scripts\deactivate

- Delete the Python files/folder.
- Uninstall packages (optional):

->pip uninstall flask openai flask-cors

## 12.2 User Help

### Web Application:

Open the local URL in any browser:

<http://localhost/CodeNexus> or open index.html directly.

### AI Features (Chatbot / Text Summarizer):

Use designated form inputs that interact with Flask routes.

## Troubleshooting Common Issues

- **PHP not working?** Make sure Apache is running in XAMPP.
- **Database errors?** Check MySQL is running and phpmyadmin is accessible.
- **AI Feature not responding?** Ensure Flask is running at localhost:5000.
- **CORS errors (frontend ↔ Flask)?** Use flask-cors in Python backend:

from flask\_cors import CORS

CORS(app)

## Other Notes

- **Future Mobile Version:** You can add a mobile-friendly frontend or a React Native app later.
- **OpenAI API Key Handling:** Store securely using .env or config files, don't hardcode in public code
- **Data Visualization:** Chart.js can be added to the frontend for quiz or usage stats.

## Component Deployment and Maintenance

| Component              | Deployment Platform      | Maintenance Summary  |
|------------------------|--------------------------|--|
| Frontend (HTML/CSS/JS) | IIS / Apache (Localhost) | Check manually monthly; update static files as needed; test on multiple browsers |
| Backend (PHP + MySQL)  | XAMPP (Localhost)        | Monitor error logs weekly; secure user inputs; update PHP version quarterly      |
| Python Microservices   | AI Flask on Localhost    | Monitor performance; rotate API keys; update dependencies quarterly              |
| Future Mobile App      | Optional                 | Plan future strategy if mobile features are added                                |

# Conclusion and Future Scope

## 10.1 Conclusion

This project is taking into consideration various techniques that can be used to make a successful “Online programming language learning Platform” for the web. We also learned the languages and technologies used to obtain the goal of our project. We have successfully designed the application using the various technologies.

We are making the use of different techniques and try to minimize the time required and quality of education experience in online learning based platform. Also in this project we are trying to include the important and advance features that make the learning journey of the user easy and enjoyable.

The facilities which we provide students to learn programming languages in more efficient way under a single roof where they can learn from text based, video based tutorial and enhance their skills competing in quiz's

Opening up borders of programming language learning 24-7 with latest and trending topics loaded with detail descriptions and providing implementable infrastructure.

## 10.2 Future Scope

❖ Our Project technologies can be extended in the following future scope.

➤ Live instruction:

Certain curriculum may require specialized instructors. By using live broadcasts, these instructors can remain in one location and provide instruction to many students in other locations. This type of specialization increases as students move into higher levels of education, for example towards advanced degrees in medicine

➤ Video content delivery:

Pre-recorded content such as lectures, documentaries, and other video content may be delivered in a forward and store model so that the material can be viewed when needed which are created by our dedicated faculties which we hire

- Student-to-student interactions (video conferencing):

Related to the first point, students may learn just as much from each other as they do from teachers. Thus, communications technology can be used to connect students in different regions or even different parts of the world so that they may interact.

- Industry recognizes certifications:

Providing certification courses which provide industry standard and recognized certificate course. Which can be used to show great background and skill in that particular field

- Up-to-date materials:

As mentioned above, the basics seldom change. However, virtually all textbooks must be updated. Textbooks are very expensive to purchase, maintain, and deliver. Again, digital delivery solves this issue when coupled with e-Readers such as tablets.

- Premium subscription-based learning resources:

Providing some additional feature like personal assistance, video-based doubt solving session, interactive classes with students, practice sets etc. Which will enhance students' ability to code in more efficient way.

# References

1. Google for searching
2. <https://www.php.net/manual/en/>
3. <https://www.tutorialspoint.com/php/>
4. <https://www.w3schools.com/php/>
5. <https://stackoverflow.com/>
6. <https://www.geeksforgeeks.org/php-statet/>
7. <https://www.youtube.com/watch?v=tiLWCNFzThE>
8. <https://getbootstrap.com/>
9. <https://codewithharry.com/videos/php-mysql-tutorials-hindi-0>
10. <https://www.mysql.com/>
11. <https://en.wikipedia.org/wiki/PHP>
12. <https://www.quora.com/>
13. <https://github.com/>

# Laboratory assignments on Project

## Analysis of Algorithmic Design

### Project Analysis of Algorithmic Design

#### 1. Problem Statement

The goal of Codenexus is to provide a real-time programming support system that offers tailored code suggestions, debugging tips, and learning resources based on the user's input and contextual coding habits. The system will utilize real-time data processing and machine learning techniques to deliver personalized code and continuously enhance the user's coding experience.

#### 2. IDEA Matrix Feasibility Assessment

The **IDEA Matrix** helps assess the potential of the project from various angles.

**IDEA Matrix**

| I               | D   | E  | A  |
|-----------------|---|--|--|
| <b>Increase</b> | Increase user engagement by providing real-time personalized coding recommendations based on previous errors, coding styles, and habits.    | Deliver benefits like faster bug resolution and personalized learning experiences. | Enhance user productivity by recommending optimized coding techniques or best practices. |
| <b>Improve</b>  | Improve the accuracy of coding suggestions through adaptive machine learning models that adjust based on user feedback and coding patterns. | Provide better accuracy by analyzing code context and user interaction history.    | Cluster similar coding profiles to suggest advanced solutions based on their expertise.  |
| <b>Ignore</b>   | Avoid recommending irrelevant solutions that don't match the user's coding style or needs.  | Minimize repetitive or overly basic suggestions for more experienced users.        | Remove any bias that could limit the variety of suggested coding resources or languages. |
| <b>Decrease</b> | Reduce response time for code suggestions to ensure real-time support, even for complex code.   | Eliminate unnecessary features that don't contribute to core coding support.       | Avoid overcomplicating the UI to keep the user experience smooth and intuitive.          |

This matrix highlights opportunities for improvement and evaluates the potential impact on user satisfaction and system performance in Codenexus.

### 3. Feasibility Assessment

#### NP-Hard and NP-Complete Considerations:

In the context of **Codenexus**, evaluating and suggesting the best coding solution can be considered an NP-complete problem because the system needs to analyze multiple parameters (e.g., programming language, coding history, syntax patterns, user skill level). Given the vast number of possible code solutions, Codenexus must solve this problem while optimizing for multiple factors, including:

- **Coding efficiency** (e.g., execution speed, code length).
- **User preferences** (e.g., preferred coding style).
- **Real-time feedback** (e.g., processing suggestions quickly).

Using optimization algorithms (e.g., genetic algorithms or heuristics), Codenexus can balance user satisfaction (accuracy of the solution) with computational efficiency (speed of analysis).

#### Satisfiability Issues:

Codenexus must satisfy multiple conditions such as:

- The user's specific code error or request.
- Constraints like language syntax and available learning resources.
- Response time, which is critical for real-time feedback.

# Laboratory assignments on Project Quality and Reliability Testing of Project Design

## Project Quality and Reliability Testing of Project Design

### 1. Divide and Conquer Strategies

#### Objective:

Leverage distributed/parallel/concurrent processing to handle different aspects of the Codenexus system for efficiency and performance.

- **Divide and Conquer Approach:**

Break the Codenexus system into smaller, manageable modules:

- **User Input Processing:** Handle requests from users (e.g., code snippets, queries).
- **Code Parsing & Analysis Module:** Analyze user input for code errors, optimizations, etc.
- **Feedback Loop:** Continuously improve suggestions and recommendations through feedback from users.

- **Identification of Objects and Morphisms:**

**Objects:** Define key entities in the system

- **User:** The coder interacting with the platform.
- **Code Snippet:** The user's input code.
- **Analysis Result:** The feedback or recommendations generated.
- **Recommendation:** The suggested resources or solutions.

**Morphisms:** Define the relationships between objects:

- **User to Code Snippet:** The relationship between the user and their code submission.
- **Code Snippet to Analysis Result:** Mapping submitted code to its analysis or feedback.

### Function Overloading:

- Identify overloaded functions in your system:
  - `analyzeCode(User user, String codeSnippet)` can be overloaded based on:
    - Different types of input (e.g., short code snippets, full projects).
    - Different analysis modes (syntax checking, optimization suggestions).

### Functional Relations:

- **Venn Diagram:** Show intersections between users with similar coding needs or preferred languages.
- **State Diagram:** Represent the various states of user interaction, such as input code submission, analysis results, and recommendations.

## 2. Functional Dependency Graphs and Software Modeling

**Objective:** Draw functional dependency graphs and use UML diagrams to represent the architecture of Codenexus.

- **Functional Dependency Graphs:**
  - **Input Dependencies:** User data, submitted code snippets, programming language preferences.
  - **Output Dependencies:** Recommendations, solutions, and learning resources based on the analysis.

### UML Diagrams:

- **Use Case Diagram:** Map user interactions, such as code submission, result review.
- **Class Diagram:** Define core entities (e.g., User, CodeSnippet, AnalysisResult) and their relationships.
- **Sequence Diagram:** Show the sequence of steps, from user input to recommendation delivery.

Tools: Lucidchart, draw.io, or Visual Paradigm.

## 3. Testing of Codenexus Problem Statement

### Objective:

Test the Codenexus system using generated test data and appropriate methodologies.

- **Test Data Generation:**
  - Use Python libraries such as NumPy, Pandas, and Faker to simulate various user scenarios (e.g., code snippets, coding history).

### Testing Methodologies:

- **Function Testing Principles:** Identify and test critical functions such as:
  - **Code Analysis Accuracy:** Does the system detect errors effectively?
- **Black Box Testing:** Verify outputs (e.g., analysis results) based on input without inspecting the internal logic.

### Reliability Testing of UML Diagrams:

- Validate the accuracy of UML diagrams to ensure they reflect the actual system design and interactions.

## 4. Test Cases for Identified Functions

### Objective:

Write test cases for each function using Black Box Testing principles.

### Example Test Cases:

- **Function:** analyze Code (User user, String codeSnippet)
  - **Test Case 1:**
    - **Input:** User with no coding experience, simple "Hello World" snippet.
    - **Expected Output:** Basic analysis with no errors.
  - **Test Case 2:**
    - **Input:** Experienced user, complex algorithm with errors.
    - **Expected Output:** Detailed feedback with specific suggestions for improvement.
- **Function:** recommend Resources (AnalysisResult result)
  - **Test Case 1:**
    - **Input:** Analysis result with syntax errors.
    - **Expected Output:** Resources related to syntax fixing.
  - **Test Case 2:**
    - **Input:** Analysis result with optimization suggestions.
    - **Expected Output:** Resources focused on optimization techniques.

# Project Planner

## 1. Project Management Tool Selection

**Objective:** Use Jira as the primary project management tool to track tasks, manage project timelines, and monitor progress.

- **Jira:** A powerful tool designed for agile software development. It allows for task assignment, issue tracking, and project visualization through boards and backlogs. Team members can create user stories, assign tasks, set priorities, and track progress easily.

## 2. Project Structure

**Objective:** Define the project structure to ensure clarity in task assignments and responsibilities.

- **Project Phases:**

- **Phase 1:** Requirement Gathering and Analysis
  - Identify user needs and define project scope.
- **Phase 2:** Design and Development
  - Design system architecture; develop modules.
- **Phase 3:** Testing and Quality Assurance
  - Implement testing strategies, generate test data, and conduct reliability testing.
- **Phase 4:** Deployment and Maintenance
  - Deploy the system, monitor performance, and gather user feedback for improvements.

## 3. Task Breakdown

**Objective:** Break down each project phase into specific tasks and assign them to team members in Jira.

Task Breakdown

| Phase                                      | Start Month | Actual Days |
|--|-------------|-------------|
| Literature review and requirement analysis | 25-07-2024  | 30          |
| UI/UX and Frontend Development             | 25-08-2024  | 20          |
| Backend Development (PHP + MySQL)          | 15-09-2024  | 30          |
| AI Feature Development                     | 16-10-2024  | 20          |
| Integration and Testing/Debugging          | 5-11-2024   | 25          |
| Final deployment and documentation         | 30-11-2024  | 30          |

#### 4. Timeline and Milestones

**Objective:** Establish a timeline with milestones to track progress in Jira.

- **Gantt Chart:** Utilize Jira's timeline features or plugins to visualize the project timeline and dependencies between tasks.
- **Milestones:**
  - **Milestone 1:** Completion of requirement gathering (15/09/2024)
  - **Milestone 2:** Completion of design and development (16/10/2024)
  - **Milestone 3:** Completion of testing and quality assurance (05/11/2024)
  - **Milestone 4:** Deployment of the application (30/11/2024)

#### 5. Communication and Collaboration

**Objective:** Facilitate communication and collaboration among team members using Google Meet.

- **Regular Meetings:** Schedule weekly meetings on Google Meet to discuss progress, roadblocks, and next steps. These meetings will help ensure all team members are aligned on project goals and tasks.
- **Documentation:** Use a shared document repository (e.g., Google Drive) to store project documentation, meeting notes, and resources.
- **Communication:** Leverage Google Meet for real-time discussions, enabling team members to collaborate effectively, especially for remote meetings.

#### 6. Risk Management

**Objective:** Identify potential risks and develop mitigation strategies.

- **Risk Assessment:**
  - **Technical Risks:** Delays in algorithm development or integration challenges.
  - **Resource Risks:** Team member availability and expertise.
- **Mitigation Strategies:**
  - Schedule buffer time for development phases in Jira.
  - Ensure cross-training among team members to handle different components

# Reviewers Comments of Paper Submitted

1. Paper Title : Survey On Integrating Ultimate Programming Hub
2. Name of the Conference/Journal where paper submitted: ISTE Approved “NCFCAI-2024” National Conference.
3. Paper accepted/rejected: Accepted.



Paper & Certificate

# Survey on Integrating Ultimate Programming Hub

1<sup>st</sup> Mrs. Shanti Kumaraguru  
*Computer Engineering*  
*D Y Patil college of Engg, Akurdi*  
*Pune, India*  
*skumarguru@dypcoeakurdi.ac.in*

2<sup>nd</sup> Mr. Chandrakant Shelke  
*Computer Engineering*  
*D Y Patil college of Engg, Akurdi*  
*Pune, India*  
*shelkechandrakant7@gmail.com*

3<sup>rd</sup> Mr. Abhishek Jogdand  
*Computer Engineering*  
*D Y Patil college of Engg, Akurdi*  
*Pune, India*  
*abhijogdand4@gmail.com*

4<sup>th</sup> Mr. Shreyash Lohar  
*Computer Engineering*  
*D Y Patil college of Engg, Akurdi*  
*Pune, India*  
*loharshreyash619619@gmail.com*

5<sup>th</sup> Mr. Abhijit Karande  
*Computer Engineering*  
*D Y Patil college of Engg, Akurdi*  
*Pune, India*  
*abhijitkarande80@gmail.com*

Codenexus is an advanced educational platform built to support both students and administrators in the field of programming education. Designed with a user-friendly interface, it empowers students to engage deeply with programming content by providing access to a diverse library of educational videos. Alongside learning, students can store personal notes directly linked to these videos, ensuring that their study materials are centralized and well-organized. The platform also promotes active learning through quizzes designed for each topic, allowing students to assess their understanding as they progress. Upon mastering topics, students can take certification exams, earning credentials that validate their skills and knowledge. In addition, students have a direct line of communication with administrators via the Contact Us section, fostering a supportive learning environment. The platform's Manage Programming Languages section enables administrators to easily add, update, or remove programming languages and their learning topics. This flexibility ensures that Codenexus stays current with the ever-evolving landscape of programming technologies.

**Index Terms**— online assessment, code similarity, programming skills, secure evaluation

## INTRODUCTION

CodeNexus aims to provide a comprehensive e-learning platform that consolidates all necessary resources for both aspiring and seasoned programmers. The platform is designed to offer an all-in-one solution that bridges the gap between learning, practice, and assessment. By integrating modern AI technologies, CodeNexus ensures a personalized learning experience, offering tailored recommendations based on a user's progress and skill level. In addition to AI-driven learning paths, the

platform incorporates interactive content such as coding challenges, real-time feedback, and collaborative tools, fostering an engaging and hands-on environment for learners. CodeNexus also places a strong emphasis on practical assessments, allowing users to test their knowledge in realistic scenarios that simulate real-world programming problems. Administrative features are provided to enable instructors and organizations to manage user progress, create custom content, and track performance analytics, ensuring a streamlined and efficient learning management system. Furthermore, CodeNexus ensures a secure environment for online assessments through advanced techniques like code similarity checks and behavior monitoring, enhancing the credibility and reliability of skill evaluations.

With a focus on a seamless user experience, the platform supports integration with external tools, enabling learners to leverage additional resources such as GitHub repositories, cloud-based development environments, and third-party APIs. CodeNexus is designed to evolve with the ever-changing tech landscape, ensuring that learners are equipped with the most relevant and up-to-date programming knowledge and skills.

## RELATED WORK

In recent years, the surge in e-learning platforms and the demand for scalable, interactive, and secure programming education solutions has led to the development of various tools and systems that aim to meet these needs. CodeNexus stands at the intersection of advanced AI technologies, interactive assessments, and secure online learning environments. Several works in related domains have informed the architecture and

features of CodeNexus, including code similarity detection, user behavior tracking, and personalized learning platforms.

### 1. Online Coding Platforms

Platforms like **HackerRank**, **LeetCode**, **Codeforces**, and **CodeChef** have revolutionized the way developers practice coding and prepare for competitive programming and technical interviews. These platforms provide extensive problem sets, coding challenges, and online assessments, allowing users to evaluate their skills in real-time. However, they often focus solely on problem-solving without incorporating a comprehensive learning management system (LMS) for structured course content. Unlike these platforms, CodeNexus not only offers coding challenges but also integrates video tutorials, interactive lessons, and quizzes in a unified e-learning environment.

### 2. AI in E-Learning

Several platforms, including **Coursera**, **edX**, and **Udemy**, have adopted AI-driven recommendation systems to personalize learning paths based on user progress and behavior. **Coursera's** system, for instance, uses machine learning algorithms to suggest courses based on a user's activity, previous enrollments, and performance. CodeNexus builds upon this idea by incorporating **Natural Language Processing (NLP)** and AI models to provide a more interactive experience. The platform offers personalized content recommendations not only based on what courses users have taken but also on the code they've written, the quizzes they've completed, and their progress through interactive exercises.

NLP models have also been explored for grading written assignments and code reviews. Studies such as **Nguyen et al. (2018)** have demonstrated the use of NLP in detecting plagiarism in programming assignments, while **Mou et al. (2016)** introduced deep learning models that analyze and categorize code snippets. CodeNexus leverages these developments by incorporating NLP into its code similarity detection algorithms to assess the uniqueness of users' submitted solutions, thus ensuring academic integrity.

### 3. Code Similarity Detection and Plagiarism Prevention

The detection of code plagiarism has been an ongoing challenge in the context of programming assessments. **MOSS (Measure of Software Similarity)**, developed at Stanford University, is one of the most widely used tools for detecting plagiarism in code submissions. MOSS compares different submissions, identifying similarities in

structure, logic, and function names while allowing flexibility for minor syntax differences. However, MOSS primarily focuses on post-assessment plagiarism detection.

CodeNexus builds on these techniques, integrating code similarity detection in real-time to prevent academic dishonesty during the assessment process. By employing more advanced machine learning models, CodeNexus can compare submissions, detect subtle similarities, and flag cases where users may have collaborated or copied answers from external sources.

### 4. Interactive Learning and Gamification

Platforms like **Khan Academy** and **Duolingo** have successfully used interactive lessons and gamified elements to engage users in the learning process. Khan Academy, for instance, provides a wealth of educational content, including videos and interactive exercises, while tracking user progress to guide them through personalized learning paths. Similarly, **Duolingo** gamifies the process of learning a language, using small rewards, progress tracking, and leveling systems to keep users motivated.

Inspired by these approaches, CodeNexus integrates interactive content with AI-powered recommendations and real-time feedback to create a dynamic learning environment for programmers. The platform provides a mix of video tutorials, quizzes, and coding challenges, keeping users engaged and motivated while

## 5. Real-Time Code Collaboration and Feedback

Collaborative coding platforms, such as **Replit**, **GitHub Codespaces**, and **Visual Studio Live Share**, have gained popularity in recent years. These platforms allow users to collaborate on code in real-time, share their work with peers, and receive instant feedback from instructors or collaborators. In an educational context, real-time collaboration can enhance peer-to-peer learning, provide immediate assistance, and create an interactive classroom environment.

CodeNexus adopts some of these collaborative features, enabling users to share code with instructors, work on group projects, and receive live feedback. This is particularly beneficial for programming education, where users can learn by solving problems together, discussing their approaches, and refining their solutions in real-time.

## 6. Learning Management Systems (LMS)

Traditional LMS platforms like **Moodle** and **Blackboard** provide a structured environment for managing course content, tracking student progress, and facilitating quizzes and assessments. However, these systems lack the integration of advanced AI and real-time code evaluation systems that are crucial for programming education.

CodeNexus takes LMS functionality a step further by integrating it with a programming-centric ecosystem. It provides seamless course management, quiz creation, user analytics, and assessment tools specific to coding. By building in real-time code evaluation, secure online assessments, and personalized recommendations, CodeNexus combines the best features of an LMS with the demands of modern programming education.

### System Flowchart

#### A. User Enrollment

The user enrollment process is the entry point for users interacting with the CodeNexus platform. This phase involves:

- **Account Creation:** Users provide necessary details such as name, email, and password to create an account.
- **User Verification:** A verification email or SMS may be sent to confirm the user's identity and activate their account.
- **Initial Setup:** Users may be guided through an onboarding process to set up their profiles, select preferences, and familiarize themselves with the platform's features.

This initial step establishes the user's identity and prepares the platform to offer personalized experiences and track their progress effectively.

#### B. User Dashboard

The user dashboard is a central hub where users interact with various features of CodeNexus:

- **AI-Driven Recommendations:** Leveraging AI algorithms, the dashboard presents personalized course and content suggestions based on user behavior, past interactions, and performance (Burke, 2002). For example, if a user frequently engages with Python-related content, the system might recommend advanced Python courses or relevant coding challenges.
- **Progress Tracking:** Users can view their progress in different courses, track completed modules, and check their performance metrics.

This dashboard ensures users receive a tailored learning experience, enhancing their engagement and learning outcomes.

#### C. Learning Modules

CodeNexus provides a variety of learning resources, which are crucial for user education:

- **Video Tutorials:** High-quality instructional videos cover various programming topics, offering visual and auditory learning experiences.
- **Interactive Lessons:** Lessons include hands-on coding exercises and practical examples that allow users to apply what they've learned.
- **Quizzes:** Quizzes are designed to test knowledge and reinforce learning. They provide instant feedback to help users understand their strengths and areas for improvement.

**Gamification Elements** (Mayer, 2005; Deterding, 2011) such as badges, achievements, and progress bars are integrated to motivate users and make learning more engaging. These elements create a sense of accomplishment and encourage continued participation.

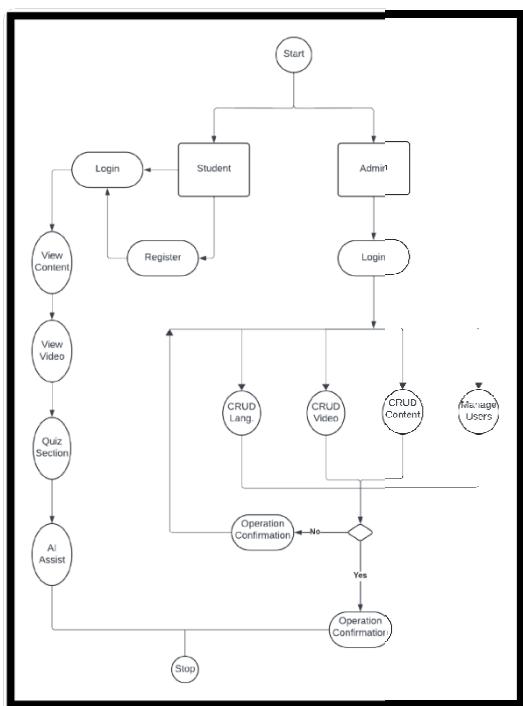


Fig.1 Flowchart

#### D. Code Challenges

CodeNexus incorporates several features to enhance the coding practice experience:

- **Real-Time Code Evaluation:** As users submit their code, the system evaluates it instantly, providing immediate feedback on correctness, efficiency, and style. This real-time feedback helps users learn and improve rapidly.

#### E. Collaborative Features

Collaboration is an essential aspect of modern programming education:

- **Real-Time Code Collaboration:** Users can work together on coding projects, share their screens, and collaborate in real-time, fostering peer-to-peer learning and teamwork.
- **Live Feedback:** Instructors and peers can provide immediate feedback on shared code, facilitating interactive discussions and quick resolution of issues (Gupta, 2021).

These features enhance the learning experience by promoting collaboration and providing real-time assistance.

#### F. Assessment Security

Ensuring the integrity of assessments is

critical:

- **Eye-Tracking Technology** (Morency et al., 2010): This technology monitors users' eye movements during assessments to detect unusual behavior, such as looking away from the screen or referencing unauthorized materials.
- **Behavioral Analysis:** Analyzes patterns of behavior to identify suspicious activities or potential cheating attempts.

These security measures help maintain the credibility of the assessment process and ensure that users complete assessments honestly.

#### G. LMS Integration

CodeNexus combines features of a Learning Management System (LMS) with additional programming-focused functionalities:

- **Course Management:** Facilitates the creation, organization, and management of courses, including scheduling, content delivery, and student enrollment.
- **Real-Time Code Evaluation:** Unlike traditional LMS platforms, CodeNexus integrates real-time evaluation tools to assess coding tasks instantly.
- **Personalized Recommendations** (Anderson, 2018): The platform provides tailored content suggestions and feedback based on user performance and interactions.

#### Literature Review

CodeNexus represents a significant advancement in programming education by integrating advanced technologies and methodologies into a cohesive learning platform. This literature review explores the foundational concepts and innovations that inform the development of CodeNexus, highlighting key studies and technologies that contribute to its functionality and effectiveness.

#### Integration of Cloud Computing

[1]Ali, Mohamed, and Al-Nuaim (Year) discuss the integration of cloud computing with e-learning systems in their work, "Towards Intelligent E-Learning Systems." They emphasize how cloud computing enhances e-learning platforms through cost efficiency, scalability, and flexibility. Cloud technologies enable CodeNexus to offer a robust and scalable infrastructure, facilitating seamless access to resources, scalable storage, and dynamic content

delivery. This integration supports large-scale user interactions and resource management, essential for a comprehensive programming education platform.

#### Personalized Learning with GPT Technology

[6] In "GPT-Empowered Personalized eLearning System for Programming Languages," Johnson, Gonzalez, and Smith (Year) explore the use of GPT technology to personalize e-learning experiences. By leveraging generative pre-trained transformers (GPT), CodeNexus can dynamically generate tailored course content and automate various aspects of the learning experience. GPT technology allows for the creation of personalized learning paths based on individual progress, coding practices, and performance. This capability enhances user engagement by adapting the learning material to meet the specific needs of each learner, thereby improving educational outcomes.

#### Recommender Systems

[3] Burke (2002) provides a comprehensive guide on recommender systems, crucial for personalizing user experiences. Recommender systems use algorithms to suggest relevant content based on user data and preferences. In CodeNexus, recommender systems play a pivotal role in curating personalized content recommendations, including coding challenges, video tutorials, and interactive lessons. This personalization helps users discover resources aligned with their learning goals and interests, fostering a more engaging and effective learning environment.

#### Gamification in Education

[4] Deterding (2011) discusses gamification in education, highlighting how game design elements can enhance student motivation and engagement. CodeNexus incorporates gamification features such as badges, leaderboards, and progress tracking to incentivize participation and achievement. By integrating these elements, CodeNexus creates a motivating and interactive learning environment that encourages users to actively engage with the content and strive for continuous improvement.

#### Code Similarity Detection

[5] Evans (2002) introduces MOSS (Measure of Software Similarity), a tool for detecting code similarity and plagiarism. MOSS is widely used to identify similarities in code submissions, helping to maintain academic integrity. CodeNexus enhances this approach by incorporating real-time code similarity detection using advanced machine learning models. This real-time detection helps prevent academic

dishonesty during assessments by identifying subtle similarities and potential plagiarism, ensuring a fair evaluation process.

#### Real-Time Collaboration

[7] Gupta (2021) explores real-time collaboration in cloud-based coding environments, which is essential for interactive learning. CodeNexus supports real-time collaboration features that allow users to work together on coding projects, share code, and provide immediate feedback. This collaborative environment enhances peer-to-peer learning, facilitates group problem-solving, and fosters a dynamic and interactive educational experience.

#### Dynamic Assessment and Feedback

[8] Hegedus (2020) reviews dynamic assessment methods in programming education, emphasizing the importance of real-time feedback and adaptive assessments. CodeNexus implements dynamic assessment techniques to provide immediate feedback on coding challenges and quizzes. This approach allows learners to receive real-time insights into their performance, helping them to identify areas for improvement and adjust their learning strategies accordingly.

#### Interactive Learning Environments

[9] Mayer (2005) examines the impact of interactive learning environments on student performance. Interactive elements, such as multimedia content and hands-on activities, are shown to improve understanding and retention. CodeNexus integrates interactive video tutorials, coding exercises, and quizzes to create an engaging learning experience. These interactive components help users grasp complex programming concepts and apply their knowledge in practical scenarios.

#### Natural Language Processing for Plagiarism Detection

[10] Nguyen, Liu, and Lo (2018) focus on using natural language processing (NLP) to detect plagiarism in programming assignments. NLP techniques analyze code submissions to identify similarities and ensure originality. CodeNexus applies advanced NLP models to assess code similarity and detect potential plagiarism, maintaining academic integrity and fostering a fair learning environment.

#### Deep Learning for Code Analysis

[11] Mou, Xu, and Zhang (2016) investigate deep learning models for analyzing and categorizing code snippets. Deep learning techniques offer advanced capabilities for code analysis and classification. CodeNexus utilizes deep learning models to enhance code evaluation processes, providing more accurate assessments of code quality and functionality.

**I.****CONCLUSION**

CodeNexus represents a significant advancement in the field of programming education by integrating a diverse array of advanced technologies and methodologies to create an all-encompassing learning environment. By leveraging cloud computing, CodeNexus ensures scalability and flexibility, accommodating a large user base and managing extensive educational resources efficiently. The platform's use of GPT technology allows for a highly personalized learning experience, tailoring content and recommendations to individual user profiles, which enhances engagement and learning outcomes. Incorporating sophisticated recommender systems and gamification elements, CodeNexus fosters an engaging and motivating learning environment. These features not only make the learning process enjoyable but also drive sustained user participation and achievement. The platform also emphasizes academic integrity through real-time code similarity detection and natural language processing, which help maintain fairness and originality in assessments..

Additionally, CodeNexus supports real-time collaboration and dynamic assessment, allowing users to engage in collaborative coding projects and receive immediate feedback. This interactive approach enhances peer learning and provides opportunities for users to refine their skills. The integration of interactive learning environments, advanced security measures like eye-tracking,

and deep learning for code analysis further enrich the user experience and ensure a secure and effective educational process. Overall, CodeNexus synthesizes these technologies to offer a cutting-edge programming education hub that addresses modern e-learning challenges. As the field of e-learning continues to evolve, CodeNexus is well-positioned to adapt and expand, ensuring it remains a leading resource for programming education and a valuable tool for fostering learning and development in the digital era.

**References**

- [1] Anderson, M. J. (2018). Moodle: The world's most popular learning management system. Educational Technology Publications.
- [2] Ali, A., Mohamed, S., & Al-Nuaim, K. (Year). Towards intelligent e-learning systems. [Journal Name, Volume(Issue), Page Numbers]. [Publisher].
- [3] Burke, R. (2002). A practical guide to recommender systems. Springer.
- [4] Deterding, S. (2011). Gamification in education: What, how, why bother?. Springer.
- [5] Evans, D. A. (2002). MOSS: A tool for detecting software similarity. Stanford University.
- [6] GPT-Empowered Personalized eLearning System for Programming Languages. Johnson, A., Gonzalez, M., & Smith, L. (Year). [Journal Name, Volume(Issue), Page Numbers]. [Publisher].
- [7] Gupta, S. (2021). Real-time collaboration in cloud-based coding environments. ACM Transactions on Software Engineering.
- [8] Hegedus, M. (2020). Dynamic assessment in programming: A review. Journal of Computer Science Education.
- [9] Mayer, R. E. (2005). The impact of interactive learning environments on student performance. Educational Psychology Review.
- [10] Morency, L.-P., Schuller, B., & Pantic, M. (2010). Using eye-tracking to enhance online assessment security. IEEE Transactions on Affective Computing.
- [11] Nguyen, T., Liu, X., & Lo, D. (2018). Detecting plagiarism in programming assignments using natural language processing. IEEE Transactions on Software Engineering.
- [12] Mou, L., Xu, J., & Zhang, X. (2016). Deep learning models for analyzing and categorizing code snippets. IEEE Transactions on Neural Networks and Learning Systems.

# Plagiarism Report

 iThenticate® Page 2 of 70 - Integrity Overview Submission ID trn:oid::3618:99989845

## 14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

| Match Groups   |  | Top Sources |  |
|--|--|-------------|--|
|  90 | Not Cited or Quoted 12%<br>Matches with neither in-text citation nor quotation marks | 14%         |  Internet sources                 |
|  0  | Missing Quotations 0%<br>Matches that are still very similar to source material      | 2%          |  Publications                     |
|  15 | Missing Citation 2%<br>Matches that have quotation marks, but no in-text citation    | 3%          |  Submitted works (Student Papers) |
|  0  | Cited and Quoted 0%<br>Matches with in-text citation present, but no quotation marks |             |  |

### Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

**Dr.Shanthi Kumaraguru**  
**final\_report(2).pdf**

 D Y Patil College of Engineering, Akurdi

---

#### Document Details

|                               |                   |
|-------------------------------|-------------------|
| Submission ID                 | 64 Pages          |
| trn:oid::3618:99989845        | 12,764 Words      |
| Submission Date               | 86,260 Characters |
| Jun 9, 2025, 1:05 PM GMT+5:30 |                   |
| Download Date                 |                   |
| Jun 9, 2025, 1:08 PM GMT+5:30 |                   |
| File Name                     |                   |
| final_report(2).pdf           |                   |
| File Size                     |                   |
| 1.2 MB                        |                   |

 iThenticate® Page 1 of 70 - Cover Page Submission ID trn:oid::3618:99989845

# Information of Project Group Members

1. Name : Chandrakant Shelke
2. Date of Birth : 25/07/2003
3. Gender : Male
4. Permanent Address : At post Kumbhephal ,Tal: Sambhajinagar , Dist: Sambhajinagar .
5. E-Mail : [shelkechandrakant7@gmail.com](mailto:shelkechandrakant7@gmail.com)
6. Mobile/Contact No. : 9146844236
7. Placement Details : Placed with LTIMINDTREE
8. Paper Published : Yes.

1. Name Abhishek Vithalrao Jogdand
2. Date of Birth:07/06/2001
3. Gender : Male
4. Permanent Address: Mangal Vihar Society,Beside PMPML Bus Depot,Nigdi,Pune 41103
5. E-Mail : [abhijogdand4@gmail.com](mailto:abhijogdand4@gmail.com)
6. Mobile/Contact No. :7387585453
7. Placement Details : In Progress
8. Paper Published : Yes.

1. Name : Shreyash Manikant Lohar
2. Date of Birth : 13/09/2003
3. Gender : Male
4. Permanent Address : Arihant heights, ft no. 402, Mohan Nagar chinchwad, pune 411019
5. E-Mail : [loharshreyash619619@gmail.com](mailto:loharshreyash619619@gmail.com)
6. Mobile/Contact No. : 9960877355
7. Placement Details : Placed with LTIMINDTREE
8. Paper Published : Yes.

1. Name :Abhijit Rajendra Karande
2. Date of Birth : 10/06/2003
3. Gender : Male
4. Permanent Address : Moshi Pradhikaran, Pimpri-Chinchwad, Maharashtra 411039
5. E-Mail : [abhijitkarande80@gmail.com](mailto:abhijitkarande80@gmail.com)
6. Mobile/Contact No. : 8080069211
7. Placement Details : In Progress
8. Paper Published : Yes.