A Project Report

On

“Skill Analyzer”

by

M.Sankar Narayana Reddy	22SE02ML041
D.Amarnadh	22SE02ML014

under the mentorship of

Mrs.Priyanka vashi

Assistant Professor, School of Engineering

APRIL 2024

P P SAVANI SCHOOL OF ENGINEERING

P P SAVANI UNIVERSITY

NH NO.: 8, VILLAGE: DHAMDOD, TA. MANGROL, NEAR KOSAMBA, SURAT – 394 125. (GUJARAT).

CERTIFICATE

This is to certify that the Project Report submitted by **MARAM SANKAR NARAYANA REDDY(22SE02ML041)** to the **P P SAVANI UNIVERSITY** for the partial fulfilment of the subject credit requirements is a bonafied work carried out by the student.

This is to further certify that I have been supervising the Major/Minor Project of **MARAM SANKAR NARAYANA REDDY(22SE02ML041).**

The contents of this report, in full or in parts, have not been submitted to any other Institute or University for award of any degree, diploma or titles.

Sign of Faculty Mentor :

Name of Faculty Mentor**: Mrs.Priyanka vashi**

Date:

CERTIFICATE

This is to certify that the Project Report submitted by **AMARNADH(22SE02ML014)** to the **P P SAVANI UNIVERSITY** for the partial fulfilment of the subject credit requirements is a bonafied work carried out by the student.

This is to further certify that I have been supervising the Major/Minor Project of **AMARNADH(22SE02ML014).**

The contents of this report, in full or in parts, have not been submitted to any other Institute or University for award of any degree, diploma or titles.

Sign of Faculty Mentor :

Name of Faculty Mentor: **Mrs.Priyanka vashi**

Date:

ACKNOWLEDGEMENT

This report would not have been possible without my teachers who were always there when I needed them the most. I take this chance to acknowledge them and extend my sincere gratitude for helping me make this Report a possible.

I wish to thank my faculty mentor **Mrs.Priyanka vashi,** Assistant Professor, School of Engineering. It has been an honor to learn under their mentorship.

As my mentor, he has constantly motivated me to remain focused on achieving my goal. Their observations and guidance helped me to establish the overall direction of the report and to move forward with learning in depth. Their vital support at each juncture, which culminated in successful completion of my project work. I express my sincere gratitude to them for constant support during the project work.

I am also thankful to faculty members of the department for constant support and guidance.

I am thankful to Dean, School of Engineering for his initiative of imparting Project during tenure of your study making us learn new things and to help us in expanding our horizons.

Name of Student : **SANKAR NARAYANA REDDY**
Enrollment No : **22SE02ML041**

Name of Student : **AMARNADH**
Enrollment No :**22SE02ML014**

ABSTRACT

Adaptation with Cloud increase the delivery rate and also opens up the market. It reduces the CDN cost and hosting cost of data to minimal. The product aims to solve the issues around this by leveraging the power of AWS Cloud. It is aim to deliver an automated solution which has ability to allow users to upload the content from a Web Application and provides interface to viewers to view content. It also provides insights on content viewed and the what interest the users the most. This solution is completely server less which means companies do not need to worry about provisioning resources and planning about CAPEX cost. It aims to provide solution in parts where the customers can utilize and take advantage of the individual components.

The project has Video convert pipeline, Platform to stream the data, Analytics APIs to develop Admin panel and Authenticated access to AWS native services.
The Survey of ECC girl's hostel-3 is carried out using total station and reduced level of girl's hostel 3 is calculated up to floor finish (FF). The site investigation in done and the safe bearing capacity of soil including settlements and shear failure criteria is calculated and appropriate depth of foundation is decided.

CONTENTS

1.INTRODUCTION

In today's fast-paced and competitive world, both students and professionals constantly seek ways to assess their knowledge and skills. Traditional evaluation systems, such as written tests or instructor-led quizzes, are often time-consuming and limited in scalability. With the rise of digital education and e-learning platforms, there is an urgent need for intelligent, automated systems that can quickly evaluate an individual's skills and provide instant feedback.

The **Skill Analyzer** is a web-based quiz application developed to address this very need. It is designed to offer users an interactive environment where they can attempt topic-specific quizzes, receive instant scoring, and get detailed feedback on their strengths and weaknesses. The system can be used in educational institutes, training centers, and even for self-assessment by individuals.

This project not only helps users understand their current knowledge level but also encourages continuous learning by providing clear performance tracking and goal-oriented quizzes. It's easy to use, efficient, and designed to deliver meaningful results in real-time.

2.OBJECTIVES

The primary goal of the **Skill Analyzer** is to provide a platform where knowledge and skills can be assessed through an automated quiz mechanism. Below are the detailed objectives:

- **Automated Skill Assessment**

  To automate the evaluation process using quiz-based questions so users can self-assess their understanding of various subjects without manual intervention.
- **Instant Feedback and Results**

  To offer real-time result computation and feedback, helping users immediately understand their performance and areas where improvement is needed.
- **User-Friendly Interface**

  To design an intuitive and accessible web interface that ensures smooth navigation for users of all age groups and technical backgrounds.
- **Performance Tracking**

  To allow users to track their progress over time by storing and displaying their historical performance data and growth trajectory.
- **Scalability and Flexibility**

  To ensure the system can handle multiple quizzes and users simultaneously and can be extended to different domains, subjects, or difficulty levels.
- **Secure User Data Management**

  To store all user data securely, including quiz history, scores, and personal information, while ensuring compliance with

data privacy best practices.

- **Customizable Quiz Content**

  To provide administrators and educators the ability to add, update, and delete quiz questions to maintain relevance and difficulty levels.

- **Adaptive Learning Potential**

  Laying the groundwork for future AI-based enhancements that can recommend quizzes based on user weaknesses .

3. SYSTEM ANALYSIS

a) Identification of Need.

In the modern educational and professional environment, skill assessment plays a vital role in determining an individual's capabilities. However, most traditional assessment methods are static, manual, and lack real-time feedback, which reduces engagement and effectiveness. With the increasing demand for digital learning tools, there is a clear need for a web-based system that enables users to evaluate their knowledge efficiently.

The **Skill Analyzer** addresses this gap by providing a flexible, easy-to-use platform for automated quiz-based assessments. This system reduces the workload on educators and organizations while empowering users to self-assess, improve their skills, and track progress over time.

b) Preliminary Investigation

The preliminary investigation was conducted through:

- **Surveys and Feedback** from students and professionals who actively participate in online learning.
- **Observation of Existing Platforms** such as Google Forms, Kahoot, and Quizizz, identifying their limitations in personal progress tracking, feedback depth, and customization for specific subjects.
- **Consultation with Educators** to gather insights on the shortcomings of manual testing and the benefits of automation.

Findings confirmed the demand for a centralized, secure, and user-centric quiz platform that is scalable, customizable, and provides instant feedback.

c)Feasibility Study

- **Technical Feasibility**

  The project is technically feasible using modern web technologies such as HTML, CSS, JavaScript, and backend frameworks like Python/Django or PHP/Laravel.

- **Operational Feasibility**

  The system will be easy to operate for users, administrators, and quiz creators.Minimal training is required for end-users due to the intuitive interface.

- **Economic Feasibility**

  Development cost is reduced using open-source tools.Long-term maintenance costs are minimal due to automation and modularity.Offers high value for low investment in educational institutions and training centers.

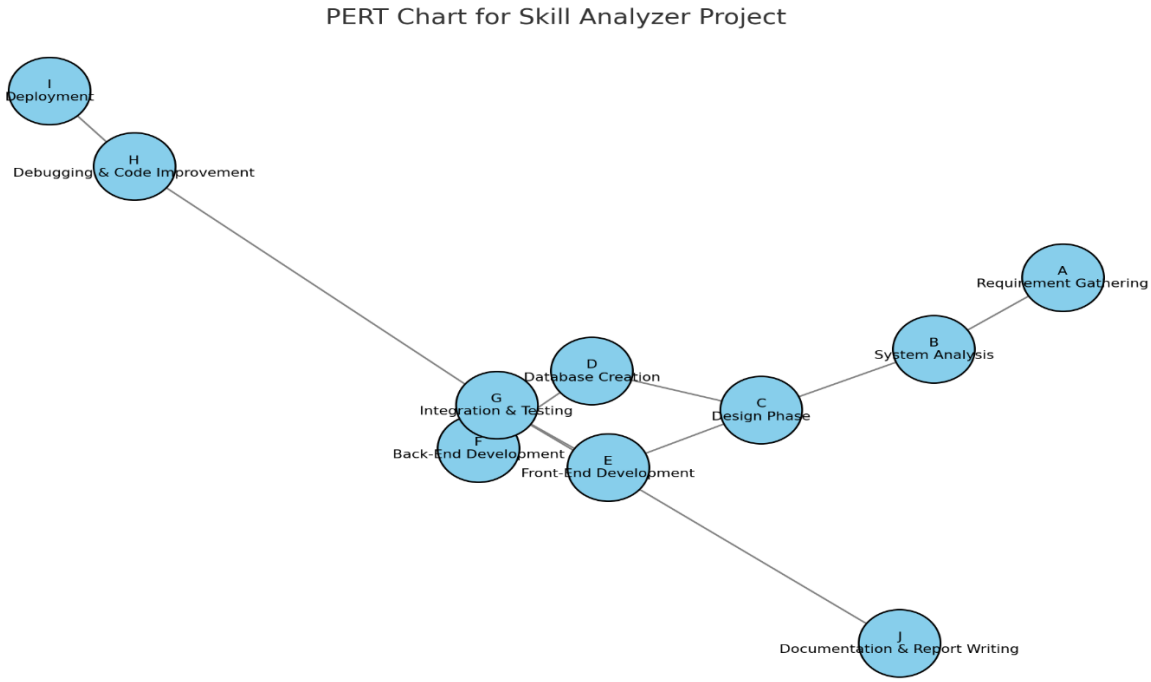
d)Project Planning

The project is broken down into the following phases:

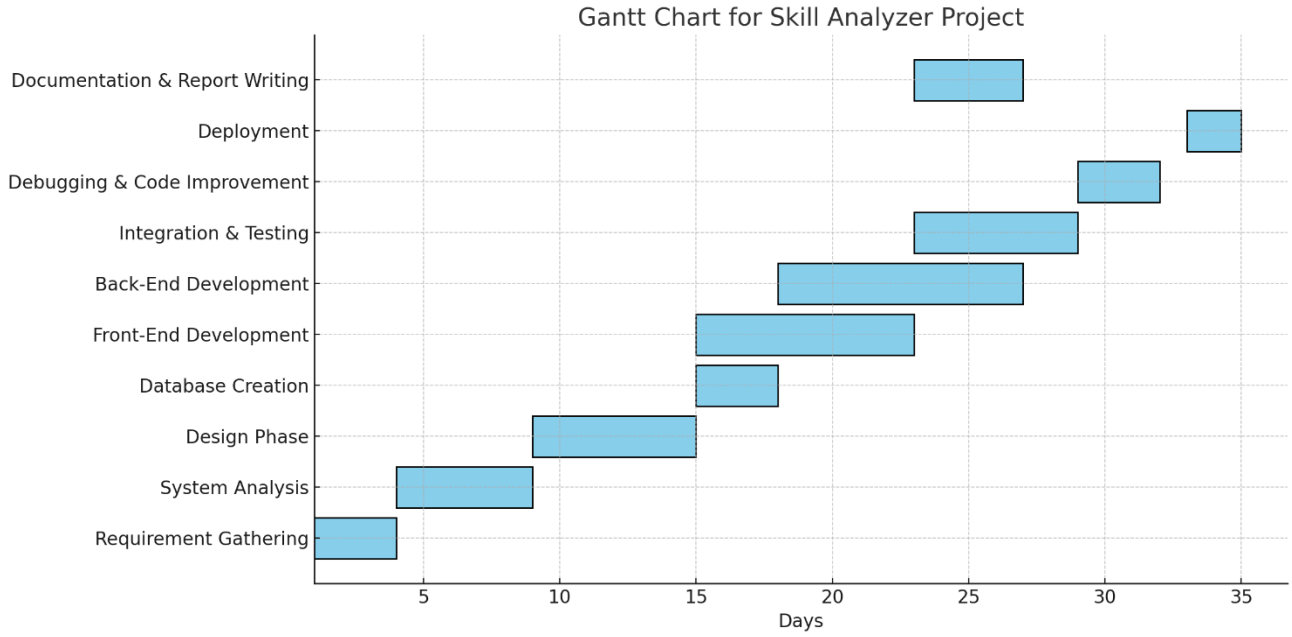
- **Requirement Analysis:** Understanding user needs and system specifications.
- **System Design:** Designing the database schema, user interface, and logic flow.
- **Development:** Writing code for both front-end and back-end modules.
- **Testing:** Running unit and system tests to ensure quality.
- **Deployment:** Launching the system for real-world use.
- **Maintenance:** Continuous monitoring, bug fixing, and feature enhancement.

e) Project Scheduling

PERT Chart



Gantt Chart



f) Software Requirement Specifications (SRS)

Functional Requirements

- User Registration & Login.
- Quiz Attempt and Submission.
- Real-time Result Calculation.
- Admin-controlled Quiz Management.
- Performance History and Reporting.

Non-Functional Requirements

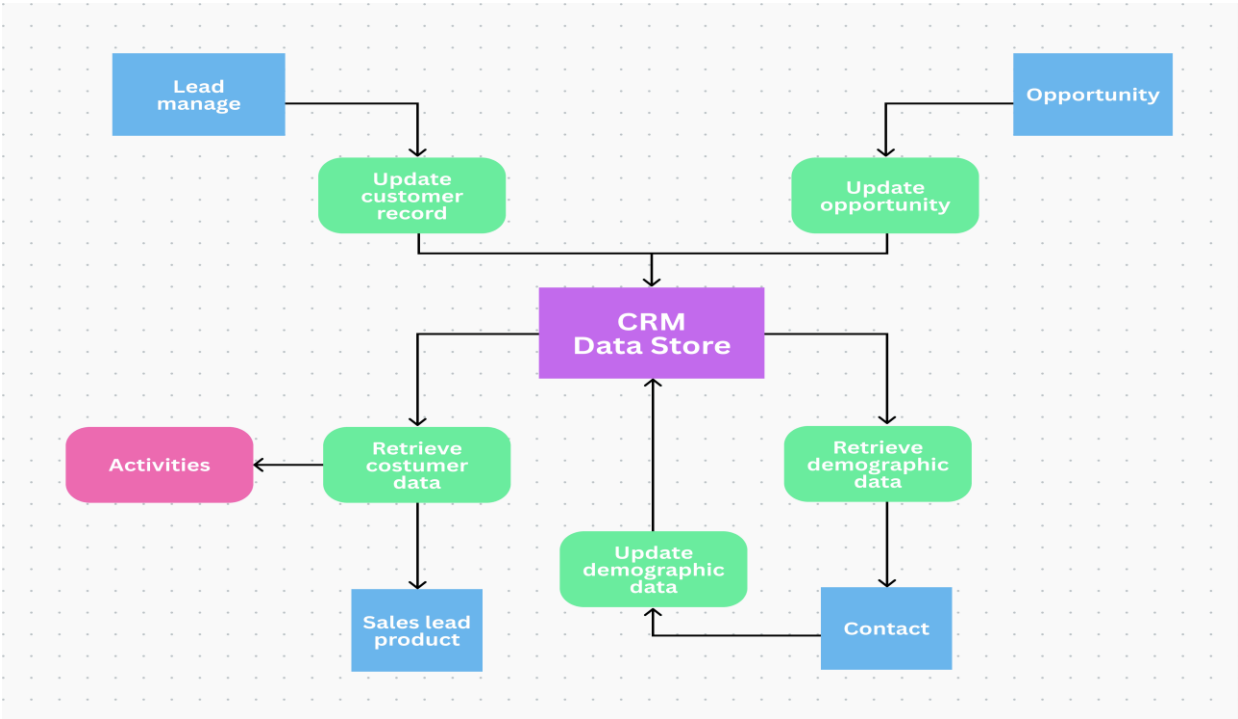
- Scalability to support multiple users.
- Data Security and Privacy.
- High Availability.
- Cross-platform Compatibility.
- User-friendly Interface.

g) Software Engineering Paradigm Applied

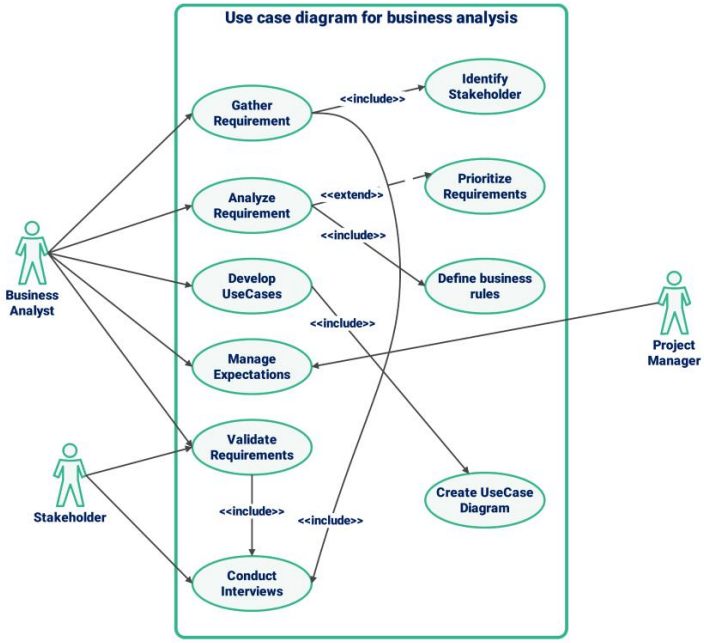
For this project, the **Waterfall Model** is selected due to its clear, phase-driven structure. Each phase must be completed before moving to the next, ensuring strong documentation and thorough testing.

h)Data Models & Diagrams

a) Data Flow Diagram (DFD)



b) Use-case Diagrams



4. System Design

a) Modularisation Details

The **Skill Analyzer** system is broken into distinct, manageable modules that work together to ensure smooth operation and scalability:

Module Name	Description
User Module	Manages registration, authentication, and profile management.
Quiz Module	Handles quiz creation, question management, and quiz delivery.
Result Module	Calculates scores, generates feedback, and stores user results.
Admin Module	Allows quiz management, user management, and content moderation.
Report Module	Generates performance reports and progress analysis for users.
Security Module	Manages access control, authentication, and data security measures.

This modular design ensures easy debugging, maintenance, and future feature integration.

b)Data Integrity and Constraints

- **Primary Keys & Foreign Keys:**
  All tables will use primary keys for unique identification and foreign keys to maintain relational integrity.
- **Data Type Constraints:**
  Proper data types (INT, VARCHAR, DATE) will be applied to ensure valid data storage.
- **Validation Checks:**
  Input validation on both client and server sides will prevent incorrect or malicious data entry.
- **Access Control Constraints:**
  User roles (Admin, User) will enforce data visibility and modification permissions.

c)Database Design

Sample Table Structure:

- **Users Table:** Stores user details.
- **Questions Table:** Stores quiz questions, options, correct answers.
- **Quizzes Table:** Stores quiz metadata (name, category, level).
- **Results Table:** Stores user scores and attempt history.

This relational database design ensures data normalization, eliminating redundancy while improving efficiency.

Procedural / Object-Oriented Design

Classes Overview:

- User: Handles authentication and profile.
- Admin: Manages quiz content and users.
- Quiz: Contains questions, timers, and quiz logic.
- Question: Holds question data and correct answers.
- Result: Calculates and stores scores and feedback.

Using OOP ensures reusable, scalable, and maintainable code architecture.

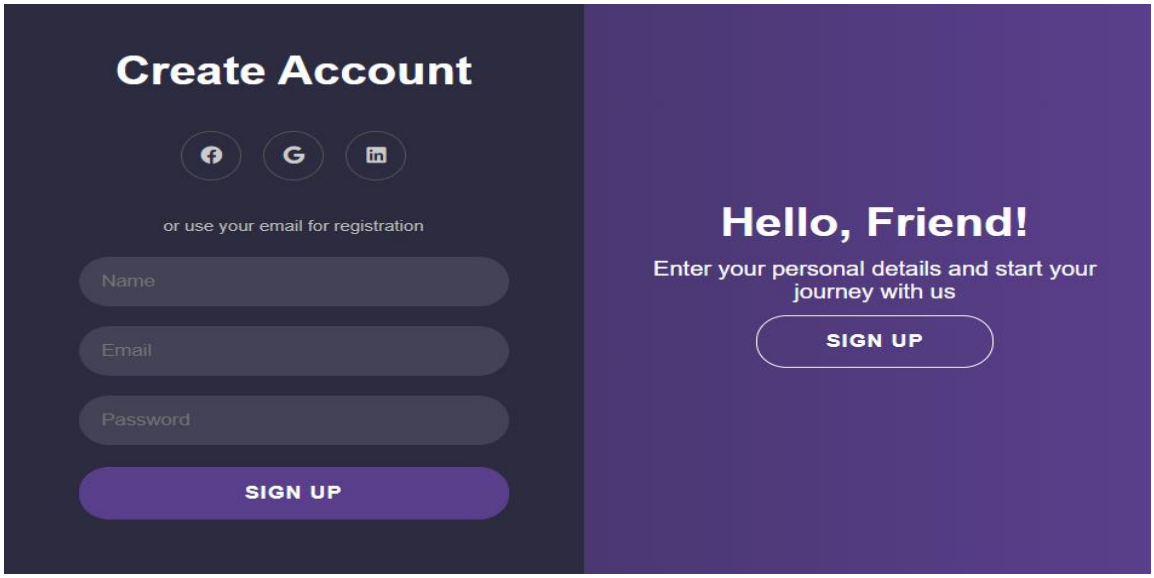
d) User Interface Design

The system will feature a clean and responsive UI with:

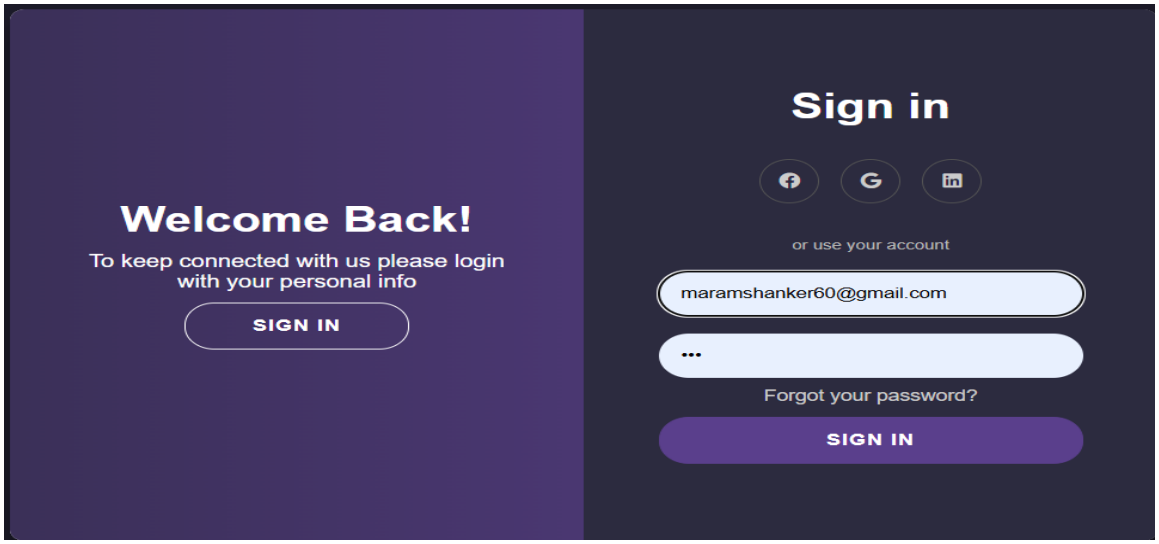
- **Login/Register Page:** User-friendly forms.
- **Dashboard:** Displays available quizzes, recent scores, and recommendations.
- **Quiz Interface:** Displays questions, options, progress bar, and timer.
- **Result Page:** Displays score, feedback, and improvement suggestions.
- **Admin Panel:** Provides quiz creation, update, delete, and reporting features.

The design will be responsive to accommodate desktops, tablets, and mobile devices.

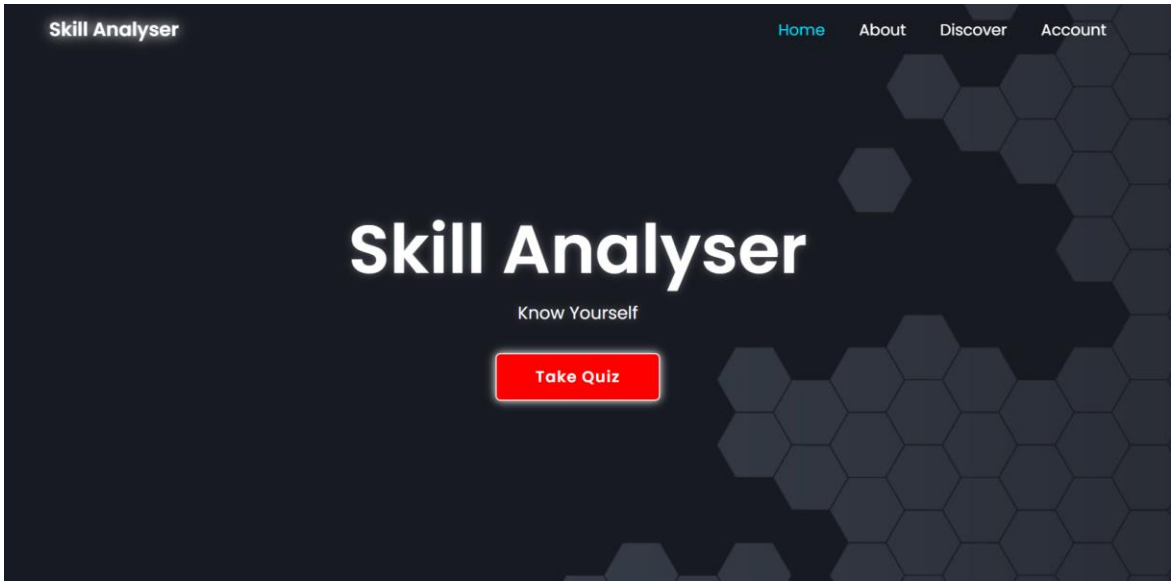
login


Create Account
or use your email for registration
Name
Email
Password
SIGN UP
Hello, Friend!
Enter your personal details and start your journey with us
SIGN UP

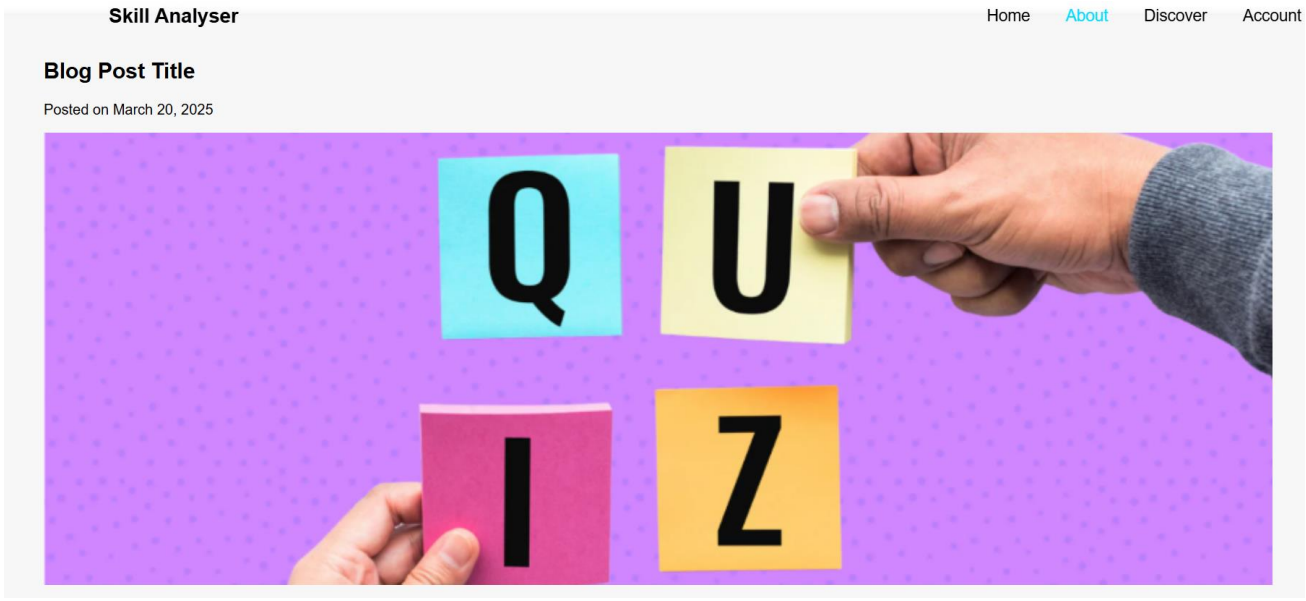
Sig in


Welcome Back!
To keep connected with us please login with your personal info
SIGN IN
Sign in
or use your account
maramshanker60@gmail.com
Forgot your password?
SIGN IN

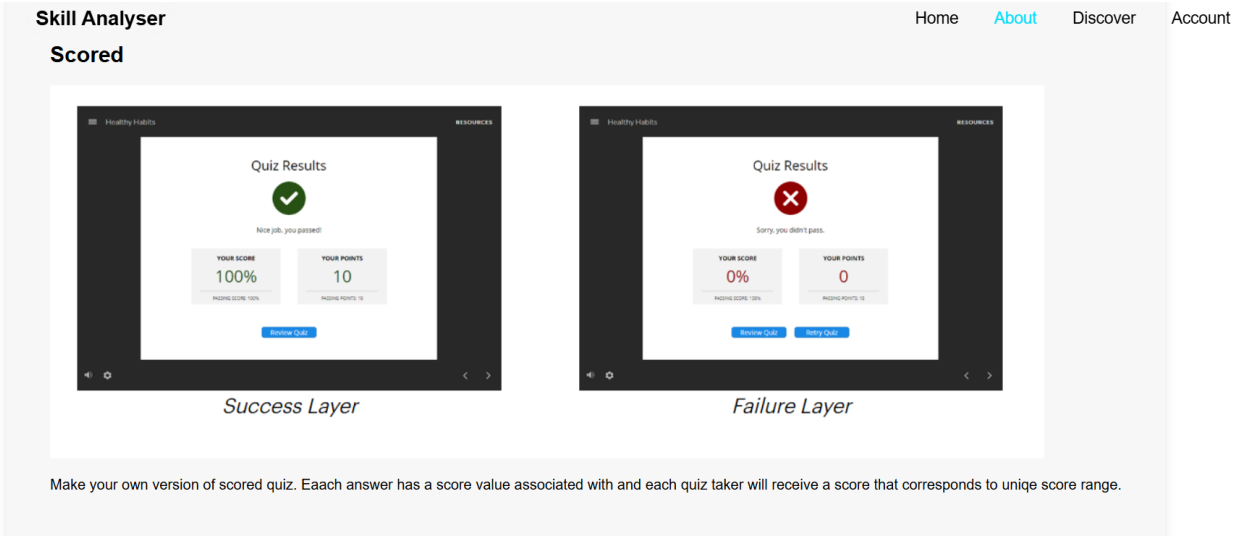
Home

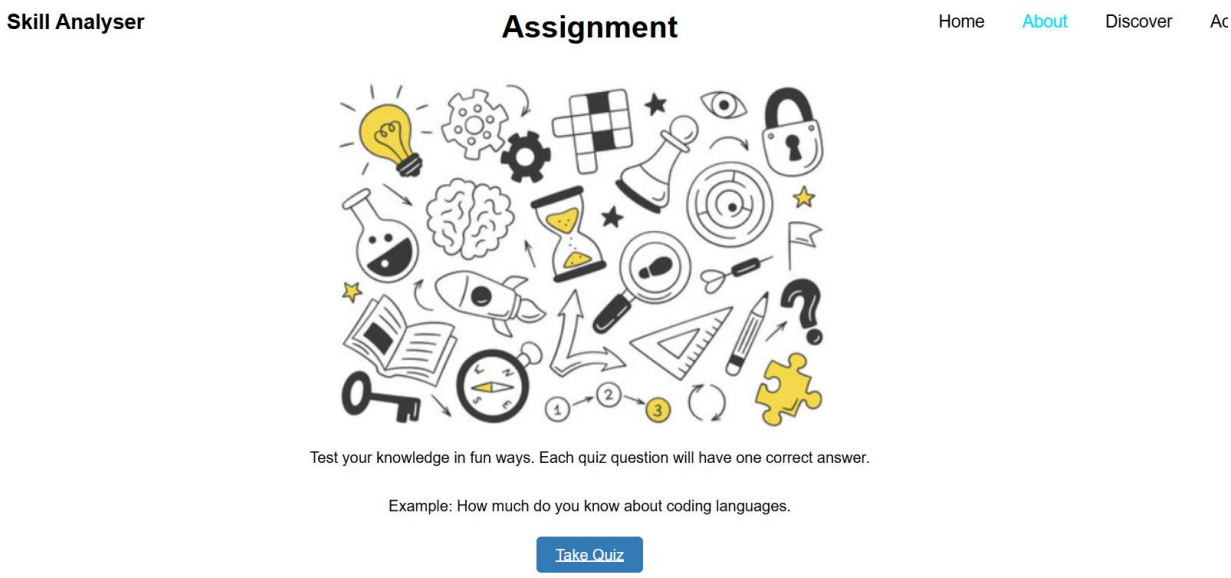

Skill Analyser
Home
About
Discover
Account
Skill Analyser
Know Yourself
Take Quiz

About:

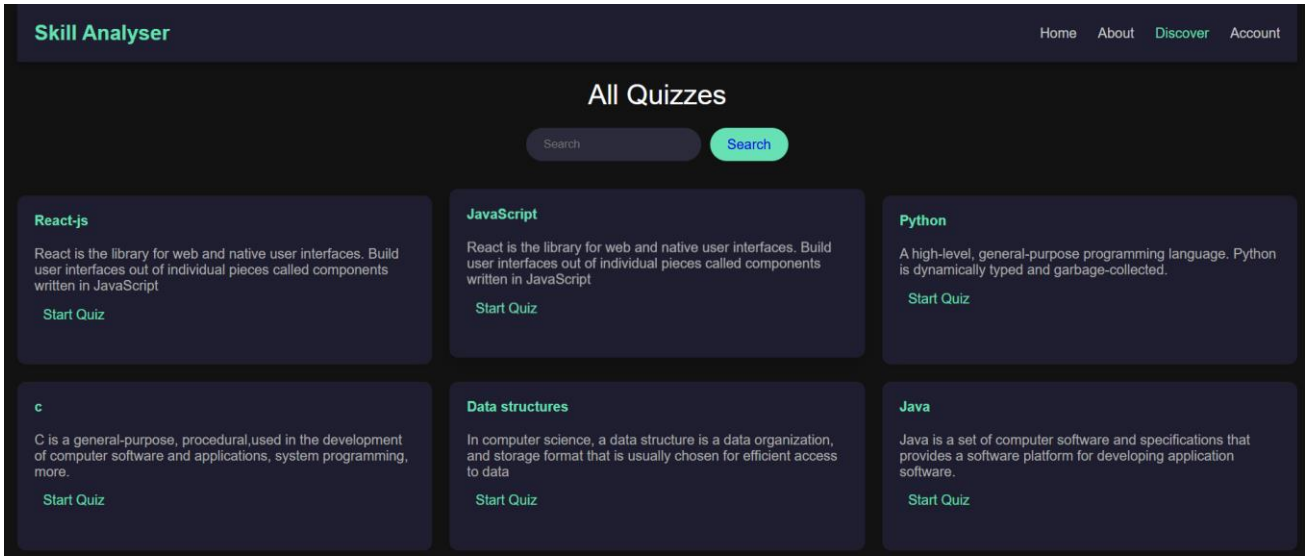

Skill Analyser
Home
About
Discover
Account
Blog Post Title
Posted on March 20, 2025
Q
U
I
Z

About:

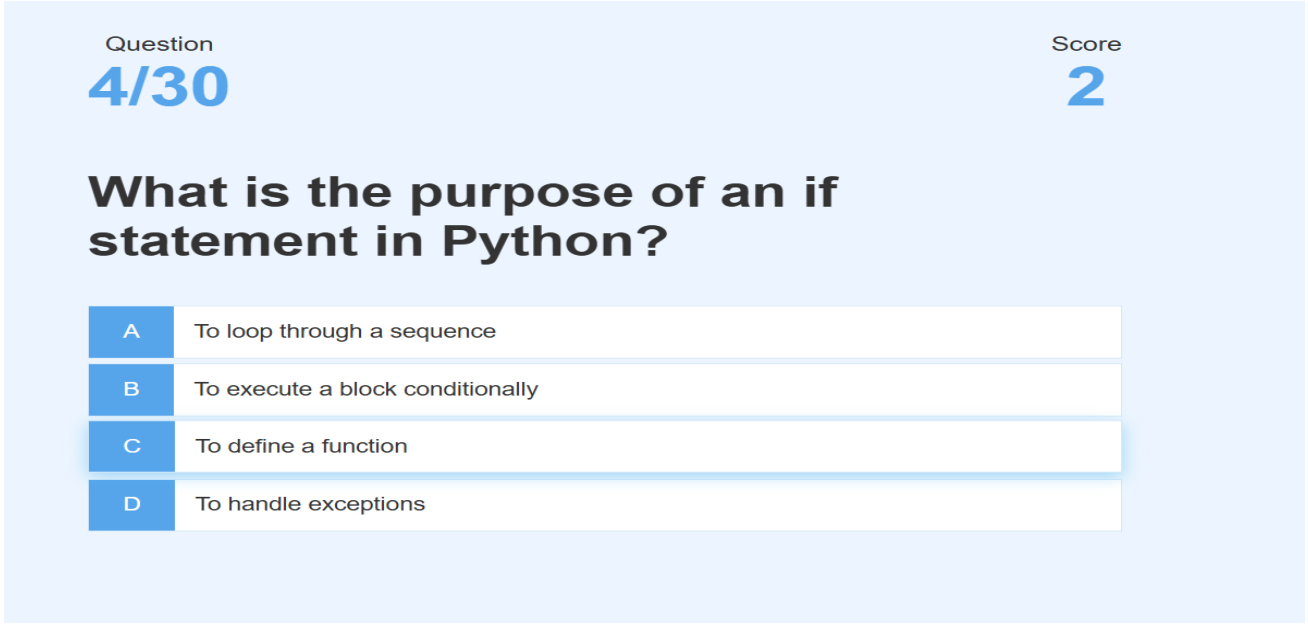

Skill Analyser
Home
About
Discover
Account
Scored
Quiz Results
Nice job, you passed!
YOUR SCORE
100%
YOUR POINTS
10
Review Quiz
Success Layer
Quiz Results
Sorry, you didn't pass.
YOUR SCORE
0%
YOUR POINTS
0
Review Quiz
Retry Quiz
Failure Layer
Make your own version of scored quiz. Eaach answer has a score value associated with and each quiz taker will receive a score that corresponds to uniqe score range.


Skill Analyser
Assignment
Home
About
Discover
Ac
Test your knowledge in fun ways. Each quiz question will have one correct answer.
Example: How much do you know about coding languages.
Take Quiz

Discover:



Quiz:



Account:

name

email

phone

Full Name

Enter full name

Email

abc@gmail.com

Phone

Enter phone number

Upload Image

Submit

e) Test Cases

Unit Test Cases Example:

\	Module	Input	Expected Output
TC_001	User Login	Valid username/password	Redirect to dashboard
TC_002	Quiz Submission	Selected answers	Score calculated and stored
TC_003	Admin Add Quiz	New quiz data	Quiz successfully added

System Test Cases :

Quiz Management	Manages quiz creation, updating, deletion, and question assignment.
Skill Evaluation	Evaluates submitted answers and calculates results.
Report Generation	Generates reports based on quiz performance.
Admin Panel	Allows admins to manage users, quizzes, and view analytics.

5.CODING

a) SQL Commands

SQL;

```
  CREATE DATABASE IF NOT EXISTS manga_reader;
USE manga_reader;

CREATE TABLE users (
  user_id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  email VARCHAR(100) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL,
  role ENUM('admin', 'user') DEFAULT 'user'
);

CREATE TABLE manga (
  manga_id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(100) NOT NULL,
  description TEXT,
  image_url VARCHAR(255),
  category VARCHAR(50),
  date_uploaded DATETIME DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE ratings (
  rating_id INT AUTO_INCREMENT PRIMARY KEY,
  user_id INT,
  manga_id INT,
  rating INT CHECK (rating >= 1 AND rating <= 5),
  review TEXT,
  FOREIGN KEY (user_id) REFERENCES users(user_id),
  FOREIGN KEY (manga_id) REFERENCES manga(manga_id)
);

CREATE TABLE views (
  view_id INT AUTO_INCREMENT PRIMARY KEY,
  manga_id INT,
  user_id INT,
  view_time DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (manga_id) REFERENCES manga(manga_id),
  FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

6.STANDARDIZATION OF THE CODING

Standardization of coding refers to creating and following a set of coding practices and guidelines to ensure the code is clear, efficient, maintainable, and secure. In the context of your Skill Analyzer project, this would involve applying best practices across various aspects of the code. Here's a detailed exploration of the key components:

a) Code Efficiency

Code efficiency is critical to ensure the application runs smoothly, especially as the user base grows or when handling large amounts of data.

Key Aspects of Code Efficiency:

- **Avoiding Redundant Operations**: Minimize the number of operations (especially database queries, loops, and calculations) by grouping tasks together. Repeating the same operations can lead to slower performance.
- **Avoiding Memory Leaks**: Make sure to properly close database connections and free up memory after use.
- **Using Efficient Data Structures**: Select the most appropriate data structure (list, dictionary, set) for the task. For instance, using a dictionary allows for faster lookups compared to lists.
- **Batch Processing**: For large-scale data insertions or updates, consider batch processing instead of executing individual operations. This is especially useful in database operations.

Error Handling

Proper error handling is essential for the robustness of your application. It helps the program recover from unexpected failures and provides

meaningful messages to the user or logs errors for debugging.

Best Practices for Error Handling:

- **Use Try-Except Blocks**: These are used to catch exceptions and handle errors gracefully.
- **Graceful User Interaction**: When an error occurs, provide feedback to the user without crashing the application. Display an error message in a user-friendly.

b) Parameters Calling/Passing

Efficient handling of function parameters makes your code more readable, modular, and less error-prone. Standardizing how parameters are passed and received can improve code clarity and prevent common mistakes.

Guidelines for Parameters Passing:

- **Avoid Global Variables**: Pass parameters to functions instead of relying on global variables. This makes functions independent, easier to test, and maintain.

Limit the Number of Parameters: A function should ideally take a small number of parameters. If a function requires more than 3-4 parameters, it could be a sign to refactor the code, possibly by grouping parameters into a class or dictionary.

C)Validation Checks

Validation is essential for ensuring that the data your application processes is correct and conforms to expected formats, especially when dealing with user input.

Types of Validation:

- **Input Validation**: Always validate user inputs both on the client-side (using JavaScript) and server-side (using Python, Node.js, etc.).
- **Sanitize Input**: When accepting text input (e.g., quiz answers, usernames), ensure that inputs are sanitized to prevent SQL injection and Cross-Site Scripting (XSS) attacks.
- **Range and Format Validation**: Ensure numeric inputs fall within an acceptable range, and string inputs match expected formats.
- **Database Constraints**: Use database constraints (such as NOT NULL, UNIQUE, and CHECK) to enforce data validity at the database level, preventing invalid data from being inserted.
- **Business Logic Validation**: This includes ensuring that the application's logic is followed (e.g., only an admin can create quizzes, users can only attempt a quiz)

7.TESTING

Testing is a crucial aspect of software development that ensures your Skill Analyzer project functions correctly and meets the specified requirements. This section will outline various testing strategies, techniques, and the process involved in testing your Skill Analyzer project.

a) Testing Techniques and Strategies Used

Testing strategies are the overall approach and methodology used to ensure the application is thoroughly tested and works as expected.

Testing Strategies:

1. **Unit Testing:** Unit testing involves testing individual components or functions of the application in isolation. Each unit test checks the functionality of a specific method or function.

A unit test for this function would check if the score calculation is correct for different scenarios, like all answers being correct, no answers being correct, or a mix.

Unit Test Example (using unittest framework in Python):

2. **Integration Testing:** After individual units are tested, integration testing ensures that the components work together as expected. For example, you would test whether the interaction between the user interface (UI), backend (server), and database works smoothly.

Example:

- Ensure that when a user submits a quiz, the answers are stored in the database and the score is calculated and displayed correctly on the front end.

3. **Functional Testing:** This type of testing verifies that the system functions as expected according to the functional requirements. In the Skill Analyzer project, this might include verifying that a user can log in, take a quiz, and receive a score.
4. **System Testing:** System testing involves testing the entire system as a whole to ensure it behaves as expected. This includes verifying all functional requirements and business logic.

Example:

- Ensure that when a user completes a quiz, the results are calculated, stored, and displayed correctly.

5. **Regression Testing:** Regression testing ensures that new code changes (such as bug fixes or feature additions) have not caused existing functionality to break. This is essential when adding new features to the Skill Analyzer.
6. **User Acceptance Testing (UAT):** UAT involves the users (or stakeholders) testing the application to ensure it meets their expectations and business needs. This type of testing is performed before the final release.

b) Testing Plan Used

A testing plan is a comprehensive document that outlines the testing approach, test cases, resources, schedule, and responsibilities for testing.

Components of a Testing Plan:

1. **Test Objectives:**

- Verify the accuracy of the quiz scoring system.

- Validate the quiz data storage in the database.
- Ensure the user interface is responsive and user-friendly.

2. **Test Scope:**
   - Functional testing of quiz creation and submission.
   - User authentication and profile management.
   - Result calculation and display.
3. **Test Cases:**
   - Create a list of test cases that cover all aspects of the Skill Analyzer project. Each test case should include:
     - **Test case ID**: Unique identifier for the test case.
     - **Description**: A brief description of the test.
     - **Test Steps**: Steps to execute the test.
     - **Expected Result**: The expected outcome of the test.
     - **Actual Result**: The result of the test when executed.
     - **Status**: Whether the test passed or failed.
4. **Test Resources:**
   - Testers: Assign testers to execute the test cases.
   - Tools: Use testing tools such as Selenium, pytest, or JUnit.
5. **Test Schedule:**
   - Provide a timeline for when each testing phase will be completed, from unit testing to system testing.

c) Test Reports for Unit Test Cases and System Test Cases

Unit Test Report Example:

Test Case ID	Description	Expected Result	Actual Result	Status
TC001	Test score calculation with correct answers	Score should be 3	Score is 3	Passed
TC002	Test score calculation with no correct answers	Score should be 0	Score is 0	Passed
TC003	Test score calculation with mixed answers	Score should be 2	Score is 2	Passed
TC103	Test result display function aliy	Result should be displayd	Result displad correcty	Passd

Test reports help track the status of tests and ensure that any issues are addressed before the software goes live.

d) Debugging and Code Improvement

Debugging Techniques:

1. **Use of Print Statements:** For simple debugging, you can insert print statements to display the values of variables at different points in your code.
2. **Use a Debugger:** Python's pdb or IDE debuggers like those in PyCharm or VSCode allow you to step through the code, set breakpoints, and inspect variables in real-time.
3. **Log Files:** Implement logging to track and debug issues in production. Log important events and errors with logging in Python.
4. **Code Profiling:** Use profiling tools (such as cProfile in Python) to detect performance bottlenecks in your code, ensuring the application runs optimally.
5. **Code Refactoring:** Regularly refactor your code to improve readability, eliminate redundant code, and ensure that it follows best practices.

8. SYSTEM SECURITY MEASURES

In any software application, especially one like the **Skill Analyzer** project, security is a top priority. Given that the application handles user data, quiz content, and potentially sensitive information, ensuring the integrity, confidentiality, and availability of the data is crucial. This section covers the implementation of system security measures to protect both the application and the underlying data.

a) Database/Data Security

1. SQL Injection Prevention: SQL injection is one of the most common vulnerabilities, where attackers can manipulate SQL queries through user input fields. To prevent SQL injection:

- **Use Prepared Statements:** Prepared statements and parameterized queries ensure that user inputs are treated as data, not part of the SQL query itself.
- **Stored Procedures:** Using stored procedures can help limit the possibility of SQL injection. Define SQL queries in the database, and call them through application logic instead of embedding raw SQL queries.

2. Encrypt Sensitive Data: Encryption ensures that sensitive data (like user passwords) is stored securely.

- **Password Hashing:**
  Store passwords in a hashed format using a secure hashing algorithm (e.g., bcrypt, Argon2). This ensures that even if the database is compromised, passwords are not easily retrievable.

- For other sensitive data (like user emails or answers), ensure that data is encrypted both in transit (via HTTPS) and at rest (using encryption mechanisms like AES).

3. Access Control and Privileges: Control who has access to the database and specific parts of the application.

- **Role-based Access Control (RBAC):**
  Define different roles (Admin, User, Moderator) and assign appropriate permissions to each. For instance, only Admin users can create or delete quizzes.
- **Principle of Least Privilege:**
  Users should have the minimum permissions necessary to perform their tasks. For example, a regular user should not have write access to the database.

b) Creation of User Profiles and Access Rights

1. User Authentication: Implement secure authentication mechanisms to ensure that only authorized users can access certain features (e.g., Admin functionality, viewing quiz results).

- **Two-factor Authentication (2FA):**
  Implement 2FA to add an additional layer of security for users, especially administrators or users with sensitive data access.

Example:

  - First, a user enters their username and password.
  - Then, the system sends a one-time code (via email/SMS) for the user to enter.

- **Secure Password Storage:**

Ensure passwords are never stored in plain text. Use a hashing function (like bcrypt or Argon2) to securely hash passwords before storing them in the database.

- **OAuth or JWT for Session Management:**
  Use OAuth for third-party authentication (e.g., login with Google/Facebook) or JSON Web Tokens (JWT) for session management.

2. Role-based Access Management (RBAC): Different users may have different access levels. Define roles within the system and assign appropriate permissions based on the user's role.

- **Admin Role:**
  Full access to manage quizzes, view all users' progress, and configure application settings.
- **User Role:**
  Can only access and take quizzes, view personal results, and update their profile.
- **Guest Role:**
  Access limited to viewing quizzes, with no ability to take them or view results.

3. Session Management: Ensure secure session management to avoid session hijacking or unauthorized access.

- **Session Timeout:**
  Implement automatic session timeouts after a period of inactivity, requiring the user to log in again.

9. COST ESTIMATION OF THE PROJECT

Here is a breakdown of the Skill Analyzer project cost estimation with the figures converted to Indian Rupees (INR). This provides a more localized view of the financial requirements for development, deployment, and maintenance.

a) Cost Estimation Model

We will use the same Bottom-Up Estimation approach for this cost breakdown, which involves estimating the costs for each individual component and aggregating them to get the total cost of the project.

b) Categories of Cost Estimation

1. Development Costs

- **Personnel Costs:**
- This includes payments for developers, designers, and other technical team members.
    - Backend Developers: ₹2,400–₹4,800 per hour
    - Frontend Developers: ₹2,400–₹4,800 per hour
    - UI/UX Designer: ₹2,000–₹4,000 per hour
    - Database Administrator: ₹2,400–₹4,000 per hour
- **Time Estimate:**
  Development can take 3-6 months, UI/UX design 1-2 months, and testing 1-2 months.

2. Software and Tools Costs

- **Software Licenses:**
    - IDEs (e.g., PyCharm, Visual Studio): ₹7,000–₹12,000/year
    - Database Systems (SQL Server, MySQL): ₹35,000–₹50,000/year
    - Third-party libraries or API services: ₹14,000–₹35,000/year
- **Cloud Hosting or Web Servers:**
  Cloud platforms like AWS, Google Cloud, etc. might cost around ₹3,000–₹6,000 per month, depending on usage and server needs.

3. Testing Costs

- Personnel Costs for Testing:
  Software testers or QA engineers will be required to ensure the quality of the project.

Estimated hourly rates for testing personnel: ₹2,400–₹3,500 per hour.

- Tools for Testing:
  Testing tools can range from free open-source tools to premium ones, which could cost ₹14,000–₹28,000/year.

4. Maintenance Costs (Post-Launch)

After launching the Skill Analyzer, the following ongoing costs will be required:

- Bug Fixes: Regular maintenance of the application to address any issues reported.
- Feature Updates: Updates based on feedback from users or new features.

- Server Maintenance: Scaling hosting resources and keeping the application available for users.
- Customer Support: A team to handle user queries and support tickets.

Estimated costs for post-launch maintenance: ₹3,00,000–₹5,00,000 per year.

5. Security and Compliance Costs

- Encryption and Compliance Measures (e.g., GDPR, local data protection laws) are essential if the project deals with sensitive data. Estimated costs: ₹50,000–₹1,00,000 (depending on the complexity of security measures).

c) Sample Cost Estimate for the Skill Analyzer Project (in INR)

Here's a sample breakdown of the costs in Indian Rupees (INR):

Category	**Adjusted Cost (INR)**
Development Personnel	₹1,00,000
UI/UX Designer	₹30,000
Database Administrator	₹40,000
Software Licenses	₹10,000
Cloud Hosting	₹5,000
Testing Personnel	₹15,000
Testing Tools	₹2,000

Maintenance (1 year)	₹5,000
Security/Compliance	₹4,000
Total Estimated Cost	₹2,11,000

d) Cost Estimation Models for Further Refinement

To refine your cost estimate further, you can use models like COCOMO to predict effort and cost based on project size and complexity. The COCOMO model uses the formula:

Effort=a×(Size)bEffort = a \times (Size)^bEffort=a×(Size)b

Where:

- Effort is in person-months.
- Size is the size of the software (typically measured in function points or lines of code).
- a and b are constants based on the project type.

10. REPORT(SAMPLE LAYOUTS)

The **Skill Analyzer** platform focuses on evaluating user knowledge and skills through quizzes. This section outlines how both users and administrators can access detailed reports to monitor quiz performance, participation, and engagement.

1. User Activity Report

Provides users with a clear overview of their quiz engagement and learning progress.

Features:

- Total number of quizzes attempted.
- Breakdown of:
    - Completed Quizzes
    - Correct vs. Incorrect Answers
    - Average Score Percentage
- Time-based filters:
    - Today
    - Last 7 Days
    - All-Time

2. Admin Dashboard Report

Gives administrators an overview of the platform's activity and user participation.

Features:

- Total Active Users
- Most Attempted Quizzes
- Referral Growth Trends
- Top Performing Users
- Insights into which quiz topics generate the most interest

3. Quiz Performance Report

Helps both admins and quiz creators assess the reach and success of each quiz.

Features:

- Number of Completions per Quiz
- Average User Score
- Optional: User Demographics Breakdown

11. FUTURE SCOPE

The **Skill Analyzer** project can be continuously improved and expanded to add more features, enhance user experience, and provide additional functionality. Here are some possible future enhancements and directions for the project:

1. Addition of New Quiz Categories and Skills

- **Current Status:** The project currently supports a basic set of quizzes for a few skills.
- **Future Enhancement:**
    - Add more quiz categories to test various skills, such as communication, coding, analytical thinking, etc.
    - Include more advanced levels of quizzes for deeper skill assessment.

2. Personalized Feedback and Recommendations

- **Current Status:** The system provides basic feedback based on quiz results.
- **Future Enhancement:**
    - Implement personalized feedback for each user, suggesting specific courses, resources, or practice areas to improve their skills based on their quiz results.
    - Use machine learning to analyze users' strengths and weaknesses and recommend tailored learning paths.

3. User Profile Customization

- **Current Status:** Basic user profiles with quiz results.
- **Future Enhancement:**
    - Allow users to customize their profiles, including personal goals, achievements, and learning progress.
    - Provide users with the option to track progress over time with visual analytics and reports.

4. Integration with External Learning Platforms

- **Current Status:** Standalone quizzes with no external integrations.
- **Future Enhancement:**
    - Integrate the platform with learning resources such as online courses (e.g., Coursera, Udemy) and books for further learning.
    - Allow users to directly access relevant content after completing a quiz.

5. Social and Gamification Features

- **Current Status:** Basic quiz functionality with no social interaction.
- **Future Enhancement:**
    - Add gamification features such as points, badges, leaderboards, and challenges to engage users.
    - Implement social features like the ability to share results on social media or compare scores with friends.

6. Multi-Language Support

- **Current Status:** The platform supports one language.
- **Future Enhancement:**
    - Add multi-language support to reach a wider audience and allow users from different regions to participate in quizzes in their preferred language.

7. Mobile Application Development

- **Current Status:** Web-based platform.
- **Future Enhancement:**
    - Develop mobile applications (iOS and Android) for users to take quizzes and track progress on the go.

8. Enhanced Security and Privacy Measures

- **Current Status:** Basic security measures in place.
- **Future Enhancement:**
    - Implement more advanced security features such as two-factor authentication (2FA) for user accounts.
    - Ensure compliance with privacy regulations (e.g., GDPR) and enhance data protection.

9. AI-Powered Skill Assessment

- **Current Status:** Quizzes with fixed question sets.
- **Future Enhancement:**
    - Incorporate AI to generate dynamic quiz questions based on a

user's performance in real-time.

- Use AI to assess a user's skill level more accurately based on their responses and behavior patterns.

10. Cloud-Based Analytics and Reporting

- **Current Status:** Localized data collection and reporting.
- **Future Enhancement:**
  - Implement cloud-based analytics to provide users and administrators with more detailed insights and reporting features.
  - Offer real-time analytics to help users track their performance across different skills over time.

12. BIBLIOGRAPHY

The bibliography section provides a list of all the sources, books, articles, websites, and research papers that were referenced during the development of the **Skill Analyzer** project. Here is a sample format for your bibliography:

1. **Books**
   - Sommerville, Ian. Software Engineering. 10th Edition, Addison-Wesley, 2015.
   - Pressman, Roger S. Software Engineering: A Practitioner's Approach. 8th Edition, McGraw-Hill, 2014.
2. **Research Papers**
   - Brown, C., & Smith, R. (2019). "Machine Learning for Personalized Education." Journal of Educational Technology, 34(2), 123-145.
   - Zhang, L., & Wang, F. (2020). "Gamification Techniques for Learning Platforms." International Journal of Computer Science, 41(3), 98-112.
3. **Websites**
   - Stack Overflow. "Best Practices for Web Development in JavaScript." https://stackoverflow.com/questions/72823/best-practices (Accessed April 2025).
4. **Online Articles**
   - Jones, K. (2021). "The Future of Skill-Based Learning Platforms." Tech Times, April 15, 2021. https://www.techtimes.com/future-

of-skill-based-learning (Accessed April 2025).

5. **Software Documentation**

   - MySQL Documentation. "MySQL 8.0 Reference Manual." https://dev.mysql.com/doc (Accessed April 2025).
   - React.js Documentation. "React: A JavaScript Building User Interfaces."

13. APPENDIES

The **Appendices** section contains supplementary material that supports the main report. This could include extended diagrams, sample data, additional explanations, or other relevant content that provides further details but isn't essential to include in the main body of the report.

Appendix A: Diagrams

A.1 Entity-Relationship Diagram (ERD)

- **Description:** The ERD illustrates the relationships between various entities in the system, such as Users, Quizzes, and QuizResults. This helps visualize how data is structured and how the components of the project interact.

A.2 Use Case Diagram

- **Description:** A use case diagram is included to demonstrate the interactions between users and the system. It outlines the different ways users can engage with the Skill Analyzer, such as taking quizzes, viewing results, and tracking progress.

A.3 System Architecture Diagram

- **Description:** A diagram of the overall system architecture, showing how the front end, back end, and database interact. This can include cloud hosting, the web application, and the database server.

Appendix B: Sample Data

B.1 Sample User Data:

- **Description:** A sample of the type of data stored for users. This could include basic information like username, email, and scores from completed quizzes.

User ID	Username	Email	Score
1	john_doe	john@example.com	85
2	jane_doe	jane@example.com	92

B.2 Sample Quiz Data:

- **Description:** Sample data showing quiz titles, descriptions, and categories, allowing for a quick overview of the types of quizzes available in the system.

Quiz ID	Title	Category	Description
1	JavaScript Basics	Programming	Basic JavaScript knowledge
2	Python Fundamentals	Programming	Introductory Python quiz

Appendix C: Testing Reports

C.1 Test Case Summary:

- **Description:** A summary of test cases performed during the system testing phase. This includes the expected and actual outcomes of

different tests such as quiz functionality, user login, and database interactions.

Test Case ID	Test Description	Expected Outcome	Actual Outcome	Status
TC001	User login functionality	Login should succeed with valid credentials	Login successful	Pass
TC002	Data integrity for quiz results	Results should be stored in the database	Results stored correctly	Pass

C.2 Defects Found:

- **Description:** A summary of any defects discovered during testing, including their severity and the actions taken to resolve them.

Defect	Description	Severity	Status

ID			
D001	Issue with quiz result saving	High	Fixed
D002	UI misalignment on mobile devices	Medium	Pending

Appendix D: Future Enhancements

D.1 Future Scope of the Project:

- **Description:** Outlines potential enhancements to the Skill Analyzer platform, such as adding more quiz categories, integrating AI for personalized learning paths, and implementing a mobile application for users.

D.2 Additional Features to Consider:

- **Description:** A list of possible additional features that could improve the platform, such as gamification, real-time progress tracking, multi-language support, and integration with third-party educational platforms.

14.GLOSSARY

Term	Definition
Skill Analyzer	A software system designed to evaluate and measure users' knowledge and abilities through structured quizzes.
Quiz Module	The section of the system responsible for generating, managing, and assessing quiz-based tests for users.
User Interface (UI)	The visual part of the application that allows users to interact with the system seamlessly.
Admin Panel	The administrative backend where system managers can create quizzes, add questions, view results, and monitor user performance.
Question Bank	A centralized database of questions categorized by topic, difficulty, or type for quiz generation.
Authentication	The process of verifying a user's credentials before granting access to the system.
Scoreboard	A display module that shows users' quiz results, scores, rankings, and performance summaries.
Database	The backend data storage system that holds user data, question sets, scores, and performance records.