**A Project Report**

**On**

**Adaptive Online Platform for Enhanced Teaching and Learning**

submitted for partial fulfillment of the requirement for the award of the degree of

**Bachelor of Technology**

**In**

**Computer Science**



**KIET Group of Institutions, Ghaziabad**

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

**May, 2022**

**Submitted by**

Amod Katiyar (2000290120022)

Himanshu Kumar (2000290120076)

Aniket Bhardwaj (2000290120024)

**Under supervision of:**

Dr. Gaurav Dubey

**TABLE OF CONTENTS**

DECLARATION…………………………………………………………………….

CERTIFICATE………………………………………………………………………

ACKNOWLEDGEMENTS………………………………………………………….

ABSTRACT………………………………………………………………………....

LIST OF FIGURES………………………………………………………………….

LIST OF TABLES……………………………………………………………………

LIST OF ABBREVIATIONS……………………………………………………….

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION TO PROJECT

1.2 PROJECT CATEGORY (INTERNET BASED, APPLICATION OR SYSTEM

DEVELOPMENT, RESEARCH BASED, INDUSTRY AUTOMATION, NETWORK OR

SYSTEM ADMINISTRATION)

1.3 OBJECTIVES

1.4 PROBLEM FORMULATION

1.5 PROPOSED SYSTEM

1.6 UNIQUE FEATURES OF THE SYSTEM

CHAPTER 2. REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

2.1 FEASIBILITY STUDY (TECHNICAL, ECONOMICAL, OPERATIONAL)

2.2 SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT WHICH MUST

INCLUDE THE FOLLOWING:

(DATA REQUIREMENT, FUNCTIONAL REQUIREMENT, PERFORMANCE

REQUIREMENT

, MAINTAINABILITY REQUIREMENT, SECURITY REQUIREMENT)

2.3 SDLC MODEL TO BE USED

CHAPTER 3. SYSTEM DESIGN

3.1 DETAIL DESIGN

3.2 SYSTEM DESIGN USING DFD LEVEL 0 AND LEVEL 1

3.3 USE CASE DIAGRAM

3.4 DATABASE DESIGN

3.4.1 ER DIAGRAMS

CHAPTER 4. IMPLEMENTATION, TESTING, AND MAINTENANCE

4.1 INTRODUCTION TO LANGUAGES, TOOLS AND TECHNOLOGIES USED FOR

IMPLEMENTATION

4.2 TESTING TECHNIQUES AND TEST CASES USED

4

CHAPTER 5. RESULTS AND DISCUSSIONS

5.1 USER INTERFACE REPRESENTATION (OF RESPECTIVE PROJECT)

5.1.1 BRIEF DESCRIPTION OF VARIOUS MODULES OF THE SYSTEM

5.2 SNAPSHOTS OF SYSTEM WITH BRIEF DETAIL OF EACH

5.3 BACK-ENDS REPRESENTATION (DATABASE TO BE USED)

5.3.1 SNAPSHOTS OF DATABASE TABLES WITH BRIEF DESCRIPTION

CHAPTER 6. CONCLUSION AND FUTURE SCOPE

REFERENCES (REFERENCES SHOULD BE IN IEEE STANDARD)

**DECLARATION**

I/We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name:-

Roll No.:-

Date:-

## CERTIFICATE

This is to certify that Project Report entitled “**Adaptive Online Platform for Enhanced Teaching and Learning”** which is submitted by **Amod Katiyar, Aniket Bhardwaj, Himanshu Kumar** in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

**Date: Supervisor**

Dr. Gaurav Dubey

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to Professor Dr. Gaurav Dubey, Department of Computer Science, KIET, Ghaziabad, for his/her constant support and guidance throughout the course of our work. His/Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Pradeep Kumar Singh, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date :

Signature:

Name : :

Roll No.:

**ABSTRACT**

This project introduces a user-friendly platform aimed at assisting students in navigating the diverse landscape of programming languages. By focusing on industry trends, the website provides clear and concise language recommendations, offering valuable insights into the rapidly evolving tech environment. The primary objective is to streamline students' learning journeys by creating a centralized hub that not only identifies current language demands but also serves as a strategic guide, helping them connect their passion with proficiency.

In addition to its recommendation feature, the platform goes beyond the basics, encouraging students to explore learning paths aligned with emerging trends. By leveraging industry data, the platform empowers students to make informed decisions, fostering a deeper connection between their interests and the ever-changing world of technology. The platform aims to be a comprehensive resource, offering guidance on honing fundamental concepts and adapting to new industry demands, ensuring that students stay ahead in the competitive landscape of programming.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Description** | **Page No.** |
| 1.1 | Agile Method (SDLC) | 4 |
| 3.2 | Data Flow Diagram (level 0) | 5 |
| 3.2 | Data Flow Diagram (level 1) | 7 |

**LIST OF ABBREVIATIONS**

1. UI: User Interface

2. DFD: Data Flow Diagram

3. ER: Entity-Relationship

4. RDBMS: Relational Database Management System

5. SRS: Software Requirement Specification

6. SDLC: Software Development Life Cycle

7. HTML: Hypertext Markup Language

8. CSS: Cascading Style Sheets

9.JS: JavaScript

10.AI: Artificial Intelligence

11.ML: Machine Learning

12.IEEE: Institute of Electrical and Electronics Engineers

13.API: Application Programming Interface

**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

In today's fast-paced tech world, students face the challenge of choosing relevant programming languages to learn. This project introduces a user-friendly platform designed to simplify this decision-making process. By focusing on real-time industry trends, the platform provides clear language recommendations, aiming to bridge the gap between students' passion and the demands of the ever-evolving technology sector. This report explores the development and functionality of the platform, highlighting its significance in guiding students towards informed choices and proficiency in programming.

**1.2 Project Category:** The project falls under the category of an internet-based application development initiative. It revolves around the creation of a user-friendly platform accessible through the web to assist students in navigating programming language choices.

**1.3 Objectives:** The primary objectives of the project are:

1. Simplifying the process of selecting programming languages for students.
2. Providing real-time insights into industry trends.
3. Bridging the gap between students' passion and the demands of the technology sector.
4. Offering a centralized hub for language recommendations and strategic guidance.

**1.4 Problem Formulation:** The project addresses the challenges students face in selecting programming languages amid a myriad of choices. The goal is to streamline decision-making, considering both individual interests and the dynamic demands of the industry.

**1.5 Proposed System:** The proposed system is an internet-based platform that offers concise language recommendations and insights into current industry trends. It serves as a guide for students to make informed decisions about which programming languages to learn based on their interests and the evolving needs of the tech sector.

**1.6 Unique Features of the System:** The system's unique features include:

1. Industry trend analysis.
2. Clear and concise language recommendations.
3. A user-friendly interface for easy navigation.
4. Encouraging exploration of dynamic learning paths.
5. Bridging the gap between passion and proficiency in programming.

**Chapter 2: Requirement Analysis and System Specification**

**2.1 Feasibility Study**

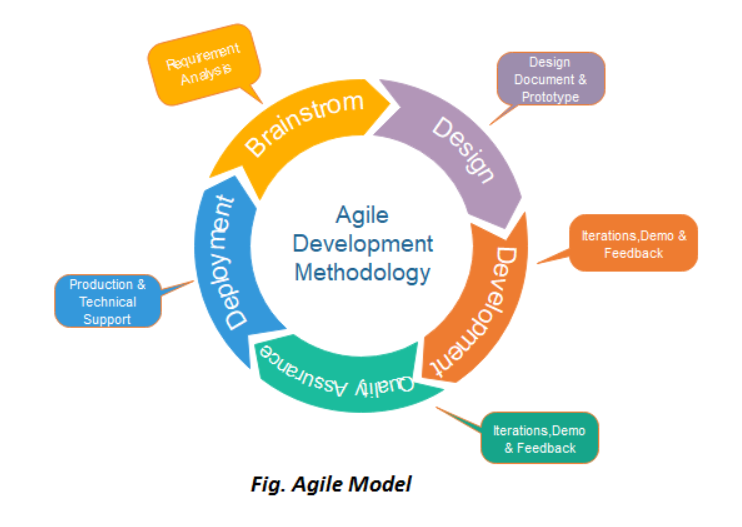
* **Technical Feasibility**: The project's technical feasibility is high, given the availability of web development frameworks that can support real-time data processing and analytics. The development team needs to have skills in web development, database management, and data analysis.
* **Economical Feasibility**: Economically, the project requires initial investment in web hosting, domain registration, and potentially premium APIs for accessing real-time industry trends. However, these costs are manageable and can be offset by potential monetization strategies, such as subscriptions or advertisements.
* **Operational Feasibility**: Operationally, the project is feasible as it aligns with the increasing demand for tech education resources. The platform's ease of use and the value it provides make it likely to be welcomed by its target users – students and learners interested in programming.

**2.2 Software Requirement Specification Document**

* **Data Requirement**
  + Real-time industry trends data.
  + Comprehensive database of programming languages and their resources.
  + User data for personalized recommendations.
* **Functional Requirement**
  + User account creation and management.
  + Dynamic display of programming languages and trends.
  + Search and filter options for languages and resources.
  + Feedback and rating system for resources.
* **Performance Requirement**
  + The platform should be capable of handling high traffic volumes without significant lag.
  + Real-time updates and notifications without system overload.
  + Quick response time for user queries and actions.
* **Maintainability Requirement**
  + The system should be designed for easy updates and scalability.
  + Regular updates for the tools and technologies used.
  + Efficient error logging and debugging capabilities.
* **Security Requirement**
  + Implementation of secure authentication mechanisms.
  + Encryption of sensitive user data.
  + Regular security audits and compliance with data protection regulations.

**2.3 SDLC Model to be Used**

The Agile SDLC model will be utilized for this project. Agile is suitable due to its flexibility, emphasis on customer feedback, and iterative approach, allowing for continuous improvement and adaptation to changing requirements. This model supports the dynamic nature of the project's goals, especially considering the need for regular updates based on real-time industry trends and user feedback.



**CHAPTER 3: SYSTEM DESIGN**

**3.1 Detail Design**

In the detail design phase, we focus on elaborating the architecture, components, and interactions of the system. This includes designing the database schema, defining APIs and endpoints, outlining user interfaces, and specifying the integration of external services or modules.

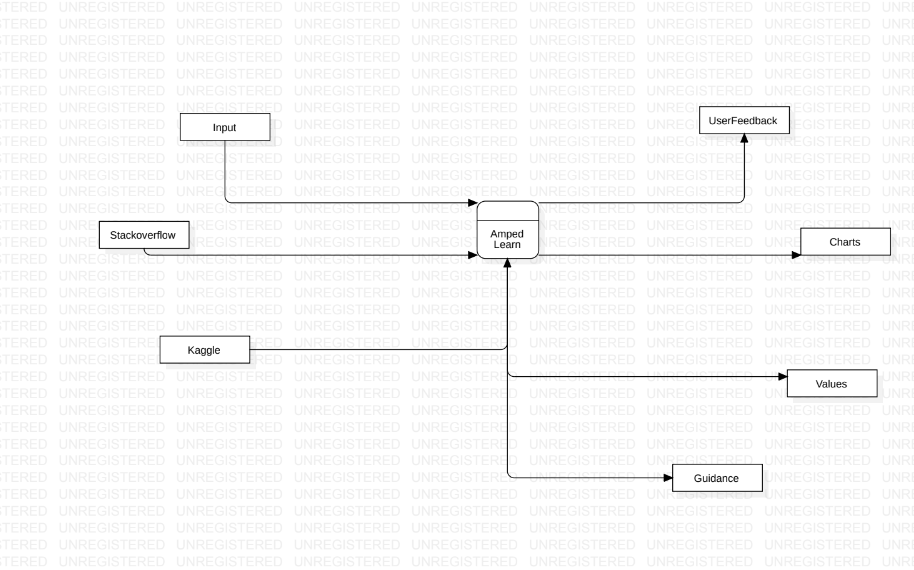
**Components of Detail Design:**

1. **Database Schema Design:** Define the structure of the database including tables, relationships, constraints, and indexes. Use MongoDB for scalability and flexibility in handling structured and unstructured data related to users, courses, progress tracking, and system configurations.
2. **API Design:** Design RESTful APIs for communication between frontend and backend components. Define endpoints for user authentication, course management, progress tracking, recommendation system interactions, and community features like forums and chats.
3. **User Interface Design:** Create wireframes and mock-ups for the user interface using tools like Figma or Adobe XD. Design intuitive and responsive interfaces for user registration, login, dashboard, course catalogue, progress dashboard, community forums, and admin panels.
4. **Integration Design:** Plan integration with external services such as NLP libraries for text analysis, machine learning models for recommendation systems, data visualization tools for analytics, and email services for notifications and alerts.

**3.2 System Design Using DFD Level 0 and Level 1**

**DFD Level 0: System Overview**

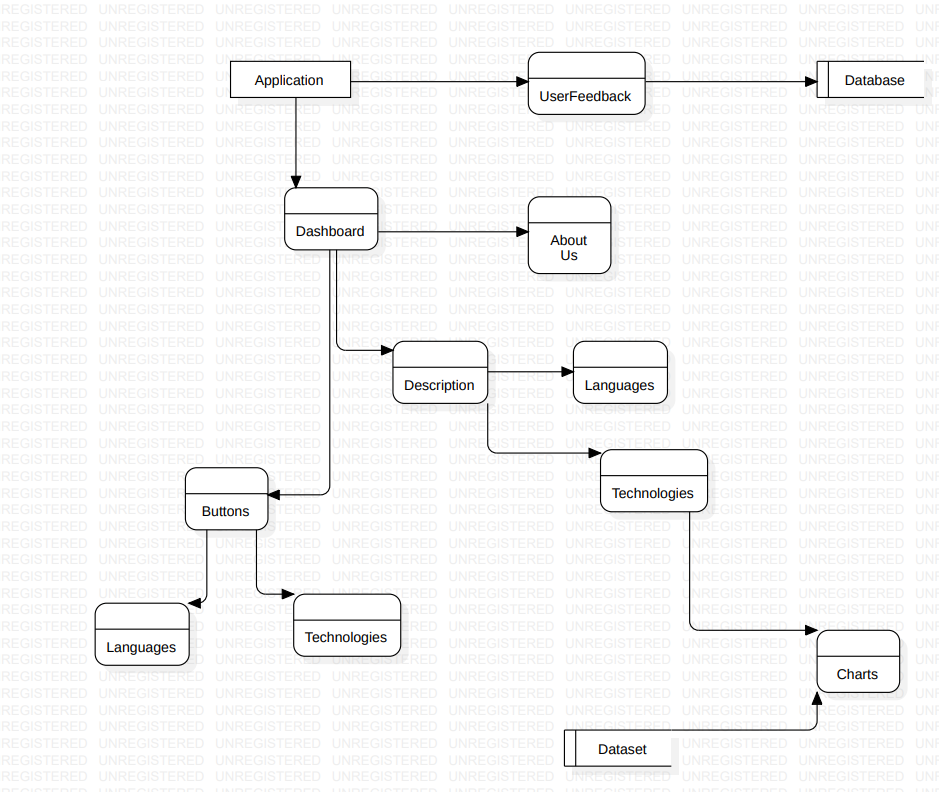
* **Processes:** Represented as circles, processes include user authentication, course management, recommendation system, community interactions, and admin functionalities.
* **Data Stores:** Represented as rectangles, data stores include the MongoDB database for storing user data, course information, progress details, and system configurations.
* **External Entities:** Represented as squares, external entities include users, admins, external APIs for NLP and machine learning, and external databases if required.



**DFD Level 1: Detailed Process Decomposition**

* **User Authentication Process:** Includes sub-processes for user registration, login, password management, and email verification.
* **Course Management Process:** Includes sub-processes for course creation, editing, deletion, and content updates.
* **Recommendation System Process:** Involves sub-processes for data analysis, recommendation generation, and personalized course suggestions.
* **Community Interaction Process:** Includes sub-processes for forum postings, chat interactions, user collaborations, and content sharing.
* **Admin Management Process:** Includes sub-processes for user management, content moderation, analytics monitoring, and system configurations.

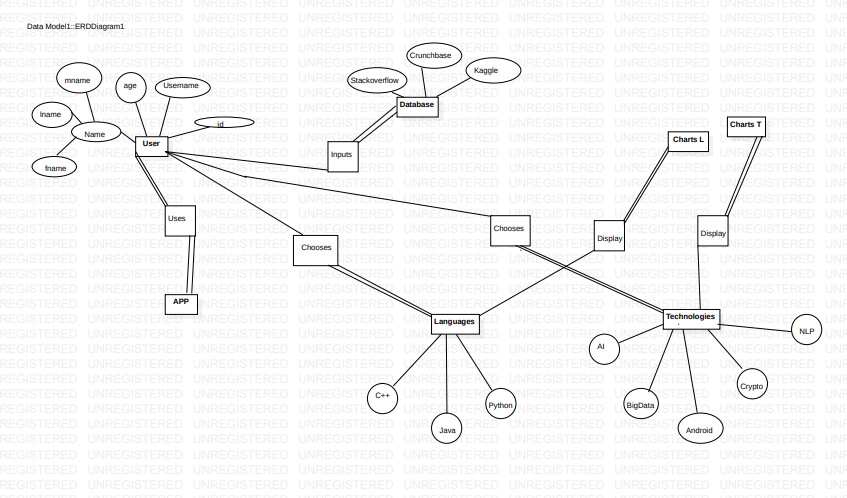
These DFDs provide a structured view of the system's processes, data flows, and interactions at different levels of abstraction, aiding in understanding system functionalities and dependencies for further implementation and testing phases.



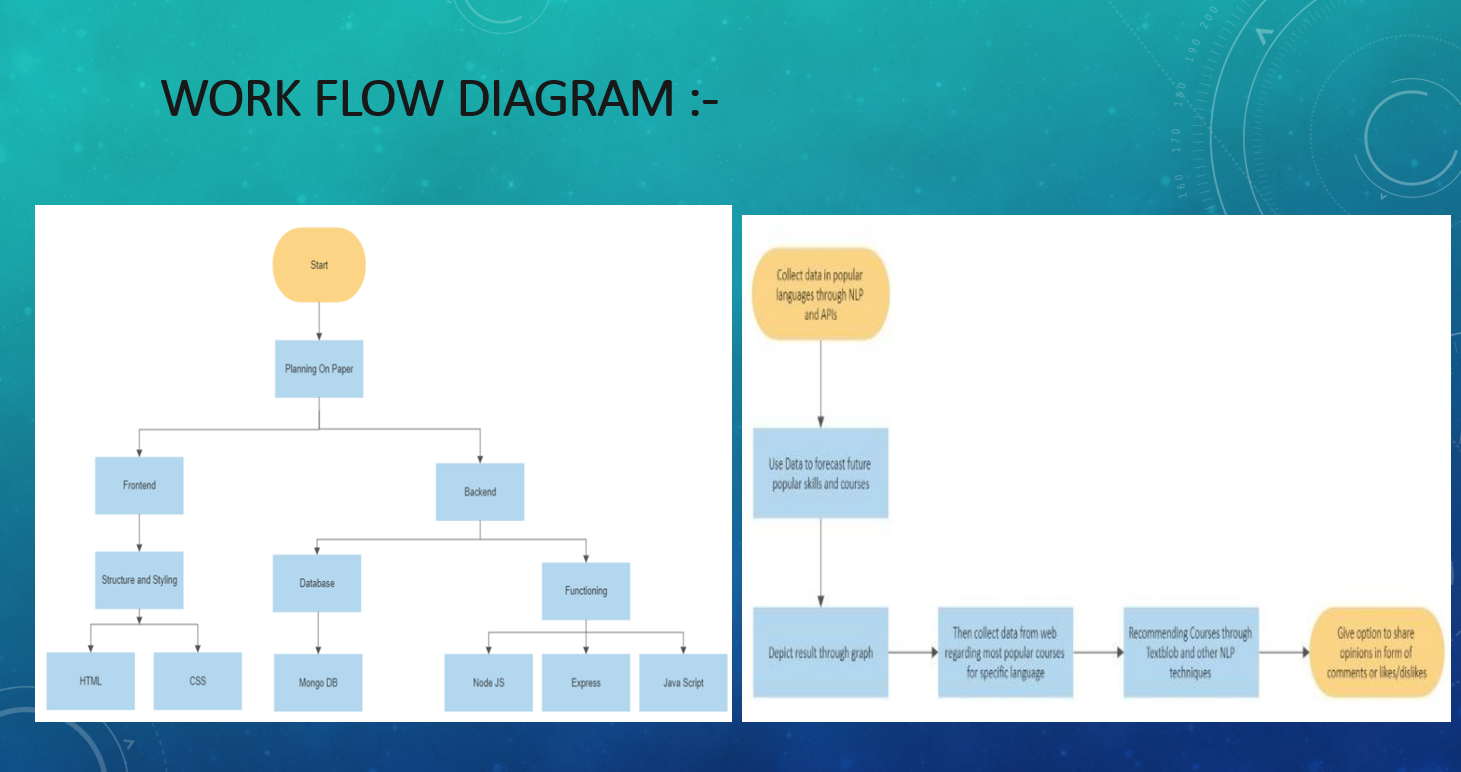
**3.3 DATABASE DESIGN**

MongoDB has been used as the database.

MongoDB is a popular NoSQL database used for its flexible, document-oriented approach. In contrast to traditional relational databases, MongoDB stores data in JSON-like documents, making it well-suited for handling diverse data types and schemas. Its scalability, high performance, and ease of use have made it a preferred choice for many modern applications, particularly those with large volumes of unstructured or semi-structured data.



**Workflow Diagram**



**CHAPTER 4: IMPLEMENTATION TESTING, AND MAINTENANCE**

**4.1 Introduction to Languages, Tools, and Technologies Used for Implementation**

In the development of our platform, we have carefully selected a range of languages, tools, and technologies to ensure robustness, scalability, and user-friendliness. Below is an overview of the key components used in the implementation:

**Programming Languages:**

* JavaScript: JavaScript plays a central role in both frontend and backend development. It is a versatile language known for its ability to create interactive and dynamic web pages. For backend development, we utilize Node.js, which allows us to run JavaScript server-side, enabling efficient handling of server requests and real-time data processing.

**Frameworks:**

* React: React.js is a powerful JavaScript library for building user interfaces. We leverage React for our frontend development to create modular and reusable components, resulting in a responsive and intuitive user experience.
* Express.js: Express is a minimalist web framework for Node.js, providing essential features for building robust web applications and APIs. We use Express to handle routing, middleware, and database interactions on the server side.

**Web Technologies:**

* HTML5 (Hypertext Markup Language): HTML5 forms the backbone of our web pages, defining the structure and content elements that users interact with.
* CSS3 (Cascading Style Sheets): CSS3 is used for styling and layout purposes, ensuring visual consistency and aesthetics across different devices and screen sizes.
* MongoDB: MongoDB, a NoSQL database, serves as our data store for storing user information, course data, and progress tracking. Its flexible schema and scalability make it ideal for our needs.
* NLP (Natural Language Processing): NLP techniques are employed to analyse and extract insights from textual data. This aids in building a recommendation system that suggests relevant courses and learning paths based on user preferences and industry trends.
* XGBoost and Decision Trees: Machine learning algorithms such as XGBoost and Decision Trees are utilized for data analysis, trend forecasting, and generating personalized recommendations for users.

By leveraging this stack of languages, frameworks, and technologies, we aim to create a feature-rich and user-centric platform that empowers learners to explore, compare, and excel in various programming languages and domains while providing valuable insights and guidance throughout their learning journey.

**Top of Form**

**4.2 Testing Techniques and Test Cases Used**

• **4.2.1 Testing Techniques-**

• **Functional Testing:**

Functional testing ensures that each function of the system operates as intended. In our testing approach, we rigorously examined key functions and modules to guarantee seamless performance.

• **Usability Testing:**

Usability testing was employed to assess the user-friendliness of the system. Many criteria were considered during this testing phase to enhance the overall user experience.

• **Compatibility Testing:**

To ensure widespread accessibility, we conducted compatibility testing across various browsers, operating systems, and devices. The application was tested on Google Chrome, Microsoft Edge, Windows 10, Desktop, Laptops ensuring consistent functionality.

• **Performance Testing:**

While primarily focusing on manual testing, we incorporated performance considerations.

• **4.2.2 Test Cases-**

• **Test Case Design:**

Our test case design followed a meticulous methodology to cover all aspects of system functionality. We used Microsoft Excel to document detailed test cases for comprehensive coverage.

• **Test Case Execution:**

Execution involved setting up a controlled environment with specific configurations. Each test case was executed systematically, and the outcomes were recorded for analysis.

• **Results and Observations:**

Upon test case execution, we summarized results, highlighting any discrepancies, issues, or noteworthy observations. This comprehensive analysis forms the basis for refining and improving the system.

• **Regression Testing:**

To ensure that new updates or fixes did not adversely impact existing functionalities, we conducted regression testing. This step was crucial to maintain the integrity of the system across iterations.

**4.3 Maintenance**

The maintenance phase involves continuous monitoring, updates, and improvements to ensure the system's longevity and relevance. Regular checks for data quality, model performance, and software updates are implemented. User feedback and changing environmental conditions are considered for ongoing enhancements to the prediction system. Maintenance activities aim to address emerging challenges, implement security patches, and incorporate advancements in technology for sustained system effectiveness.

**CHAPTER 5. RESULTS AND DISCUSSIONS**

**• 5.1 User Interface Representation (of Respective Project)**

The user interface (UI) of the pollution forecasting system is thoughtfully designed for user-friendly interaction. It encompasses intuitive navigation, clear visualizations, and accessibility features to ensure a seamless experience for users. The design prioritizes displaying key information, trends, and alerts to facilitate effective decision-making.

**• 5.1.1 Brief Description of Various Modules of the System**

1. **User Authentication and Management Module:**
   * This module handles user registration, login, and profile management functionalities.
   * It includes features such as email verification, password management, and user role assignments for admin and regular users.
2. **Language Comparison and Analysis Module:**
   * This module allows users to compare different programming languages based on various parameters such as popularity, job opportunities, salary trends, and industry demand.
   * It integrates data analysis and visualization techniques to present insights in a user-friendly manner.
3. **Learning Resources Recommendation System:**
   * Using natural language processing (NLP) techniques and machine learning algorithms, this module recommends learning resources (courses, tutorials, articles) based on user preferences, historical data, and industry trends.
   * It provides personalized recommendations to help users choose the most suitable learning path for their goals.
4. **Progress Tracking and Performance Analysis Module:**
   * This module tracks user progress within courses and learning modules, recording completion status, quiz scores, and time spent on different topics.
   * It generates performance reports and insights to help users understand their strengths, weaknesses, and areas for improvement.
5. **Community Interaction and Support Module:**
   * Facilitating community engagement, this module includes features such as discussion forums, chat rooms, and peer-to-peer support channels.
   * Users can interact with peers, share experiences, ask questions, and provide assistance, fostering a collaborative learning environment.
6. **Admin Dashboard and Content Management Module:**
   * This module provides administrators with a centralized dashboard to manage users, content, and system configurations.
   * Admins can add/edit/delete courses, monitor user activity, analyse system metrics, and ensure overall platform functionality and security.
7. **Real-time Data Analysis and Visualization Module:**
   * Integrating data analytics tools and visualization libraries, this module generates real-time insights into language usage trends, job market demands, and user engagement metrics.
   * Interactive charts, graphs, and dashboards enhance data interpretation and decision-making for both users and administrators.
8. **Certification and Assessment Module:**
   * Offering certification programs and assessment tools, this module allows users to validate their skills and progress through structured learning pathways.
   * Users can take quizzes, exams, and practical assignments to earn certifications and showcase their proficiency to potential employers.

These interconnected modules form the backbone of the system, providing a comprehensive learning ecosystem that empowers users with knowledge, skills, and networking opportunities in the tech industry.

**CHAPTER 6. CONCLUSION AND FUTURE SCOPE**

**6.1 CONCLUSION**

The final platform would be able to overcome all of the problems that students are currently experiencing in their learning. The students will have a clear idea of which domain to choose and a perfect road map to help them achieve their goals and land a job in an MNC.

The recommendation system will help students understand which domains are currently thriving and what job profiles are needed in the industry.

This allows the student to easily plan their journey ahead and learn something that they are truly interested in rather than something that they will not feel like continuing in the future.

**6.2 Future Scope**

**Mobile App Development:** Develop a mobile app version of the platform to cater to users' on-the-go learning needs. Ensure seamless synchronization of progress, notifications, and access to learning materials across multiple devices.

**Enhanced Learning Paths:** Expand the platform to offer curated learning paths tailored to specific career goals and industry sectors. Integrate advanced analytics to track user preferences and suggest customized learning journeys.

**Advanced Certification Programs:** Develop advanced certification programs in collaboration with industry experts and organizations. Offer specialized certifications in emerging technologies and niche domains to enhance user credibility and career prospects.

**REFERENCES**

[1] isi.edu/nsnam/ns/tutorial Marc Greis tutorial on ns2 on, 12 October 2013.

[2] Matthias Transier “Ns2 tutorial running simulations”.

[3] D. Kim, J. Garcia and K. Obraczka, “Routing Mechanisms for Mobile Ad Hoc Networks based on the Energy Drain Rate”, IEEE Transactions on Mobile Computing. Vol 2, no 2, 2003, pp.161-173.

[4] C.E. Perkins & P. Bhagwat, “Highly Dynamic Destination Sequence-Vector Routing (DSDV) for Mobile Computers”, Computer Communication Review, vol. 24, no.4, 1994, pp. 234-244.

[5] C.E. Perkins and E.M. Royer, “Ad-Hoc on-Demand Distance Vector Routing,” Proc. Workshop Mobile Computing Systems and Applications (WMCSA ’99), Feb. 1999 pp. 90-100.

[6] David B. Johnson and David A. Maltz. “Dynamic source routing in ad hoc wireless networks”, Mobile Computing, Kluwer Academic Publishers. 1996 pp.153–181, 1996.

[7] M. S. Corson, J. P. Maker and G. H. Cirincione , "Internet-Based Mobile Ad Hoc Networking," IEEE Internet Computing, Vol. 3, no. 4, July-August 1999, pp. 63-70.

[8] V. Ramesh, Dr. P. Subbaiah, N. Koteswar Rao and M. Janardhana Raju,"Performance comparison and analysis of DSDV and AODV for MANET," (JJCSE) International Journal on Computer Science and Engineering , vol. 02 , pp. 183-188, 2010 Md. Monzur Morshed; Franz I. S. Ko; Dongwook Lim; Md. Habibur Rahman; Md. Rezaur Rahman Mazumder; Jyotirmoy Ghosh.