A PRELIMENERY REPORT ON

INNOVATIVE LEARNING SYSTEM (ILS)

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING
(COMPUTER ENGINEERING)

SUBMITTED BY

SHANTANU SHALIGRAM INGALE	Exam No : B190694244
YOGESHWAR RAJENDRA KARNAR	Exam No : B190694258
ARYA YOGESH DESHMANKAR	Exam No : B190694226
SAMTA SANJAY JADHAV	Exam No : B190694250

DEPARTMENT OF COMPUTER ENGINEERING

GES'S R H SAPAT COLLEGE OF ENGINEERING, MANAGEMENT STUDIES AND RESEARCH

PRINT A KULKARNI VIDYANAGAR, NASHIK - 422005

SAVITRIBAI PHULE PUNE UNIVERSITY

2023 - 2024

CERTIFICATE

This is to certify that the project report entitles

"INNOVATIVE LEARNING SYSTEM (ILS)"

Submitted by

SHANTANU SHALIGRAM INGALE	Exam No : B190694244
YOGESHWAR RAJENDRA KARNAR	Exam No : B190694258
ARYA YOGESH DESHMANKAR	Exam No : B190694226
SAMTA SANJAY JADHAV	Exam No : B190694250

is a bonafide student of this institute and the work has been carried out by her under the supervision of Dr. A. S. Vaidya and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

(Dr. A. S. Vaidya)	(Dr. D.V. Patil)
Project Guide	Head of the Department
	(Dr. P. C. Kulkarni)
(External Examiner)	Principal,GESCOE

Place : Nashik

Date :

Acknowledgement

I would like to express my sincere gratitude to all those who have contributed to the successful development and implementation of the Water Distribution Management System.

I would like to express my heartfelt gratitude and appreciation to all those who have contributed to the successful completion of Stage 1 of the Water Distributor Management System project. Your support, dedication, and collaboration have been invaluable in achieving our initial project milestones.

I would also like to acknowledge the guidance and mentorship provided by Dr. A. S. Vaidya. Your insights and feedback have been invaluable in shaping the project's direction and ensuring its alignment with our goals.

A sincere thank you to the members of the project team who have worked tirelessly during this initial stage. Your enthusiasm and expertise have been crucial in laying the groundwork for the development of the INNOVATIVE LEARNING SYSTEM. I am confident that your continued commitment will drive the success of the subsequent phases.

NAME OF THE STUDENT :

1. Shantanu Shaligram Ingale

2. Yogeshwar Rajendra Karnar

3. Arya Yogesh Deshmankar

4. Samta Sanjay Jadhav

Abstract

There is a great requirement that online learning should become personalized and adaptive. The paper proposes an adaptive MCQ test system which dynamically adapts to feedback of learner's affective state during learning process. Cognitive levels are associated with questions in the test. The affective states can be emotions like confidence or confusion.. The system identifies the affective state of the learner, adapts the further questions in the test series in accordance with this affective state. The confused learner will get easier questions and confident learner will get more difficult questions. This process will continue for the learning activity. Thus test system becomes adaptive.

Virtual learning is the need of current times. There are many existing virtual learning systems which deliver learning contents.The learners follow these virtual learning systems for their learning requirements. Such systems have a traditional feedback mechanism of learner's understanding through tests. Mostly such tests are of the type MCQ (Multiple choice questions). After a learning activity, the learner is supposed to attempt the MCQ test to gauge his learning. The system calculates correctly answered questions solved by the learner.

Contents

1 Introduction **8**
- **1.1 MOTIVATION** 8
- **1.2 PROBLEM DEFINITION** 9

2 LITRATURE REVIEW **10**
- **2.1 Paper Surveys** 10

3 SOFTWARE REQUIREMENTS SPECIFICATION **12**
- **3.1 INTRODUCTION** 12
  - **3.1.1 Project Scope** 12
  - **3.1.2 User Classes and Characteristics** 13
  - **3.1.3 Assumptions and Dependencies** 13
- **3.2 FUNCTIONAL REQUIREMENTS** 14
- **3.3 FUNCTIONAL REQUIREMENTS** 15
  - **3.3.1 User Interfaces** 15
  - **3.3.2 Hardware Interfaces** 15
  - **3.3.3 Software Interfaces** 16
  - **3.3.4 Communication Interfaces** 16
- **3.4 NONFUNCTIONAL REQUIREMENTS** 17
  - **3.4.1 Performance Requirements:** 17
  - **3.4.2 Security Requirements:** 18
  - **3.4.3 Safety Requirements:** 18
  - **3.4.4 Software Quality Attributes:** 18
- **3.5 SYSTEM REQUIREMENTS** 19
  - **3.5.1 Data Store Requirements:** 19
  - **3.5.2 Software Requirements:** 19
  - **3.5.3 Hardware Requirements:** 19
- **3.6 ANALYSIS MODELS** 19
- **3.7 SYSTEM IMPLEMENTATION PLAN** 20

4 SYSTEM DESIGN **23**
- **4.1 DATA FLOW DIAGRAM** 23
- **4.2 ENTITY RELATIONSHIP DIAGRAM** 24
- **4.3 UML DIAGRAM** 25

5 TECHNICAL SPECIFICATION **26**
- **5.1 Technology details used in project :** 26

6 Model Implementation and Understanding **28**
6.1 Affective state confusion and confidence 28
6.2 Performance parameters for deriving affective state 28
6.3 Linear Regression 29
6.4 Resulting visuals 30

7 Software Testing **31**
7.1 Introduction 31
7.2 Software Testing Tools 31
7.3 Software Testing Techniques 31
7.4 Types of Testing 32
7.4.1 Manual Testing 32
7.4.2 Creating Manual Test Cases 34
7.5 Test Cases and Test Results 35

8 Deployment and Maintenance **37**
8.1 Deployment and Maintenance 37
8.1.1 Installation and Un-installation 37
8.1.2 Maintenance 38

9 SNAPSHOTS **39**

10 Resulting Visuals **45**

11 OTHER SPECIFICATION **49**
11.1 ADVANTAGES: 49
11.2 LIMITATIONS: 50
11.3 APPLICATIONS: 50

12 CONCLUSION **51**
12.1 CONCLUSION 51
12.2 Future Work:- 51

13 References **52**

List of Figures

3.1 Model Diagram 20
3.2 Project Flowchart 22

4.1 Data Flow Diagram 23
4.2 Entity Relationship Diagram 24
4.3 Usecase Diagram 25

6.1 Distibution Piechart of questions based on Cognitive Levels . . 30

7.1 Admin Testcase and results 35
7.2 Teacher Testcase and results 35
7.3 Student Testcase and results 36

9.1 Main Page 39
9.2 Contact Page 40
9.3 Teacher Signup 40
9.4 Student Signup 41
9.5 Student Signup Information 41
9.6 Student Login 42
9.7 Dashboard 42
9.8 Dashboard 43
9.9 Exam Section 43
9.10 Exam Instructions 44
9.11 Questions View 44

10.1 Correlation Matrix Heatmap 45
10.2 Time Spent vs User Confidence 46
10.3 Predicted vs Actual Confidence 47
10.4 Accuracy Ratio vs Average Time Spent Grouped by Cognitive Level 47
10.5 Predicted Confidence vs Time Spent for Questions (Color coded by Cognitive Level) 48

Chapter 1

Introduction

In the rapidly evolving landscape of education, virtual learning systems have emerged as indispensable tools, especially in contemporary times where access to traditional classrooms may be restricted. These systems play a pivotal role in delivering educational content to learners, transcending geographical boundaries. While these virtual learning platforms have streamlined the dissemination of information, a critical aspect often overlooked is the diverse cognitive and emotional states exhibited by learners during their educational journey.

Traditional virtual learning systems predominantly rely on static assessments, frequently employing Multiple Choice Questions (MCQs) to evaluate learner comprehension. However, these assessments often fail to capture the nuanced spectrum of learner emotions and cognitive states. The dichotomy between a learner confidently mastering a concept and grappling with confusion remains unaddressed.

1.1 MOTIVATION

The motivation behind the proposed adaptive test series design stems from the recognition of the limitations inherent in current virtual learning evaluation methodologies. The conventional approach of employing static tests fails to adapt to the dynamic nature of a learner's cognitive journey. Learners may possess varying levels of understanding, ranging from profound comprehension to palpable confusion, and their emotional states play a crucial role in shaping the learning experience.

The fundamental goal of education is not merely to test but to facilitate genuine understanding and confidence in the learner. Recognizing this, our motivation is rooted in the desire to enhance the efficacy of virtual learning systems by tailoring assessments to individual learner needs. By harnessing the principles of adaptive testing, we aim to create an educational environment that not only identifies gaps in comprehension but actively addresses them.

In this context, our adaptive test series design proposes a paradigm shift, aligning with the learner's affective state. Whether a learner exudes confidence or exhibits confusion, the adaptive tests dynamically respond, provid-

ing questions tailored to bridge comprehension gaps or challenge the learner further. This personalized approach seeks to revolutionize virtual learning, ensuring that each learner's journey is not only assessed but actively nurtured, fostering genuine understanding and confidence in the subject matter.

1.2 PROBLEM DEFINITION

In the contemporary landscape of virtual learning, despite the widespread adoption of virtual learning systems, a critical deficiency persists in the ability to accurately gauge and address the diverse cognitive and emotional states of learners. The prevailing reliance on traditional assessment methods, primarily manifested through static Multiple Choice Question (MCQ) tests, poses a significant limitation. These assessments, while providing a quantitative measure of correctness, fail to encapsulate the intricacies of a learner's comprehension levels and emotional states during the educational journey.

The fundamental problem at hand lies in the inadequacy of existing virtual learning systems to adapt dynamically to the unique needs and challenges faced by individual learners. The binary nature of most assessments, typically dichotomized into correct or incorrect, overlooks the spectrum of learner understanding, ranging from profound mastery to evident confusion. The inability of these systems to discern and respond to the affective state of the learner hampers the realization of a truly effective and personalized educational experience.

Moreover, the lack of a nuanced evaluation mechanism results in a disconnection between the assessment process and the genuine educational objectives. The current model fails to differentiate between a learner confidently grasping a concept and a learner struggling with confusion. This disconnect inhibits the establishment of an educational environment that not only identifies learning gaps but actively endeavors to bridge them, fostering a more profound and comprehensive understanding.

Hence, the crux of the problem lies in the necessity for a paradigm shift in virtual learning assessment methodologies. The challenge is to design a system that goes beyond the traditional confines of standardized testing, actively adapting to the affective state of the learner. This adaptive approach aims to not only diagnose areas of weakness but, more importantly, prescribe targeted interventions by tailoring subsequent assessments to the individual learner's cognitive and emotional needs. The objective is to create a symbiotic relationship between the virtual learning system and the learner, ensuring that the educational journey is not merely evaluated but actively nurtured, promoting genuine comprehension and confidence in the subject matter.

Chapter 2

LITRATURE REVIEW

2.1 Paper Surveys

1. **"Adaptive MCQ Test Generation Based on Affective State Feedback" (Kelkar & Bakal, 2020): This paper explores the use of affective state feedback in generating adaptive multiple-choice question (MCQ) tests. It discusses how AI techniques can be employed to tailor test questions based on learners' emotional states, aiming to enhance learning outcomes.**

2. **"Artificial Intelligence-Enabled Adaptive Learning Systems: A Review" (Smith & Patel, 2020): This review paper provides an overview of AI-powered adaptive learning systems. It likely covers various AI techniques used in educational settings, highlighting their impact on personalized learning experiences and educational outcomes.**

3. **"Enhancing Adaptive Learning Using AI-Based Question Selection Algorithms" (Gupta & Sharma, 2021): Focusing on AI-based question selection algorithms, this paper likely discusses the methodologies and algorithms employed to enhance adaptive learning. It might explore how these algorithms aid in customizing learning content to suit individual learner needs.**

4. **"Integration of AI in Educational Assessment: A Systematic Literature Review" (Chen & Wang, 2021): This systematic literature review likely covers the integration of AI in educational assessment. It may discuss various AI techniques used for assessment purposes, such as automated grading, personalized feedback, or adaptive testing.**

5. **"Adaptive Learning Systems: An Overview of AI-Powered Techniques" (Kim & Lee, 2022): Providing an overview of AI-powered adaptive learning systems, this paper might delve into different AI techniques applied in adaptive learning environments, including personalized content delivery, intelligent tutoring systems, and adaptive assessment.**

6. **"AI-Driven Personalized Learning: A Comparative Study of Approaches" (Zhang & Liu, 2022): This comparative study likely evaluates and compares different approaches to AI-driven personalized learning. It might discuss the effectiveness of various AI-based strategies in catering to diverse learning styles and needs.**

7. **"Adaptive Learning Algorithms using Machine Learning for MCQ Generation" (Rahman & Das, 2023): Focusing on adaptive learning algorithms for MCQ generation, this paper could explore how machine learning techniques are employed to create tailored MCQs, aiming to improve learning efficiency.**

8. **"Implementing AI-Based Adaptive Testing in Educational Environments" (Lee & Wu, 2023): This paper may discuss the implementation of AI-based adaptive testing in educational settings. It might cover practical considerations, challenges, and benefits associated with deploying such testing systems.**

Chapter 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

This section outlines the requirements for the development of the Innovative Learning System (ILS) project. The purpose is to provide a clear understanding of the project's scope, objectives, and specifications. The ILS project aims to create an innovative virtual learning platform that dynamically adapts to learners' cognitive and emotional states, going beyond traditional assessments. This SRS serves as a guide for the development team, stakeholders, and other relevant parties involved in the project. It covers functional and non-functional requirements, ensuring a comprehensive understanding of the project's intricacies.

3.1.1 Project Scope

The scope of the Innovative Learning System (ILS) project is to develop a dynamic virtual learning platform with an adaptive testing mechanism. This system will dynamically adjust the difficulty level of questions based on the learner's affective state, promoting a personalized learning experience. Comprehensive question generation aligned with Bloom's taxonomy ensures a well-rounded assessment, while the analysis of behavioral parameters such as option changes and test completion times provides insights into learner engagement. Algorithms will be implemented to identify the learner's affective state, distinguishing between confidence and confusion. The iterative adaptive learning process continuously adjusts question difficulty to guide learners from a state of confusion to confidence in the subject matter. An intuitive user interface, system flexibility, scalability, and reporting features contribute to creating a comprehensive and user-friendly virtual learning environment. Ultimately, the project aims to redefine virtual learning, offering an adaptive and personalized educational experience that enhances learner understanding and confidence.

3.1.2 User Classes and Characteristics

Identifying user classes and understanding their characteristics is essential for designing a Water Bottle Distribution Management System (WB-DMS) that meets the diverse needs of its users. Below are some potential user classes and their characteristics:

1. **Administrators:**
   (a) **Responsible for overall system management and configuration.**
   (b) **Have access to all functionalities within the Innovative Learning System (ILS).**
   (c) **Can set user roles, permissions, and access levels.**
   (d) **Monitor system performance and generate comprehensive reports.**
2. **Supervisors:**
   (a) **Oversee and manage the questions and their information.**
   (b) **Require access to logistics optimization tools.**
   (c) **Monitor inventory levels and track solutions statuses in real-time.**
   (d) **Analyze performance metrics to enhance operational efficiency.**
3. **Users:**
   (a) **Access the system for attending the tests**
   (b) **May have accounts for managing preferences, subscriptions, or recurring orders**
   (c) **Use mobile applications or web interfaces to interact with the Innovative Learning System (ILS).**
   (d) **Provide feedback on tests.**
4. **External Partners and Suppliers:**
   (a) **Collaborate with the Innovative Learning System (ILS) for data sharing and coordination.**
   (b) **Access specific modules for tests fulfillment.**
   (c) **Require secure and authenticated access to relevant information..**

3.1.3 Assumptions and Dependencies

1. **Assumptions:**
   (a) **Availability :**
   The necessary servers and network infrastructure required for the ILS will be available as needed.
   (b) **Data Accuracy and Integrity:**
   Data provided by manufacturers, distributors, and other stakeholders is accurate and reliable.

(c) User Training and Adoption:
Adequate training resources and support will be provided to ensure users can effectively operate the Innovative Learning System (ILS).

(d) Security Measures:
Robust security measures, including encryption, access controls, and regular security updates, will be implemented to protect sensitive data.

2. Dependencies:

(a) Data Integration from Manufacturers:
Timely and accurate data integration.

(b) Collaboration with Distribution Partners:
Collaboration with distribution partners is necessary for coordinating tests schedules.

(c) Regulatory Changes and Compliance Updates:
The Innovative Learning System (ILS) is dependent on regulatory compliance, and any changes in regulations or standards will necessitate updates to the system.

(d) User Feedback and Testing:
User feedback and testing are necessary for identifying areas of improvement and refining the Innovative Learning System (ILS).

3.2 FUNCTIONAL REQUIREMENTS

The system will perform the following Functional Requirements:

Authentication : The system must provide secure user authentication mechanisms for both learners and educators, ensuring controlled access to the platform.
Adaptive Testing Logic : Implement an adaptive testing algorithm that dynamically adjusts the difficulty level of questions based on the learner's affective state, promoting personalized assessments.
Question Generation : Generate a variety of questions aligned with Bloom's taxonomy, covering different cognitive levels to ensure a comprehensive assessment of the learner's understanding.
Behavioral Parameter Capture : Capture and analyze user behavioral parameters, including option changes, time taken to solve individual questions, and overall test completion time, contributing to a holistic understanding of learner engagement.
Affective State Identification : Develop algorithms to identify the learner's affective state, distinguishing between confidence and confusion based on their performance in adaptive tests.
Iterative Adaptive Learning : Implement an iterative process where the system continuously assesses the learner's understanding and adjusts the difficulty level of questions accordingly, guiding learners from confusion to confidence.

3.3 FUNCTIONAL REQUIREMENTS

3.3.1 User Interfaces

There will be six user interface:

- User Interface (UI): Learner and educator interfaces, Intuitive and user-friendly design.
- Authentication Interface: Secure login and registration, Password recovery and account management.
- Adaptive Testing Interface: Dynamic presentation of questions, Immediate feedback on assessments.
- Question Generation Interface: Bloom's taxonomy-aligned question creation, Input for educators to specify difficulty levels.
- Behavioral Parameter Analysis Interface: Dashboard for analyzing behavioral parameters, Insights into option changes and question-solving times.
- Affective State Identification Interface: Display of learner's affective state, Indication of confidence or confusion.
- Iterative Adaptive Learning Interface: Visualization of learner progression, Illustration of increased confidence over assessments.
- User Management Interface: Administrative management of user accounts, User role and permission control.
- Content Management Interface: Educator interface for content management, Addition, modification, and removal of learning materials.
- Reporting and Analytics Interface: Comprehensive reporting and analytics, Visualization of learner progress and trends.
- Integration Interface with Learning Management System (LMS): Seamless data exchange with an LMS, Synchronization of user information and course content.
- System Administration Interface: Interface for system administrators, Monitoring system health and managing configurations.

3.3.2 Hardware Interfaces

- PIV 2.8 GHz Processor and Above
- RAM 512MB and Above
- HDD 20 GB Hard Disk Space and Above

3.3.3 Software Interfaces

XAMPP is an easy to install Apache distribution containing MySQL, PHP and Perl. XAMPP is really very easy to install and to use - just download, extract and start.

- **WINDOWS OS (XP / 2000 / 200 Server / 2003 Server)**
- **Integrated Development Environment (IDE): Visual Studio Code or PyCharm**
- **Version Control: Git with GitHub or GitLab**
- **Database Management: Django ORM with DBeaver or pgAdmin**
- **Web Browser: Google Chrome or Mozilla Firefox**
- **Command Line Interface (CLI): Terminal or Command Prompt**
- **Virtual Environment: Python's virtualenv or venv**
- **Package Manager: Pip with requirements.txt**
- **Web Server: Django's built-in server for local development**
- **Frontend Framework: Bootstrap or Tailwind CSS**
- **JavaScript Framework: Vue.js, React, or Angular or Flask**

3.3.4 Communication Interfaces

Designing a communication interface for a water bottle management system involves creating a system that allows various components to exchange information efficiently. Here's a basic outline of how you might structure the communication interface:

1. **User-friendly Design:**
   - **Create an intuitive and easy-to-navigate interface for a positive user experience.**
   - **Ensure responsiveness across devices for accessibility.**
2. **Real-time Updates:**
   - **Implement features for instant updates to keep users informed.**
   - **Use real-time technologies like WebSocket for seamless communication.**
3. **Security Measures:**
   - **Prioritize user data security with robust authentication and encryption.**
   - **Regularly update and patch vulnerabilities to maintain a secure environment.**

4. **Analytics and User Feedback:**

   - **Integrate analytics tools for insights into user behavior and application performance.**
   - **Include a user feedback mechanism to gather opinions for continuous improvement.**

3.4 NONFUNCTIONAL REQUIREMENTS

Nonfunctional requirements are crucial aspects of a system that describe how it should behave and perform rather than what specific functions it should fulfill.

3.4.1 Performance Requirements:

Performance requirements for the project include ensuring that the system responds to user interactions for optimal user experience. The system should be designed to handle a user base increase without significant performance degradation. Additionally, the platform should support a minimum number of concurrent users accessing the system simultaneously without performance bottlenecks. Data loading times for standard-sized datasets should be optimized to facilitate quick access to information.
To ensure continuous availability, the project should maintain a high server uptime. The system should be designed to handle peak traffic efficiently, accommodating simultaneous requests during high-demand periods. Database queries should be optimized for efficient data retrieval. File uploads and downloads should be supported with speeds to facilitate quick exchange of multimedia content.
Furthermore, the project should aim for a low error rate, with a minimal percentage of user interactions resulting in errors or system failures. Load balancing mechanisms should be implemented to distribute workloads evenly and prevent performance degradation during peak usage. These performance requirements provide essential guidelines for the optimal functioning of the project.
Here are some key performance requirements:

- **Response Time**
- **Scalability**
- **Concurrent Users**
- **Data Load Time**
- **Server Uptime**
- **Peak Traffic Handling**
- **Database Query Performance**
- **File Upload/Download Speed**

- **Error Rate**
- **System Load Balancing**

3.4.2 Security Requirements:

Security is a critical aspect of any system, and the water bottle management system should incorporate robust measures to safeguard sensitive data and ensure the integrity and confidentiality of its operations. Here are key security requirements for system:

- **Authentication and Authorization : Require secure authentication mechanisms such as strong passwords, biometrics, or multi-factor authentication to verify the identity of users.**
- **Data Encryption : Use encryption protocols (e.g., TLS/SSL) to secure data transmitted between components, such as between water dispensers and the central server.**
- **Secure Communication Protocols : Implement secure APIs with proper authentication and authorization mechanisms for communication between system components.**
- **Secure Storage : Safeguard dispensing logs and user data by securely storing them in the database with proper access controls.**
- **Secure User Interface : Implement secure session management to protect against session hijacking and ensure the confidentiality of user interactions.**

3.4.3 Safety Requirements:

Safety is a critical consideration for any system, especially one that involves the management of water bottles.

- **Questioning Quality Monitoring**
- **Database Management**
- **Question Generation Processing**
- **Proper Evaluation**

3.4.4 Software Quality Attributes:

Software quality attributes, also known as nonfunctional requirements, are essential considerations for the design and implementation of any software system, including a water bottle distributor management system. Here are key software quality attributes for such a system:

- **Reliability**
- **Availability**

- **Performance**
- **Scalability**
- **Security**

3.5 SYSTEM REQUIREMENTS

3.5.1 Data Store Requirements:

- **MySQL**
- **Excel Sheets**

3.5.2 Software Requirements:

These are the software requirements for running this project :

- **WINDOWS OS (XP / 2000 / 200 Server / 2003 Server)**
- **Integrated Development Environment (IDE): Visual Studio Code or PyCharm**
- **Version Control: Git with GitHub or GitLab**
- **Web Browser: Google Chrome or Mozilla Firefox**
- **Web Server: Django's built-in server for local development**
- **Web Development Frameworks: Vue.js, React, or Angular or Flask**

3.5.3 Hardware Requirements:

- **RAM 512MB and Above**
- **HDD 20 GB Hard Disk Space and Above**

3.6 ANALYSIS MODELS

The analysis model presented evaluates a learner's affective state by considering correctness of answers (P1), number of option changes (P2), and time taken per question (P3) in a multiple-choice question (MCQ) test. Utilizing a linear regression equation, the model computes a fuzzy value, 'A,' ranging from 0 to 1, with 1 indicating confidence and 0 indicating confusion. The coefficients (a, b, c) are derived through training on MCQ tests, and the model is applied to subsequent tests to generate 'A' values for each question. Cumulatively, these values provide a topic-wise measure of the learner's affective state, aiding educators in understanding and addressing areas of confusion or confidence based on the learner's performance.

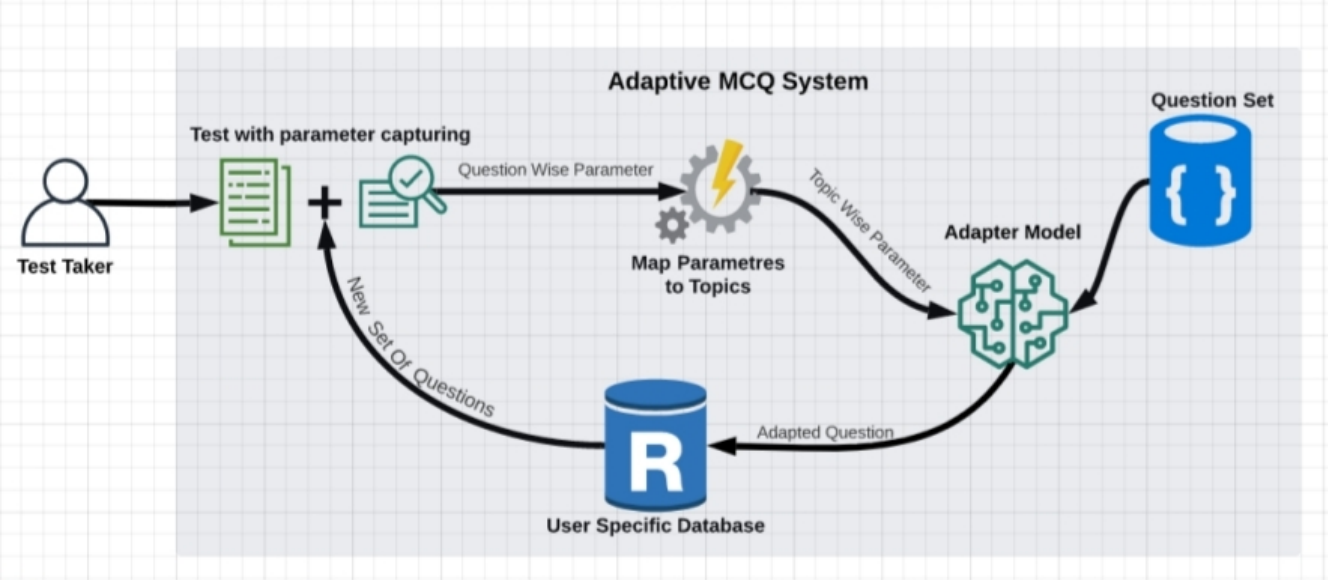


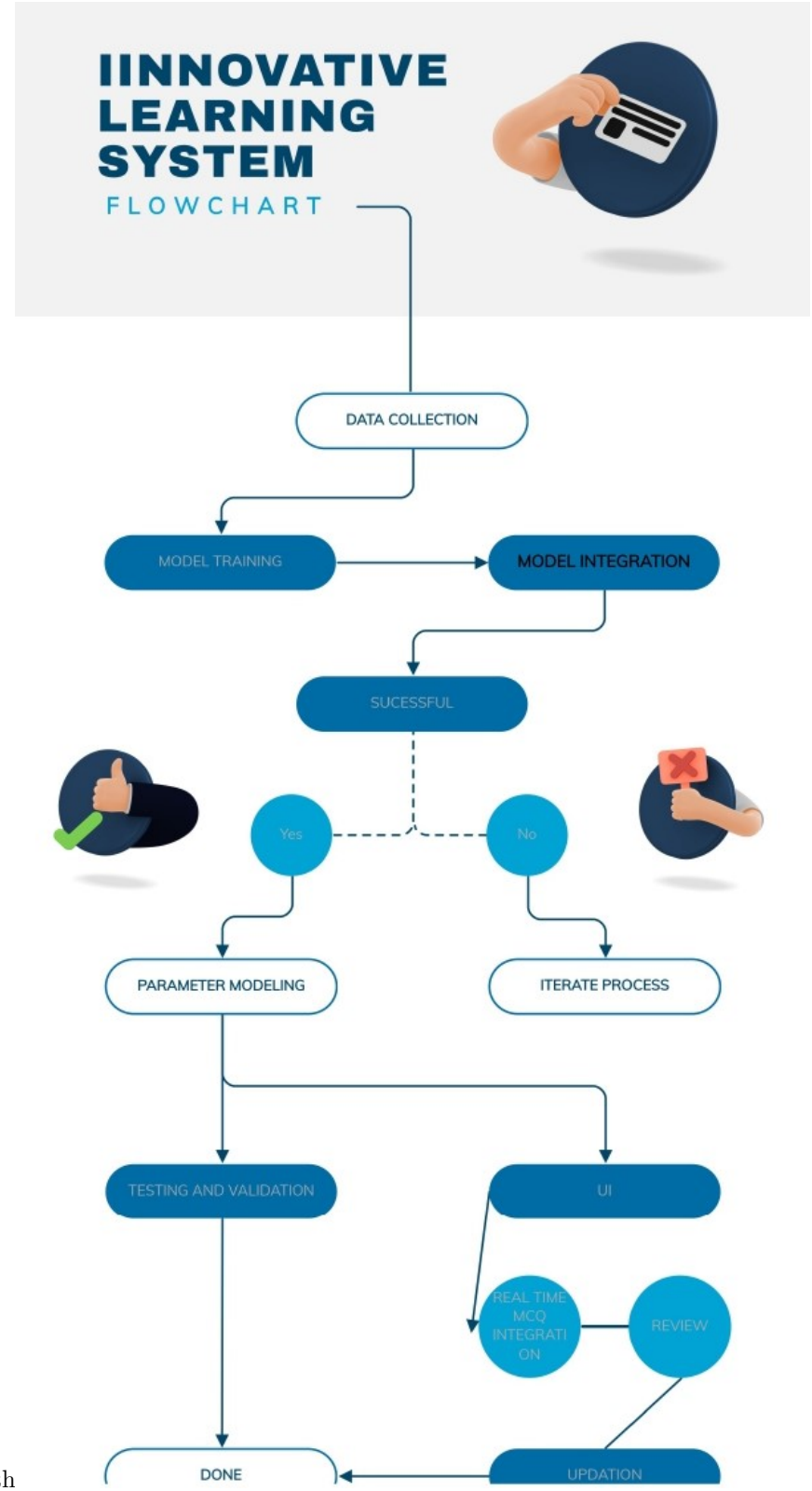
Figure 3.1: Model Diagram

3.7 SYSTEM IMPLEMENTATION PLAN

Creating a system implementation plan for a water bottle distributor management system using web technology involves a systematic approach to ensure a successful and smooth deployment. Here's a structured plan to guide you through the implementation process:

1. **Data Collection:**
   - **Gather historical data on learner performance in MCQ tests, including correctness of answers, number of option changes, and time taken per question.**
2. **Model Training:**
   - **Derive coefficients (a, b, c) for the linear regression equation using a training dataset.**
   - **Establish the class values for P1, P2, and P3 based on the provided criteria.**
3. **Model Integration:**
   - **Implement the linear regression model in the system to compute 'A' values for each question during subsequent MCQ tests.**
4. **Parameter Monitoring:**
   - **Continuously monitor and update the model as new learner performance data becomes available, ensuring the model remains reflective of learner behavior.**
5. **Testing and Validation:**
   - **Validate the model by applying it to new MCQ tests and comparing predicted 'A' values with actual learner outcomes.**

6. **User Interface Design:**
    - **Develop a user-friendly interface for educators to input MCQ test results and receive cumulative 'A' values for each topic.**
7. **Integration with Educational Systems:**
    - **Integrate the affective state analysis system with existing educational platforms for seamless implementation and data exchange.**
8. **Feedback Mechanism:**
    - **Establish a feedback mechanism for educators to provide insights on the accuracy of the affective state predictions and make necessary adjustments to the model.**
9. **Continuous Improvement:**
    - **Regularly assess the system's performance, gather user feedback, and make enhancements to improve the accuracy and relevance of affective state predictions.**



widthwidth

Figure 3.2: Project Flowchart

Chapter 4

SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM

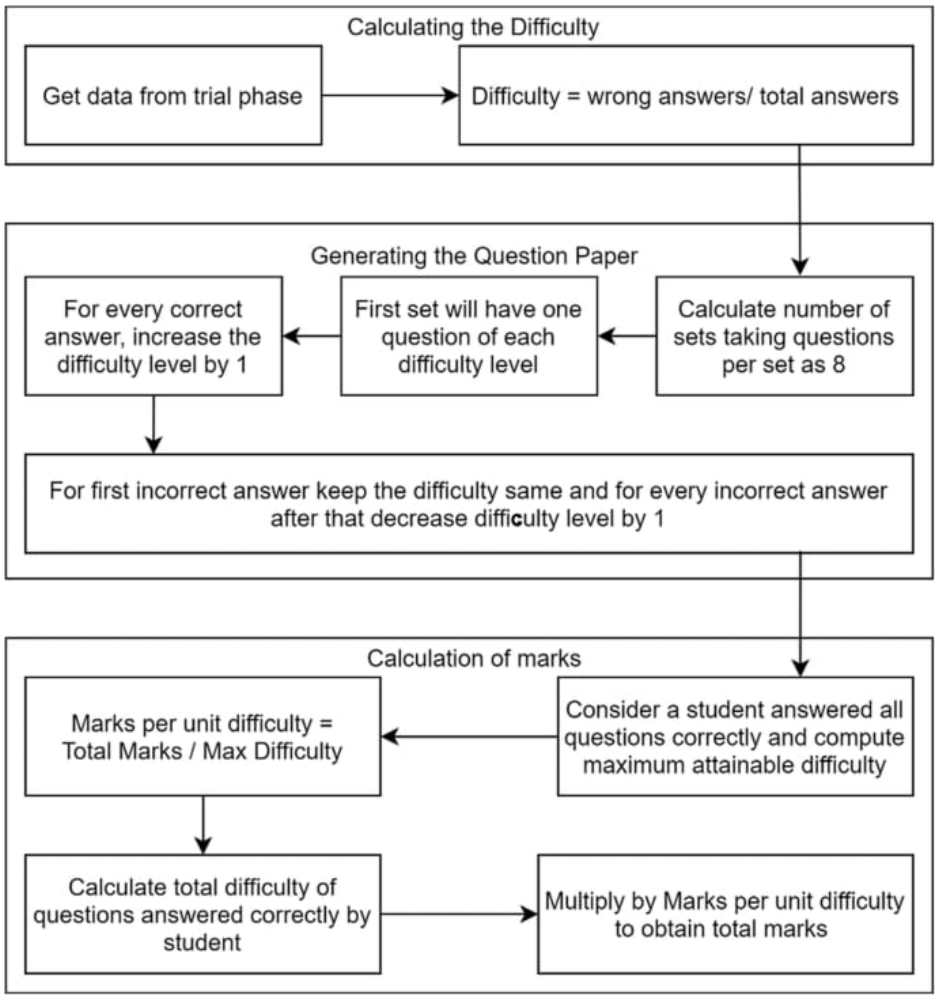


Figure 4.1: Data Flow Diagram

4.2 ENTITY RELATIONSHIP DIAGRAM

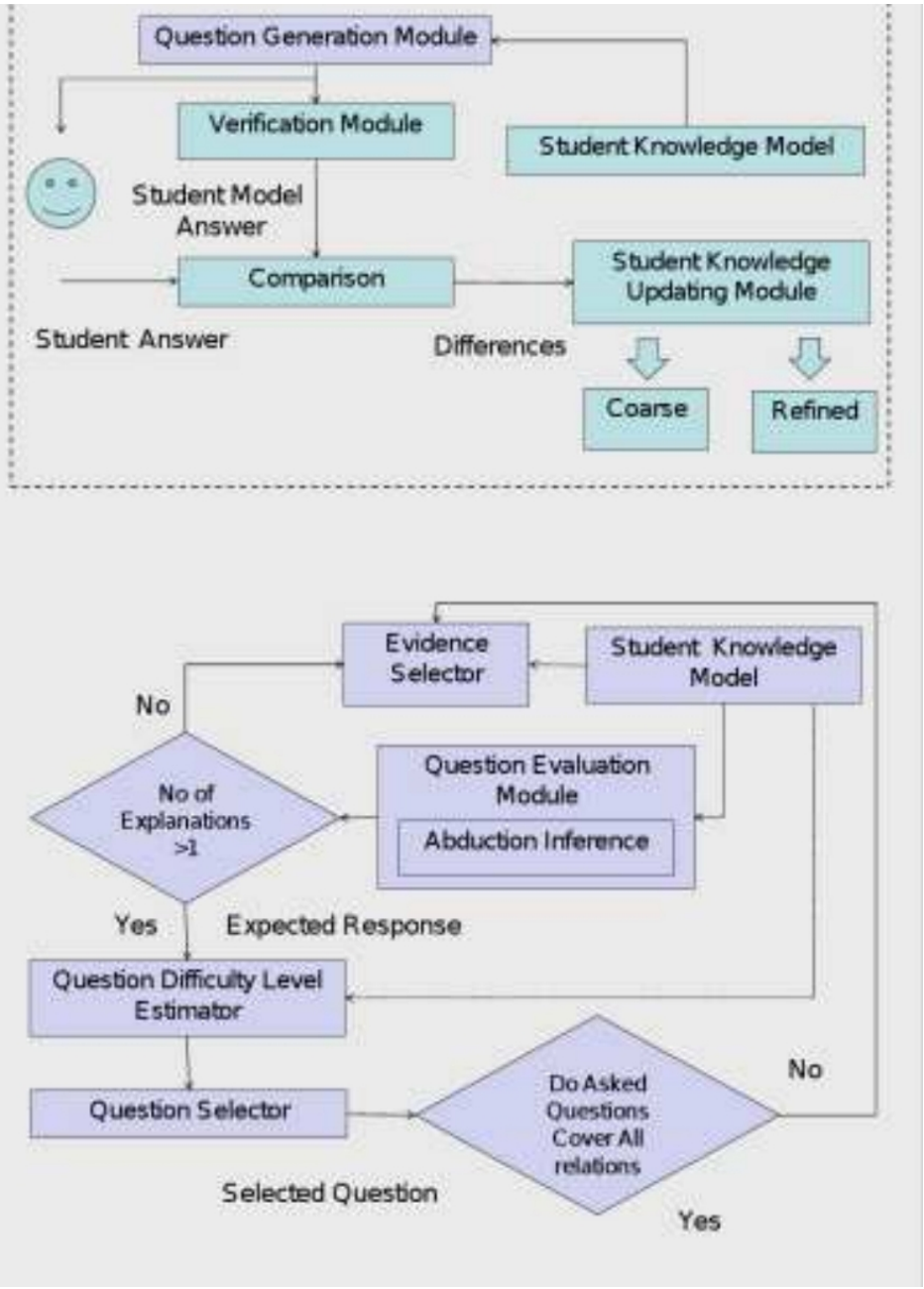


Figure 4.2: Entity Relationship Diagram

4.3 UML DIAGRAM

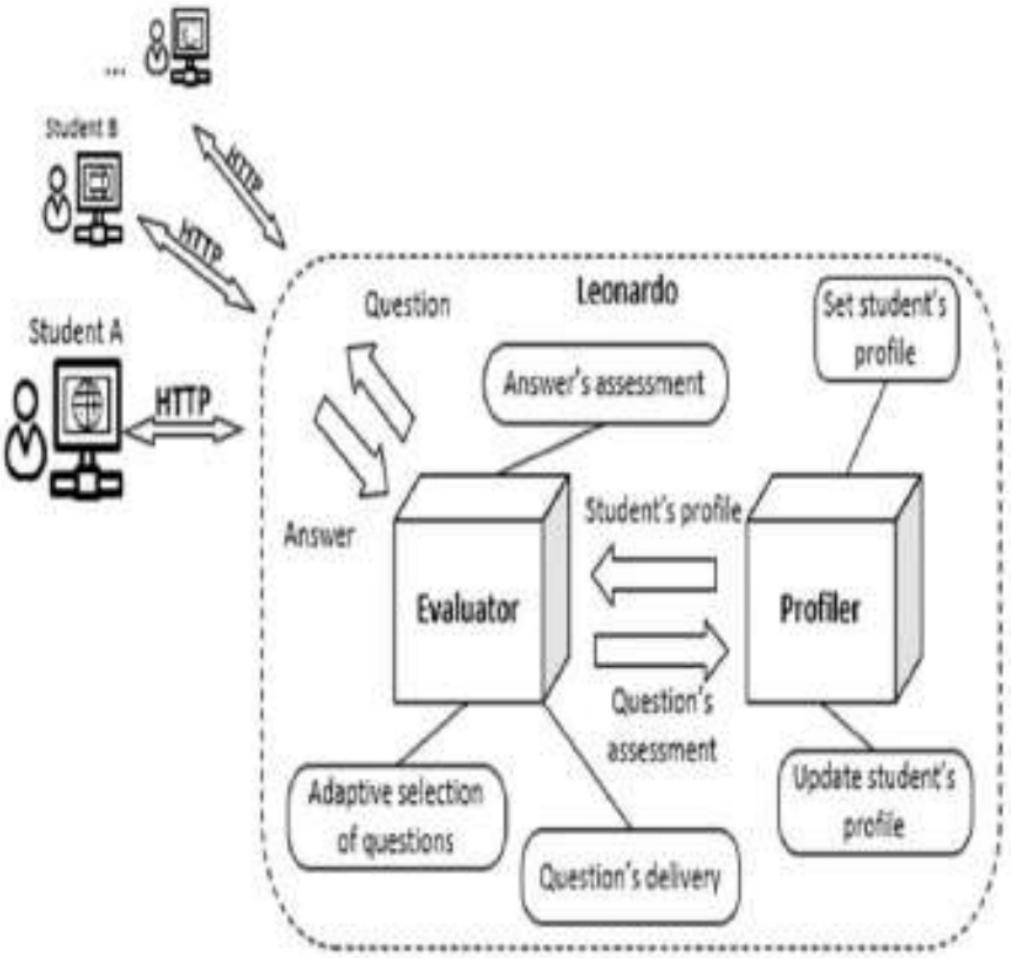


Figure 4.3: Usecase Diagram

Chapter 5

TECHNICAL SPECIFICATION

5.1 Technology details used in project :

The technical specifications of the "Tiffin Service Provider Website using Multivendor System" outline the technology stack and details used in the development and implementation of the project. These specifications encompass the programming languages, frameworks, databases, and other tools utilized to build a robust and efficient system.

1. **Front-end Development :**
   - **HTML5: Used for creating the structure and content of web pages. HTML5 introduces new elements and attributes that provide more flexibility and functionality, such as `<video>`, `<audio>`, and `<canvas>` for multimedia, as well as semantic elements like `<article>`, `<section>`, and `<footer>` that improve the readability and accessibility of web content.**
   - **CSS3: Utilized for styling and formatting the appearance of web pages. CSS3 includes advanced features such as transitions, animations, and flexible box layouts (Flexbox), allowing developers to create visually appealing and responsive designs that adapt to different screen sizes and devices.**
   - **JavaScript: Essential for implementing interactive features and functionalities on the client-side. JavaScript enables dynamic content updates, form validation, and user interactions without needing to reload the page. Libraries and frameworks such as jQuery, React, Angular, and Vue.js further enhance JavaScript's capabilities, making it possible to build complex and high-performance web applications.**
2. **Back-end Development**
   - **Python: A versatile programming language used for server-side scripting and business logic implementation. Python's simplicity and readability make it a popular choice for web development, data analysis, artificial intelligence, and more.**
   - **Django Framework: A high-level Python web framework that promotes rapid development, scalability, and security of web applications. Django follows the "batteries-included" philosophy, providing**

a comprehensive set of tools and features out of the box, such as an ORM (Object-Relational Mapping), authentication, and an admin interface, which helps developers build robust applications quickly and efficiently.

3. Database Management System (DBMS)

   - SQLite3: A popular relational database management system used for storing, retrieving, and managing structured data efficiently. MySQL supports SQL (Structured Query Language) for querying and managing data, making it a widely used choice for web applications, enterprise solutions, and data-driven services due to its reliability, performance, and ease of use.

4. Development Tools

   - Visual Studio Code: An integrated development environment (IDE) used for writing, editing, and debugging code. Visual Studio Code supports various programming languages and extensions, offering features such as syntax highlighting, code completion, version control integration, and an extensive ecosystem of plugins to enhance the development workflow and productivity.

5. Testing Framework

   - Selenium: A portable framework for automating web browsers' testing across different platforms and browsers. Selenium enables developers to write tests in multiple programming languages, such as Java, Python, and C-Sharp, ensuring the functionality and compatibility of web applications through automated browser interactions and verification steps.
   - Django Testing Framework: Built-in testing capabilities in Django for writing and executing unit tests, integration tests, and functional tests. The Django testing framework helps validate the behavior of Django applications by providing tools and utilities to create test cases, simulate requests, and check responses, ensuring the correctness and reliability of the application code.

Chapter 6

Model Implementation and Understanding

6.1 Affective state confusion and confidence

- **Confusion and confidence are learning attributes displayed by a learner. We define confusion and confidence of the learner. Confusion is defined as lack of understanding and a state of uncertainty where learner is unable to decide how to act or what to do next.**
- **Confidence is learner attribute observed through six factors studying, understanding, verbalizing, clarifying, attendance and grades.**

6.2 Performance parameters for deriving affective state

Performance in the test is only one of the major indications of such affective state. Our system captures following behavioral parameters.

- **Correctness of the question (P1)**
- **No. of times a user changes the options (P2)**
- **Time taken per question (P3)**

Using these parameters, we compute 'A' value, a fuzzy value, and affective state value. 'A' value lies between 0 to 1 which represents the affective state of confusion and confidence of a learner in a topic. 1 means the user knows the topic very well indicating affective state of confidence and, 0 means the user does not know the topic indicating affective state of confusion.

6.3 Linear Regression

In order to derive 'A' value from P1, P2, P3 we establish linear regression equation as :
A = a*P1 + b*P2 + c*P3

The explanation of deriving linear regression coefficients is as follows

- **P1 (Correctness) can have two values, 0 for wrong answer or 1 for correct answer.**
- **P2 (Option changes) can have any integral value, where 0 is the best case where as anything greater than 2 is treated as bad. When learner is confident he will not keep on changing the answer. A confused learner unsure of answer will incline more to keep changing the options.**
- **P3 (time per question) is divided into ranges, in accordance to Cognitive level of that question**

6.4 Resulting visuals

We then pass the above mentioned P1, P2, P3 class values, to our linear regression model, which returns a 'A' value for each question which will be between 0 to 1. The values of a,b and c (regression coefficients) are found in the training phase, each MCQ test attempted by the user becomes the test phase for the model. After getting a 'A' value for each question, we take average of all 'A' value for questions of a particular topic, generating a cumulative fuzzy value 'A' topic wise. Thus 'A' value is generated which is the Affective state of the learner on basis of performance in the first MCQ test. 'A' value near to 0 indicates confusion and 'A' value nearing 1 indicate Confidence.

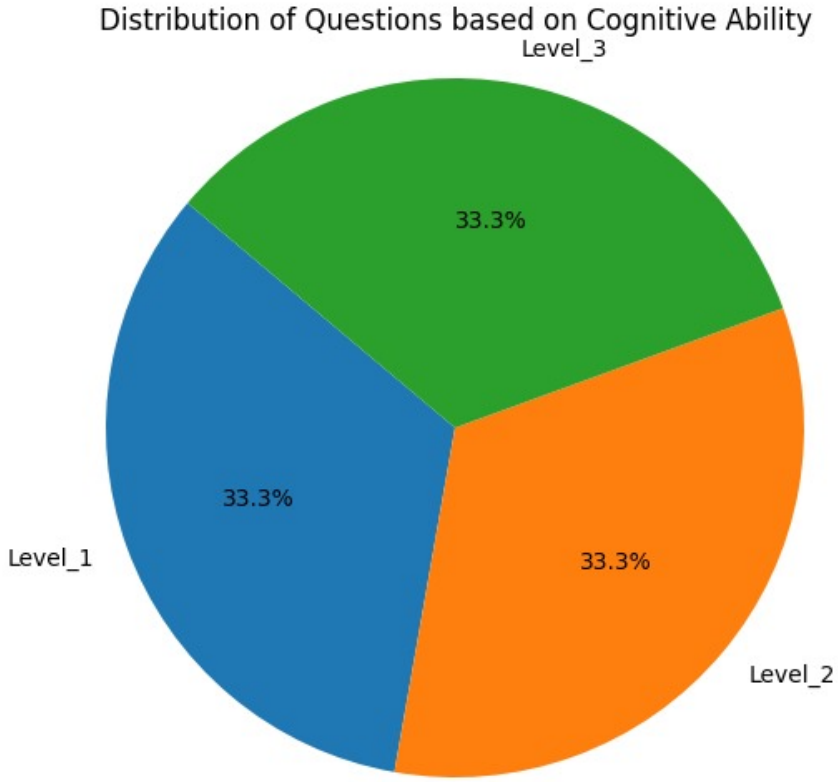


Figure 6.1: Distibution Piechart of questions based on Cognitive Levels

Chapter 7

Software Testing

7.1 Introduction

Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements. Testing is part of a lifecycle. The software development lifecycle is one in which we hear of a need, we write some code to fulfill it, and then we check to see whether we have pleased the stakeholders: the users, owners, and other people who have an interest in what the software does. Testing is a vital part of software development, and it is important to start it as early as possible, and to make testing a part of the process of deciding requirements. Testing is part of a lifecycle. The software development lifecycle is one in which we hear of a need, we write some code to fulfill it, and then we check to see whether we have pleased the stakeholders, the users, owners, and other people who have an interest in what the software does.

7.2 Software Testing Tools

Selenium is an open-source automated testing tool primarily used for web applications, offering a versatile framework for browser automation across multiple operating systems and browsers. Its core component, WebDriver, facilitates interactions with web browsers, enabling tasks like clicking buttons, entering text, and navigating pages through APIs. Selenium IDE provides a browser plugin for recording and playback of user interactions, while Selenium Grid allows for parallel test execution across various browsers and platforms, optimizing resource utilization and reducing test execution time. With support for multiple programming languages like Java, Python, and JavaScript, Selenium enables cross-browser testing to ensure consistent functionality and compatibility of web applications across different environments.

7.3 Software Testing Techniques

- Functional Testing: Functional testing for Android applications focuses on verifying the functional requirements and features of the application. It ensures that the app behaves correctly and performs the intended

tasks as expected. Here are some key aspects and techniques involved in functional testing for Android applications:

  - Test Case Design: Functional testing begins with the creation of test cases that cover various functionalities and scenarios of the application. Test cases should be designed to validate input validations, user interactions, navigation flows, and expected outputs.
  - Input Validation: Testers should validate user inputs, such as text fields, forms, or selections, to ensure that the application handles different input types correctly. This includes testing for data validation, error handling, and proper display of error messages.

- Unit Testing: During the unit testing phase, the system is tested while it is developed. Here all the options of the system are validated. During the first phase of this testing, the testing person tests the system by entering the valid data, or by performing the appropriate function which the system requests. This phase of testing is done to verify whether the system performs all the requested functions.
- Integration Testing: The integration will be based on the total whole module testing the integration of the module will be categorized as follows:
  - Checking the module and data entry status.
  - Redirectional functionalities.
  - Checking the status through ID.
  - Checking chart accuracy.
  - Error in data structures and external database error.

7.4 Types of Testing

In software development, testing plays a crucial role in ensuring the quality and reliability of the final product. There are various types of testing methodologies employed throughout the software development lifecycle, each serving different purposes.

7.4.1 Manual Testing

This involves the execution of test cases by human testers without the use of automated tools. While automation testing offers numerous advantages, there are scenarios where manual testing is preferred, especially in the case of website testing, for the following reasons:

1. User Experience Testing: Manual testing allows testers to evaluate the website's user experience from a subjective perspective, identifying issues that may not be easily detectable through automated scripts. Testers can provide valuable feedback on usability, accessibility, and overall user satisfaction.

2. **Exploratory Testing:** Manual testing is well-suited for exploratory testing, where testers can creatively explore different paths and scenarios within the website to uncover unexpected defects or usability issues. This flexibility and adaptability are difficult to achieve through automated scripts, which typically follow predefined paths.

3. **Visual Testing:** Human testers can easily detect visual anomalies such as layout issues, alignment problems, or graphical glitches that may impact the website's appearance and aesthetics. While there are visual testing tools available, manual inspection remains a reliable method for ensuring visual consistency across various browsers and devices.

4. **Ad Hoc Testing:** Manual testing allows for spontaneous, ad hoc testing where testers can quickly verify specific features or functionalities without the need for pre-scripted test cases. This agility is particularly useful during exploratory phases or when testing new features that haven't been automated yet.

5. **Complex Scenarios:** Some scenarios, such as complex business logic, multi-step workflows, or edge cases, may be challenging to automate effectively. Manual testing provides the flexibility to handle such scenarios with human judgment and reasoning, adapting test cases as needed to cover intricate use cases.

6. **Early Testing:** In the early stages of development when the website's features are still evolving rapidly, manual testing can provide immediate feedback to developers without the overhead of maintaining automated test scripts. This early validation helps in identifying issues sooner and reducing rework later in the development cycle.

7. **Cost-Effectiveness for Non-Enterprise Projects:** For non-enterprise projects or projects with limited budgets, manual testing may be more cost-effective initially, as it doesn't require the upfront investment in automation infrastructure and tooling. Automated testing becomes more cost-effective over time as the project matures and the need for regression testing increases.

8. **Human Judgment and Creativity:** Manual testers can apply their domain knowledge, intuition, and creativity to uncover subtle defects or usability issues that automated tools may overlook. Human judgment is particularly valuable in assessing subjective aspects of the website, such as content relevance or tone.

While automation testing offers benefits such as repeatability, scalability, and faster execution, manual testing remains indispensable for certain aspects of website testing, particularly those related to user experience, exploratory testing, visual validation, and handling complex scenarios. We have emphasized the positives of manual testing in our project.

7.4.2 Creating Manual Test Cases

Manual test cases are formulated based on the software requirements and specifications. The process of creating manual test cases is as follows:

1. Understanding Requirements: Review and understand the software requirements and specifications thoroughly.
2. Test Scenario Identification: Identify different scenarios that need to be tested based on various functionalities and features of the software.
3. Test Case Design: Develop detailed test cases for each identified scenario. A test case typically consists of:
   - Test Case ID: Unique identifier for the test case.
   - Description: Brief description of what the test case is supposed to accomplish.
   - Preconditions: Any necessary conditions that must be met before executing the test case.
   - Test Steps / Input: Step-by-step instructions on how to execute the test case.
   - Expected Results: Desired outcome or behavior when the test steps are executed successfully.
   - Actual Results: Space to record the actual outcome observed during test execution.
   - Status: Indicates whether the test case passed, failed, or needs retesting.
4. Review and Validation: Review the test cases to ensure completeness, clarity, and accuracy. Validate the test cases against the requirements to confirm that they adequately cover all functionalities.
5. Execution: Execute the test cases manually by following the specified steps and recording the results.
6. Defect Reporting: If any discrepancies or issues are encountered during test execution, they are documented in a defect tracking system for further investigation and resolution.
7. Regression Testing: After fixes or changes are made to the software, relevant test cases are re-run to ensure that the modifications have not introduced any new issues.

Manual testing was used effectively to create and execute test cases to validate the functionality and quality of our website.

7.5 Test Cases and Test Results

1. Admin Role						
TestCaseID	Prerequisite	Description	Input	Actual Result	Expected Result	Pass/Fail
TC_001	None	Test user registration for admin, teacher, and student roles.	User details	Successful user registration	Successful user registration	Pass
TC_004	Admin login	Test admin's ability to create, update, and delete users.	User details	User created/updated/deleted	User created/updated/deleted	Pass

Figure 7.1: Admin Testcase and results

2. Teachers Role						
TestCaseIDName	Prerequisite	Description	Input	Actual Result	Expected Result	Pass/Fail
TC 003	Teacher login	Test creating a new quiz with title, description, and rules.	Quiz details (title, description, rules)	Quiz created successfully	Quiz created successfully	Pass
TC 004	Created quiz	Test adding questions with different difficulty levels to a quiz.	Questions (text, options, correct answer, difficulty)	Questions added successfully	Questions added successfully	Pass
TC 005	Created quiz	Test setting adaptive criteria for the quiz.	Criteria (time required, option changes, correctness)	Criteria set successfully	Criteria set successfully	Pass
TC 006	Teacher login	Test viewing student performance reports.	Teacher ID, quiz ID	Performance reports generated	Performance reports generated	Pass

Figure 7.2: Teacher Testcase and results

3. Students Role						
TC 007	Registered user	Test user login with valid credentials.	Email, password	Login successful	Login successful	Pass
TC 008	None	Test user login with invalid credentials.	Email, incorrect password	Login failed	Login failed	Pass
TC 009	Student login, created quiz	Test starting a quiz for a student.	Quiz ID	Quiz started with an easy question	Quiz started with an easy question	Pass
TC 010	Quiz started	Test response to a question (correct answer within time).	Answer, time taken	Next question (medium) served	Next question (medium) served	Pass
TC 011	Quiz started	Test response to a question (incorrect answer or time exceeded).	Answer, time taken	Next question (easy) served	Next question (easy) served	Pass
TC 012	Multiple responses	Test adaptive logic by answering questions in various patterns.	Sequence of answers	Correct next question difficulty	Correct next question difficulty	Pass
TC 013	Quiz completed	Test displaying performance feedback to the student.	Student ID, quiz ID	Performance feedback displayed	Performance feedback displayed	Pass

Figure 7.3: Student Testcase and results

Chapter 8

Deployment and Maintenance

8.1 Deployment and Maintenance

- **Prepare for Deployment:**
  - **Perform thorough testing and quality assurance to ensure the website is stable and bug-free.**
  - **Optimize the website's performance and ensure it meets the required specifications.**
  - **Finalize the user interface and ensure it adheres to website design guidelines.**
- **Software Installation:**
  - **Installing the application software and its dependencies on the production servers.**
- **Distribution:**
  - **Upload the website on server host by getting a paid hosting and domain from any well-known hosting provider by following the necessary guidelines and requirements.**
  - **Provide all necessary information, including a descriptive app description, screenshots, and promotional materials.**
  - **Set pricing and distribution options based on your business model and target audience.**

8.1.1 Installation and Un-installation

During the deployment phase of the "Innovative Learning System" project, installation and uninstallation procedures are pivotal. Installation involves the setup and configuration of the software system on the production environment, ensuring that it is ready for operational use. This process includes provisioning servers, installing the necessary software components, configuring settings, and conducting thorough testing to verify the functionality of the deployed system. On the other hand, uninstallation refers to the removal of the software system from the production environment, either temporarily for

maintenance or permanently. This process involves backing up data, removing software components, and restoring the environment to its original state. Both installation and uninstallation procedures require careful planning, execution, and documentation to ensure the smooth and efficient operation of the tiffin service provider website.

8.1.2 Maintenance

- Bug Fixing:
  - Bug fixing in the tiffin service provider website involves identifying and resolving issues such as incorrect menu display, login errors, or order processing glitches.
  - Prompt resolution of bugs ensures uninterrupted service for users and maintains the credibility of the platform.
- Performance Optimization:
  - Performance optimization focuses on enhancing the speed and efficiency of the website's functions, including search queries, menu loading times, and checkout processes.
  - Optimizing performance ensures a seamless user experience, reduces bounce rates, and improves customer satisfaction.
- Security Update:
  - Regular application of security updates and patches helps protect user data, payment information, and sensitive credentials from cyber threats.
  - Implementing robust security measures safeguards the website against potential vulnerabilities and ensures compliance with data protection regulations.
- Database Maintenance:
  - Database maintenance includes tasks such as regular backups, optimization of database queries, and archiving of old data to ensure data integrity and system performance.
  - Proper database management minimizes the risk of data loss, enhances system.

Chapter 9

SNAPSHOTS

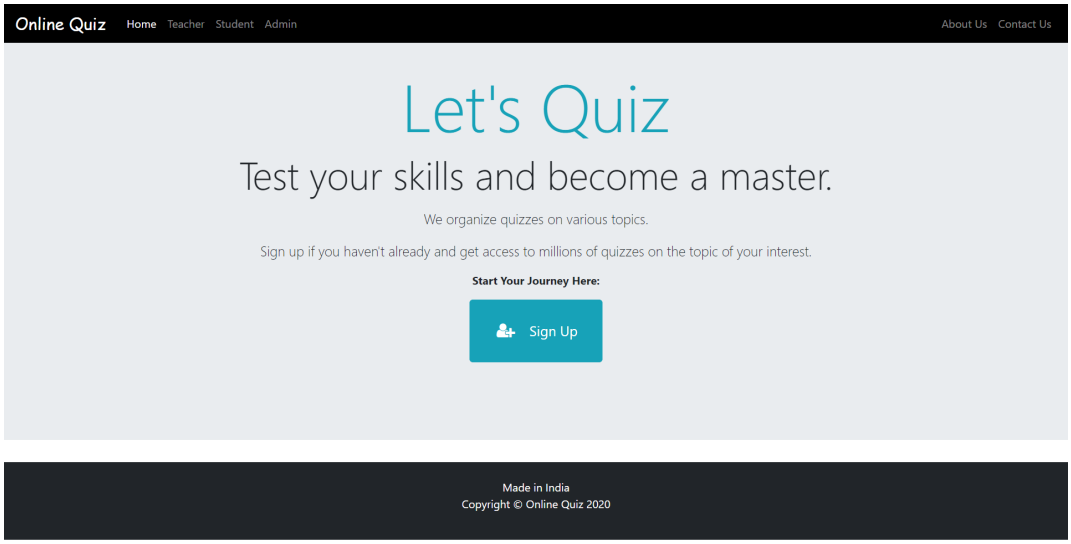


Figure 9.1: Main Page

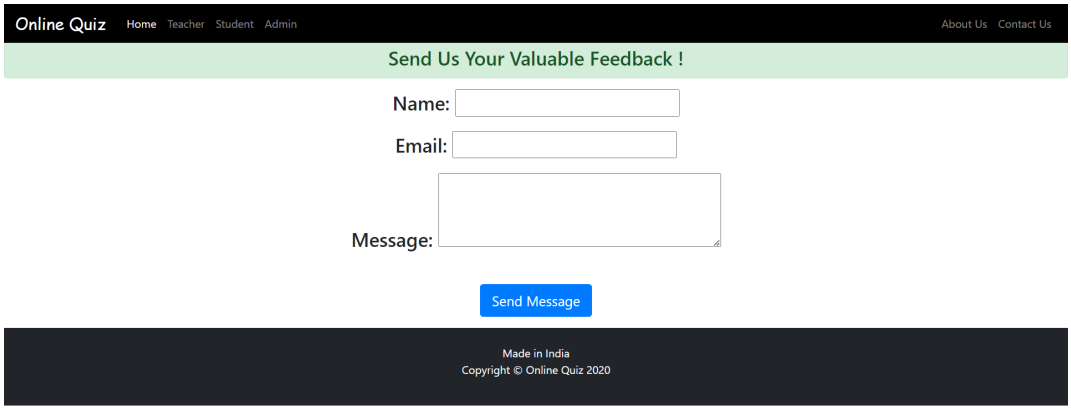


Figure 9.2: Contact Page

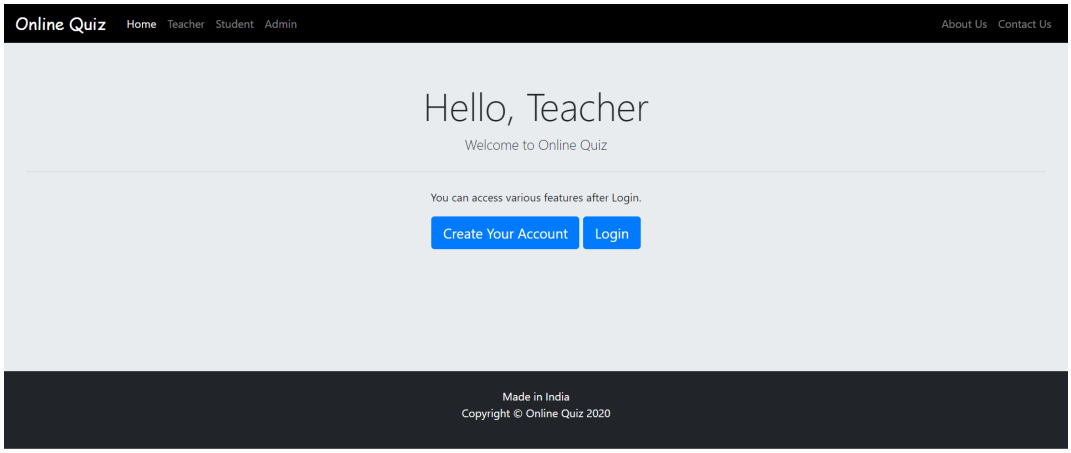


Figure 9.3: Teacher Signup

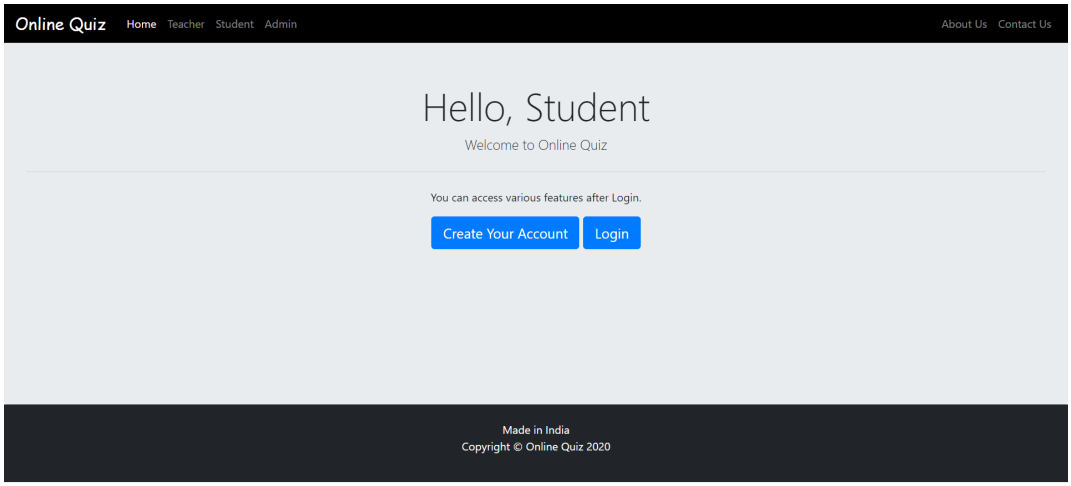


Figure 9.4: Student Signup

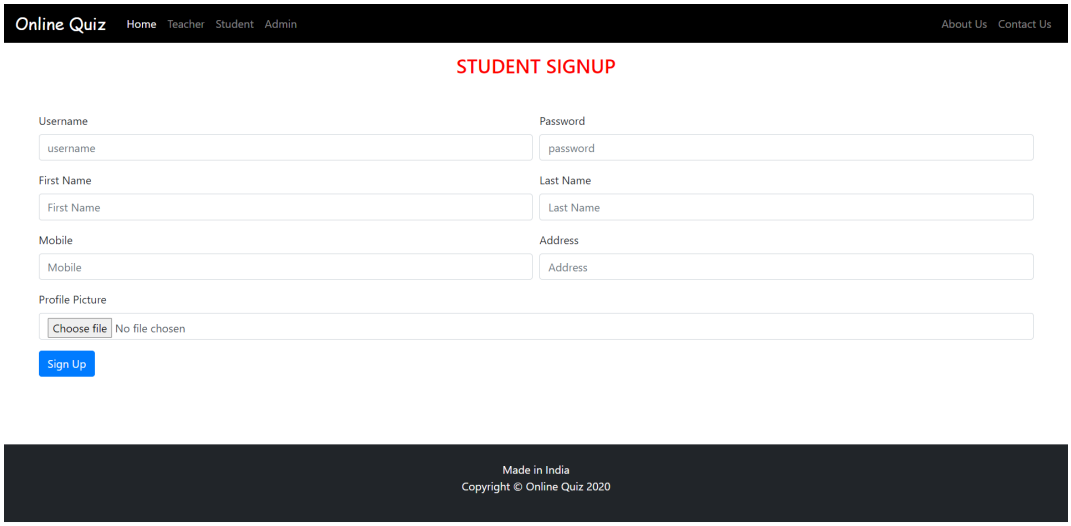


Figure 9.5: Student Signup Information

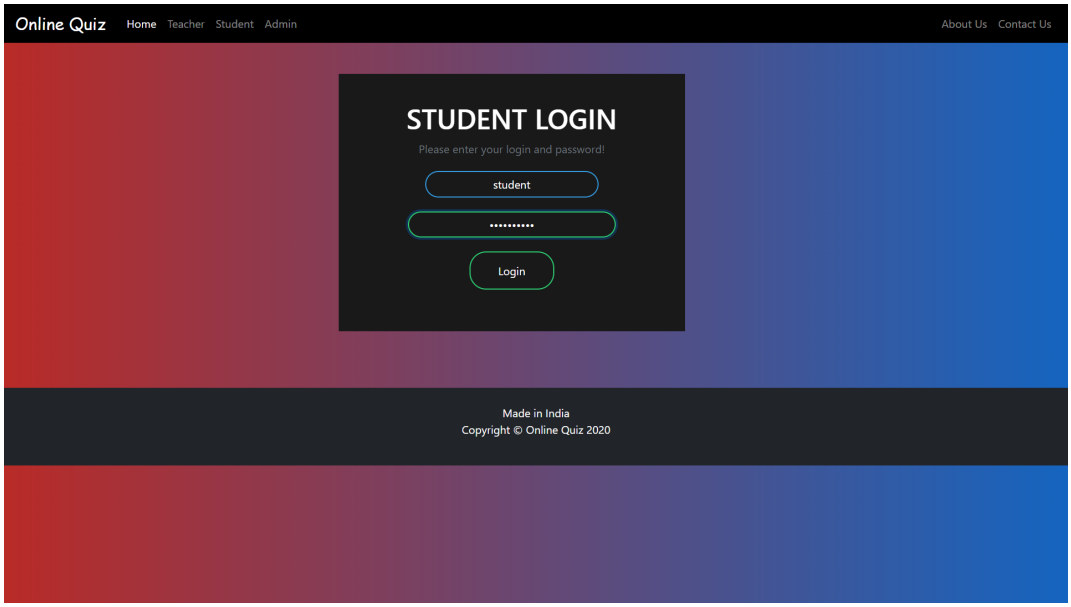


Figure 9.6: Student Login

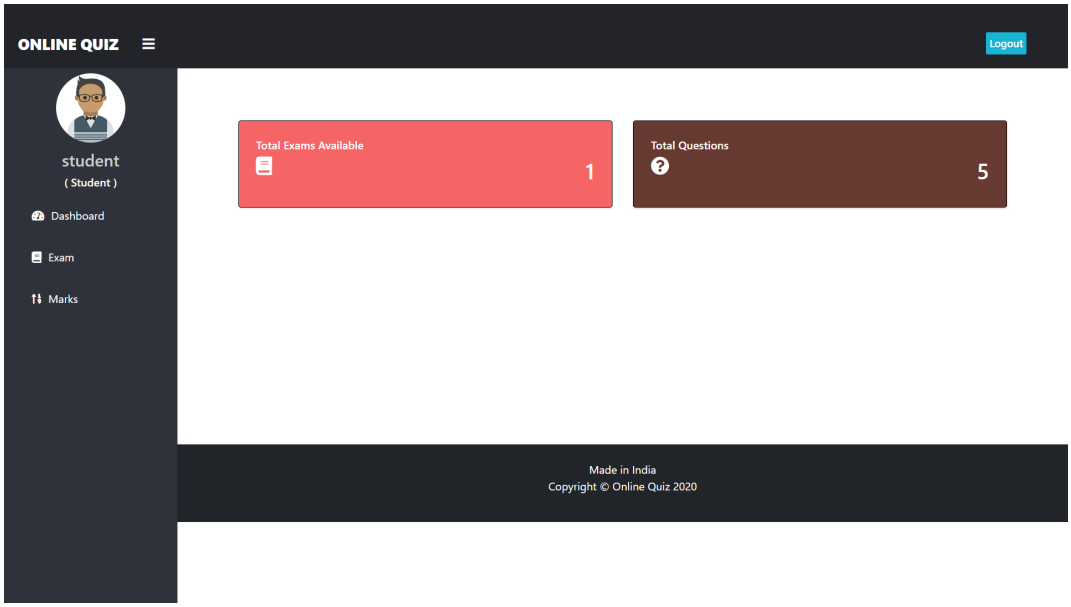


Figure 9.7: Dashboard

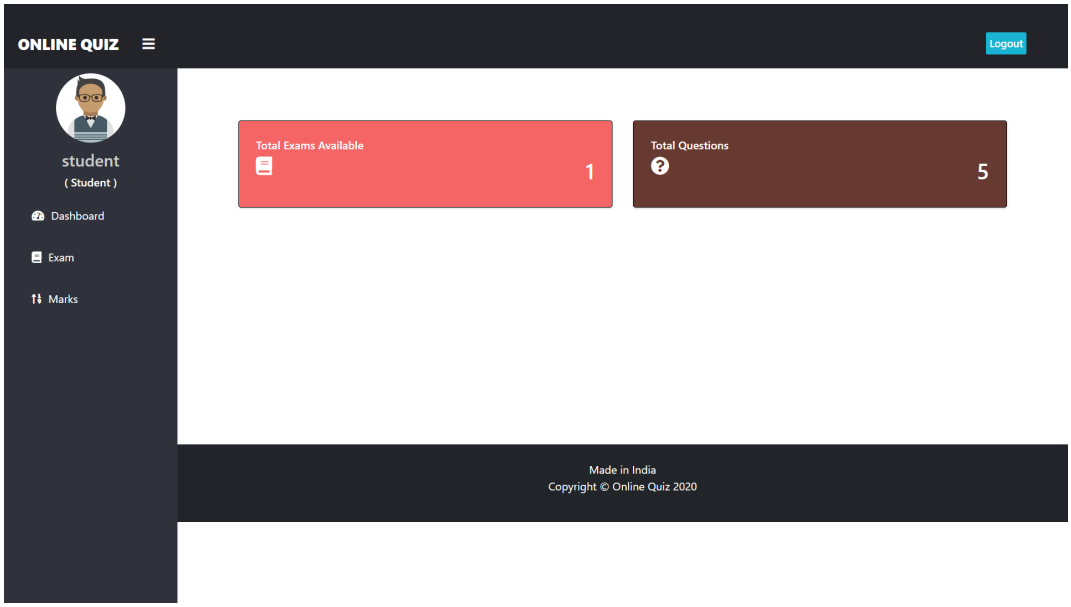


Figure 9.8: Dashboard

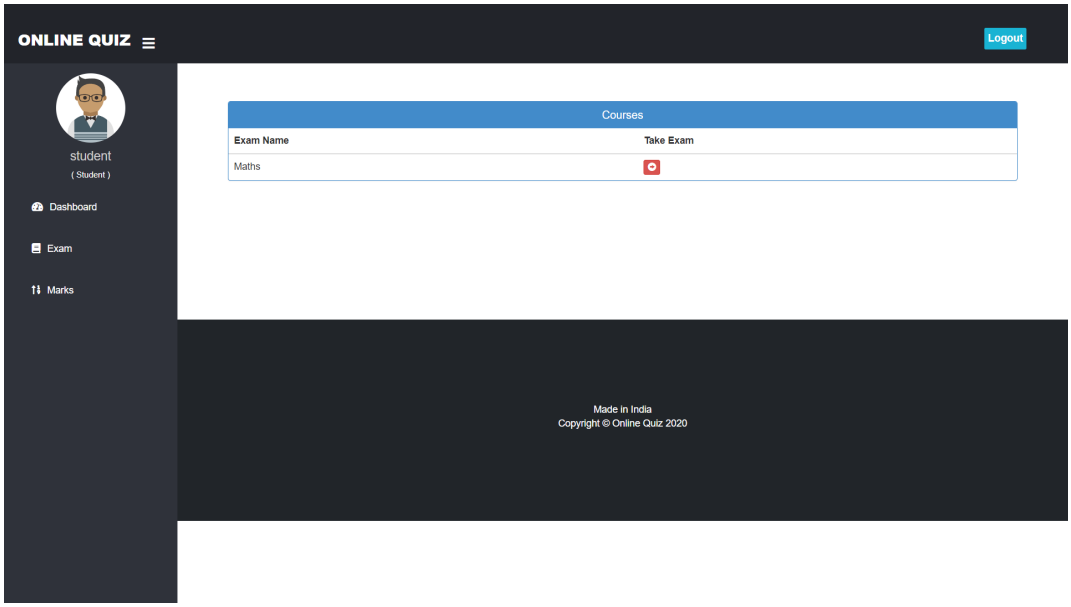


Figure 9.9: Exam Section

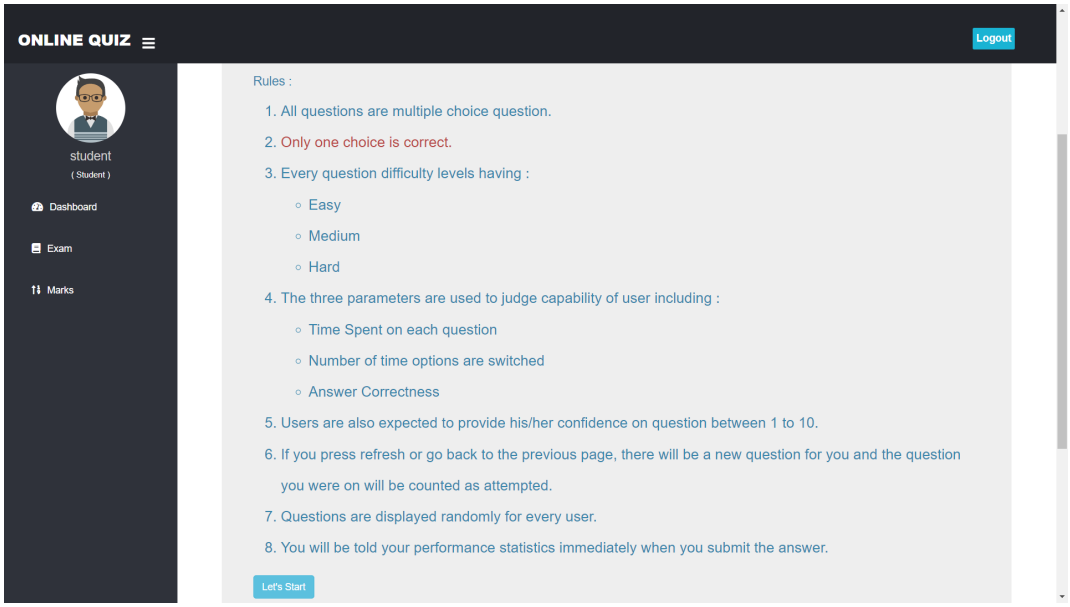


Figure 9.10: Exam Instructions

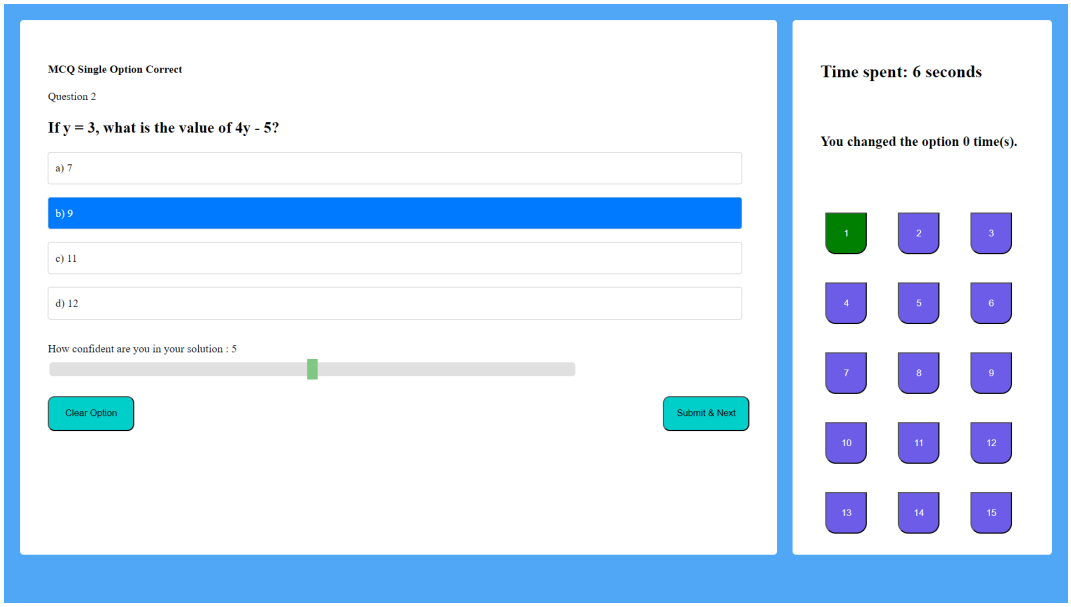


Figure 9.11: Questions View

Chapter 10

Resulting Visuals

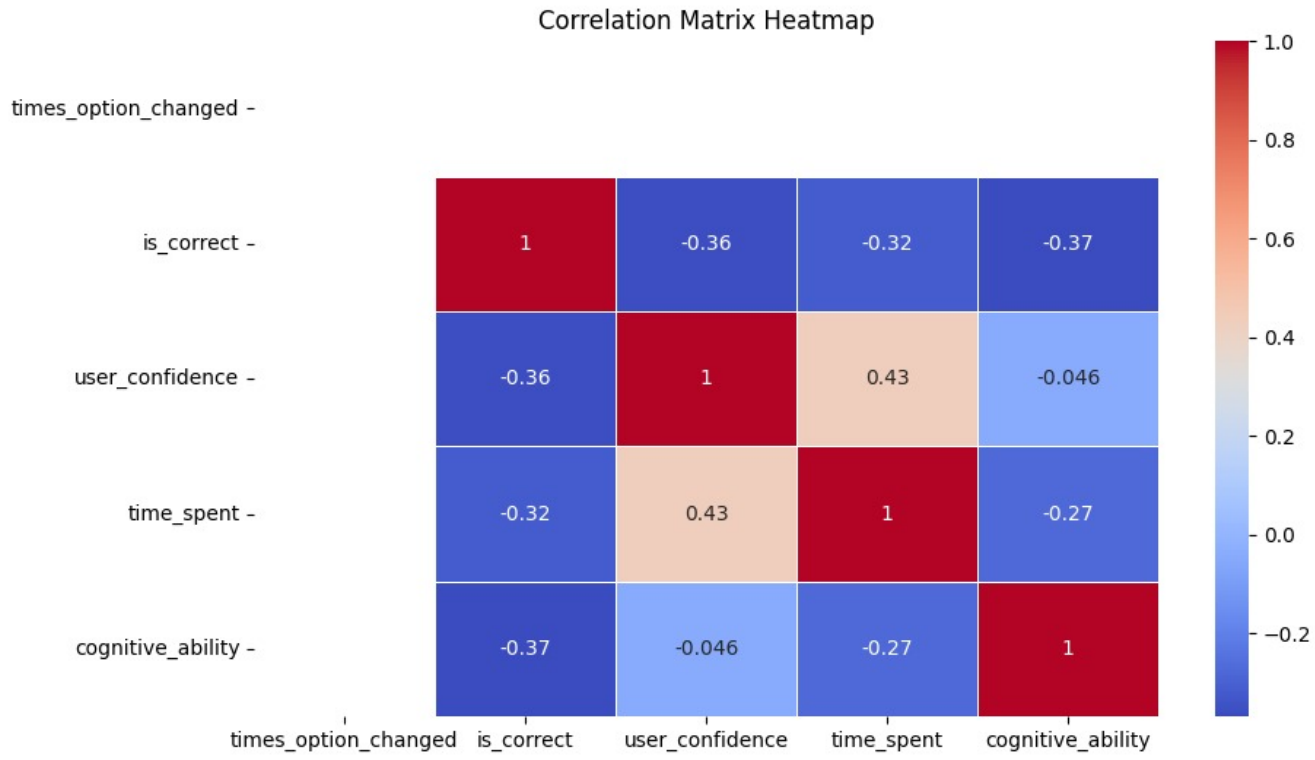


Figure 10.1: Correlation Matrix Heatmap

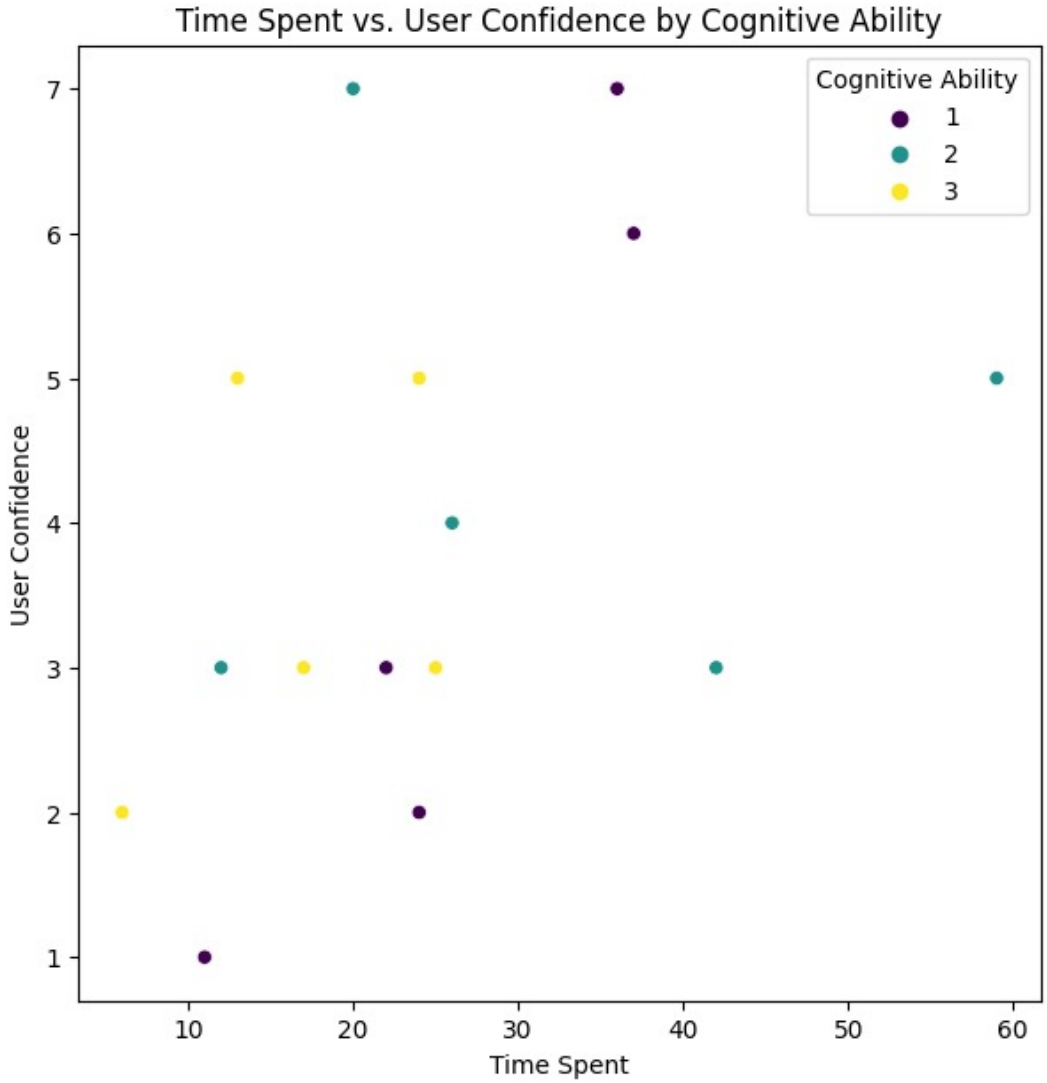


Figure 10.2: Time Spent vs User Confidence

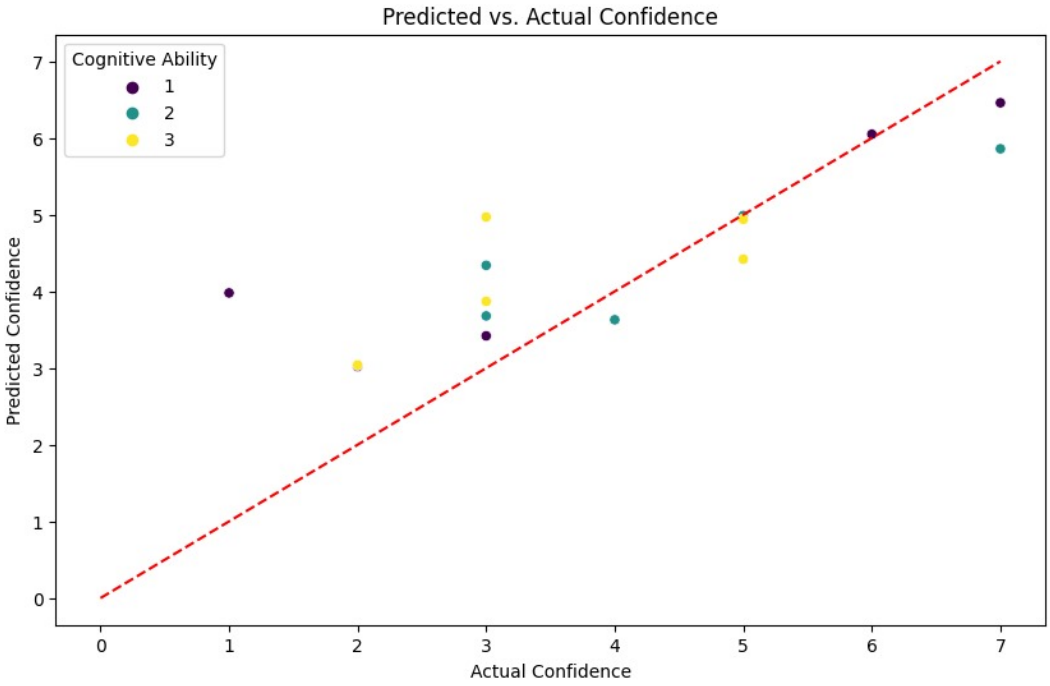


Figure 10.3: Predicted vs Actual Confidence

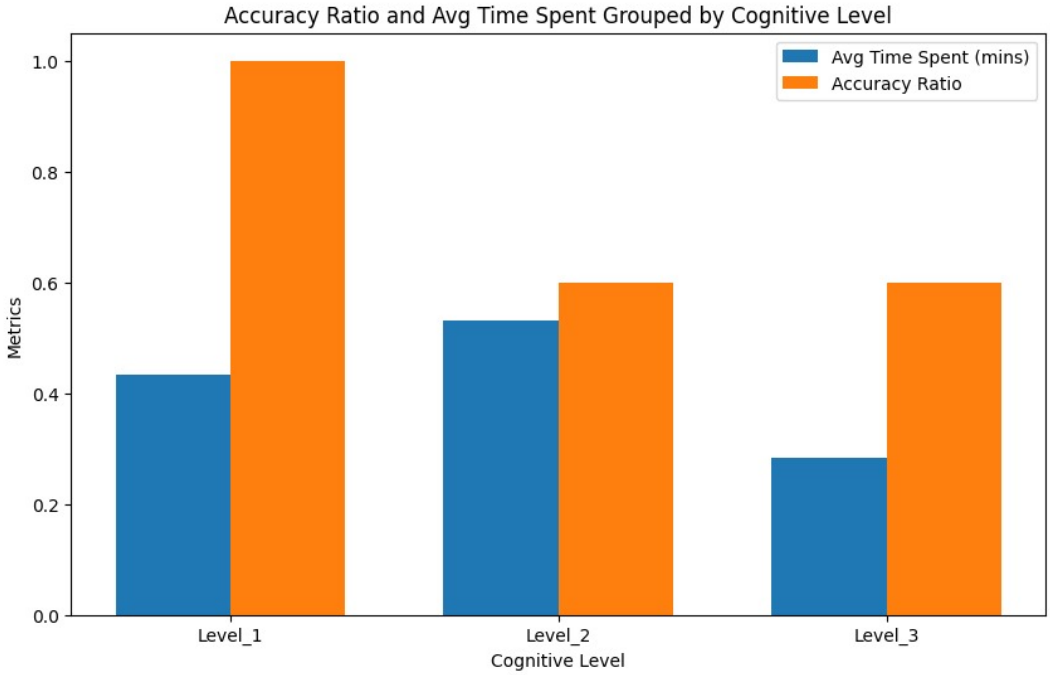


Figure 10.4: Accuracy Ratio vs Average Time Spent Grouped by Cognitive Level

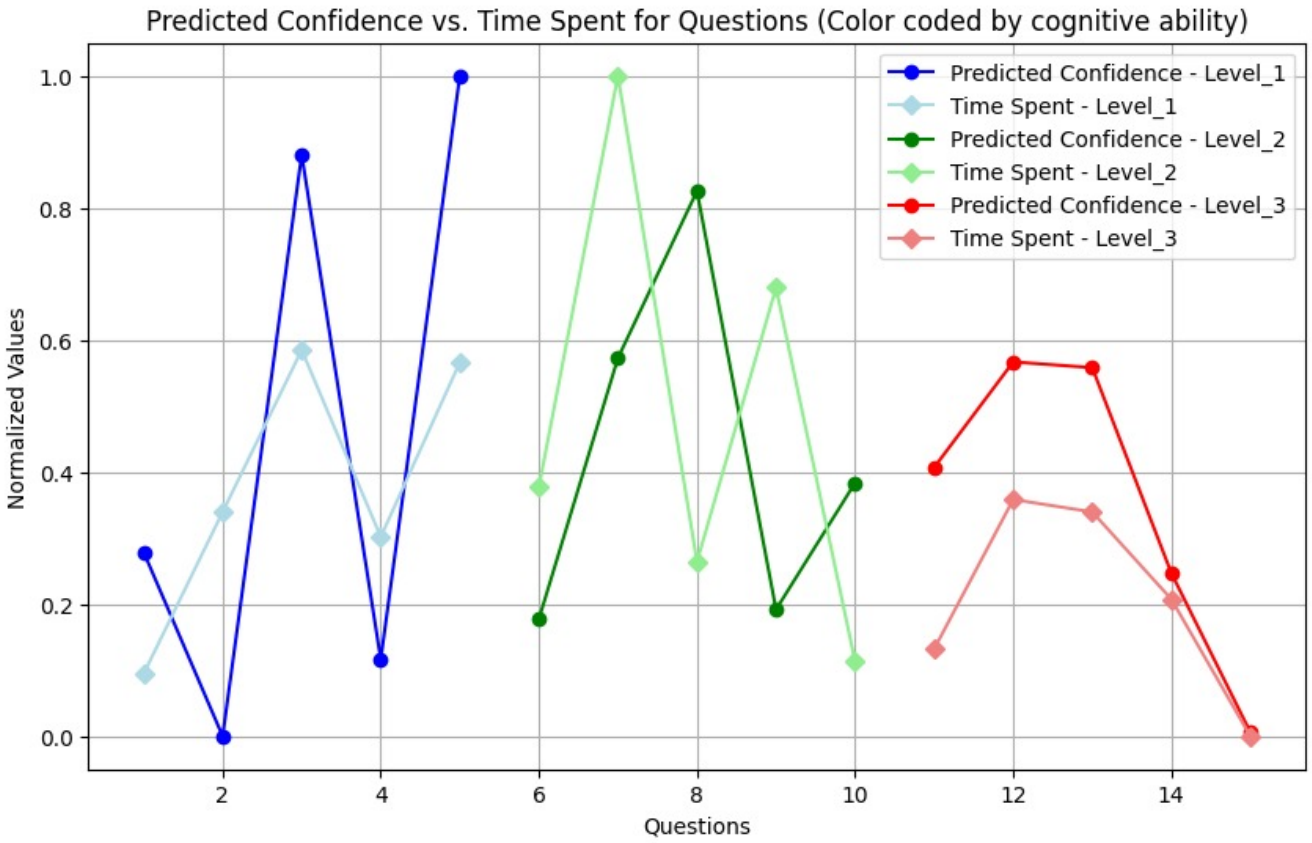


Figure 10.5: Predicted Confidence vs Time Spent for Questions (Color coded by Cognitive Level)

Chapter 11

OTHER SPECIFICATION

11.1 ADVANTAGES:

1. **Efficient Resource Allocation: By defining roles and responsibilities, the project ensures efficient allocation of resources, optimizing the contribution of each team member.**
2. **Robust System Architecture: The finalized system architecture, including the database structure and server setup, provides a solid foundation for scalability and reliability.**
3. **User-Friendly Interface: The development of wireframes and design mockups ensures the creation of a user-friendly interface, enhancing the overall user experience.**
4. **Structured Database Setup: The creation of a well-defined database structure, along with data migration procedures, ensures a structured and organized data environment.**
5. **Effective Testing and Validation: The project includes robust testing and validation processes, ensuring the accuracy and reliability of the implemented linear regression model.**
6. **Seamless Integration: Integration with existing educational platforms ensures a seamless implementation process and efficient data exchange.**
7. **Feedback-Driven Improvements: The establishment of a feedback mechanism allows for continuous improvements based on insights from educators, enhancing the accuracy of affective state predictions.**
8. **Continuous Performance Assessment: Regular assessments and user feedback contribute to continuous improvement, ensuring the system remains relevant and effective over time.**

11.2 LIMITATIONS:

1. Limited Stakeholder Involvement: The success of the project heavily relies on stakeholder involvement. Limited engagement may hinder the effectiveness of communication and understanding of project goals.
2. Resource Constraints: Despite efforts to optimize resource allocation, constraints such as budget limitations or unexpected events may impact the project's ability to allocate resources efficiently.
3. Dependency on Technology: The robustness of the system architecture is dependent on the chosen technologies. Changes in technology or unforeseen technological issues could pose challenges to scalability and reliability.
4. Usability Challenges: Despite efforts to create a user-friendly interface, individual user preferences and varying levels of technical proficiency may result in usability challenges for some users.
5. Testing Limitations: The effectiveness of load testing and performance optimization may be limited by unforeseen circumstances or complexities within the system architecture.
6. Model Accuracy Challenges: Despite robust testing and validation processes, the linear regression model's accuracy may be influenced by the complexity of learner behavior and the dynamic nature of educational contexts.
7. Integration Complexity: Achieving seamless integration with existing educational platforms may face challenges due to diverse technological landscapes and interoperability issues.
8. Continuous Improvement Challenges: Despite efforts for continuous improvement, external factors or resistance to change may pose challenges in maintaining the system's relevance and effectiveness over time.

11.3 APPLICATIONS:

- Education Sector
- Corporate Training and Development
- Healthcare Training
- Professional Development Programs
- Human Resources
- E-learning Platforms
- Customer Service Training
- Military Training

Chapter 12

CONCLUSION

12.1 CONCLUSION

In summary, this project represents a pioneering approach to educational assessment, delving beyond traditional metrics by integrating the nuanced dimensions of learner confidence and confusion. By capturing the intricacies of correctness, option changes, and time per question, the project aspires to create a more comprehensive understanding of the learner's emotional and cognitive states during assessments.

The envisaged advantages of the project extend beyond the educational landscape, promising to be a transformative force in adaptive learning systems. The system's ability to dynamically adjust content based on individual affective states opens avenues for personalized learning experiences, presenting a paradigm shift in how education is tailored to the unique needs of each learner.

While the project showcases promising prospects, it is essential to acknowledge its inherent challenges. These challenges, however, pave the way for innovative solutions and improvements that can propel the project towards its full potential.

12.2 Future Work:-

In the realm of future work, this project opens avenues for refinement and expansion. Prospective endeavors could involve leveraging advanced machine learning techniques for more precise affective state analysis, potentially incorporating multimodal data sources like video or biometrics. Longitudinal studies may track changes in learners' affective states over time, providing insights into the effectiveness of interventions. Seamless integration with Learning Management Systems (LMS) and the exploration of adaptive feedback mechanisms are also promising directions. Addressing privacy and ethical considerations, adapting the project for online learning, and investigating cross-cultural validity are critical areas for further exploration. Real-time intervention strategies and a quantitative-qualitative hybrid approach could further enhance the project's impact, revolutionizing our understanding and support for learners in diverse educational contexts.

Chapter 13

References

[1] K. Kelkar and J. Bakal, "Adaptive MCQ Test Generation Based on Affective State Feedback," ***Seybold Journal***, 2020. [Online]. Available: https://jusst.org/adaptive-mcq-test-generation-based-on-affective-state-feedback/

[2] A. Smith and R. Patel, "Artificial Intelligence-Enabled Adaptive Learning Systems: A Review," ***International Journal of Educational Technology***, 2020.

[3] S. Gupta and N. Sharma, "Enhancing Adaptive Learning Using AI-Based Question Selection Algorithms," ***Journal of Educational Computing Research***, 2021.

[4] L. Chen and Y. Wang, "Integration of AI in Educational Assessment: A Systematic Literature Review," ***Computers & Education***, 2021.

[5] J. Kim and S. Lee, "Adaptive Learning Systems: An Overview of AI-Powered Techniques," ***Journal of Interactive Learning Research***, 2022.

[6] H. Zhang and Q. Liu, "AI-Driven Personalized Learning: A Comparative Study of Approaches," ***Educational Technology Research and Development***, 2022.

[7] M. Rahman and S. Das, "Adaptive Learning Algorithms using Machine Learning for MCQ Generation," ***International Journal of Artificial Intelligence in Education***, 2023.

[8] C. Lee and T. Wu, "Implementing AI-Based Adaptive Testing in Educational Environments," ***Computers in Human Behavior***, 2023.