

SkillSprint – FastAPI Based Learning & Practice Platform

A Project-I Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &
ENGINEERING**

BY

Bhavya Hasnani EN22CS301267

Ayushi Upadhyay EN22CS301258

Under the Guidance of

Dr. Sachin Solanki



Department of Computer Science & Engineering

Faculty of Engineering

MEDICAPS UNIVERSITY, INDORE- 453331

SkillSprint – FastAPI Based Learning & Practice Platform

A Project-I Report

Submitted in partial fulfillment of requirement of the

Degree of

**BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &
ENGINEERING**

BY

Bhavya Hasnani EN22CS301267

Ayushi Upadhyay EN22CS301258

Under the Guidance of

Dr. Sachin Solanki



**Department of Computer Science & Engineering
Faculty of Engineering
MEDICAPS UNIVERSITY, INDORE- 453331**

Report Approval

The project **“SkillSprint – FastAPI Based Learning & Practice Platform”** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approve any statement made, opinion expressed, or conclusion drawn there in; but approve the “Project Report” only for the purpose for which it has been submitted.

Internal Examiner

Name:

Designation

Affiliation

External Examiner

Name:

Designation

Affiliation

Declaration

We hereby declare that the project entitled “**SkillSprint – FastAPI Based Learning & Practice Platform**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology in ‘Computer Science and Engineering’ completed under the supervision of **Dr. Sachin Solanki**, Faculty of Engineering, Medicaps University Indore is an authentic work.

Further, we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Bhavya Hasnani EN22CS301267

Ayushi Upadhyay EN22CS301258

Certificate

I, **Dr. Sachin Solanki** certify that the project entitled “**SkillSprint – FastAPI Based Learning & Practice Platform**” submitted in partial fulfillment for the award of the degree of Bachelor of Technology by “**Bhavya Hasnani and Ayushi Upadhyay**” is the record carried out by them under our guidance and that the work has not formed the basis of award of any other degree elsewhere.

Dr. Sachin Solanki

Computer Science & Engineering

Medicaps University, Indore

Dr. Kailash Chandra Bandhu

Head of the Department

Computer Science & Engineering

Medicaps University, Indore

Acknowledgements

We would like to express my deepest gratitude to the Honorable Chancellor, **Shri R C Mittal**, who has provided us with every facility to successfully carry out this project, and our profound indebtedness to **Prof. (Dr.) D. K. Patnaik**, Vice Chancellor, Mediacaps University, whose unfailing support and enthusiasm has always boosted up my morale. We also thank **Prof. (Dr.) Ratnesh Litoriya**, Associate Dean, Faculty of Engineering, Mediacaps University, for giving us a chance to work on this project. We would also like to thank our Head of the Department **Dr. Kailash Chandra Bandhu** for his continuous encouragement for the betterment of the project.

I would also like to express my gratitude to our internal guides **Dr. Sachin Solanki** who has guide us in every stage while working on this project. It is their help and support, due to which we became able to complete the design and technical report.

Without their support this report would not have been possible.

Bhavya Hasnani &

Ayushi Upadhyay

B.Tech. IV Year (E)

Department of Computer Science & Engineering

Faculty of Engineering

Mediacaps University, Indore

Abstract

The rapid growth of digital learning platforms has increased the demand for scalable, secure, and interactive systems that support continuous skill development. *SkillSprint* is a FastAPI-based learning and practice platform designed to streamline technical education through structured content delivery, coding practice modules, assessments, personalized tracking, and machine learning integration.

This project aims to develop a high-performance backend system using **FastAPI**, a modern Python framework known for its asynchronous capabilities, auto-generated API documentation, and superior performance compared to traditional frameworks. The platform incorporates essential features such as secure user registration, JWT-based authentication, role-based access control, problem-solving modules, and structured learning paths.

The backend communicates with a **PostgreSQL** database through SQLAlchemy ORM, ensuring data consistency, transactional reliability, and optimized query execution. The architecture follows modular design principles, enabling easy scalability and future enhancement. The system also adheres to RESTful principles to ensure compatibility with web and mobile frontends.

SkillSprint emphasizes learner engagement through practical exercises, automated evaluation, and progress visualization. The platform also supports the integration of artificial intelligence components such as recommendation engines, automated code feedback systems, and real-time skill assessment models.

This report documents the complete lifecycle of SkillSprint, including requirement analysis, architectural design, UML diagrams, ER modeling, implementation details, testing methodologies, screenshots, results, and evaluation. Extensive diagrams such as DFDs, Activity Diagrams, Flowcharts, Class Diagrams, ER Diagrams, and Sequence Diagrams are included to visually explain the system.

Overall, SkillSprint demonstrates how modern web technologies, FastAPI's asynchronous capabilities, and robust database management can be combined to create a scalable learning platform aligned with current industry requirements.

Keywords: FastAPI, PostgreSQL, SQLAlchemy, JWT Authentication, REST API, Learning Platform, Machine Learning Integration, SkillSprint, Software Architecture.

Table of Contents

Content	Sub-Content	Page No.
Report Approval		i
Declaration		ii
Certificate		iii
Acknowledgement		iv
Abstract		v
Table of Contents		vi–vii
List of figures		viii
List of tables		ix
Abbreviations		x
Notations & Symbols		xi
Chapter 1	Introduction	1–10
	1.1 Introduction	1
	1.2 Literature Review	2
	1.3 Objectives	3–4
	1.4 Significance	5–6
	1.5 Research Design	7–8
	1.6 Source of Data	9–10
Chapter 2	Requirement Specifications	11–18
	2.1 User Characteristics	12
	2.2 Functional Requirements	13–14
	2.3 Dependencies	15
	2.4 Performance Requirements	16
	2.5 Hardware Requirements	17
	2.6 Constraints & Assumptions	18
Chapter 3	Design	19–27
	3.1 Algorithm	20

	3.2 Function Oriented Design for procedural approach	21
	3.3 System Design	21
	3.3.1 Data Flow Diagrams (Level 0, Level 1)	22
	3.3.2 Activity Diagram	23
	3.3.3 Flow Chart	24
	3.3.4 Class Diagram	25
	3.3.5 ER Diagram	26
	3.3.6 Sequence Diagram	27
Chapter 4	Implementation, Testing, and Maintenance	28–30
	4.1 Intro to Languages, IDEs, Tools & Technologies	28
	4.2 Testing Techniques & Test Plans	29
	4.3 Installation Instructions	29
	4.4 End User Instructions	30
Chapter 5	Results and Discussions	31–38
	5.1 Description of Modules	31–32
	5.2 Snapshots of System	33–38
Chapter 6	Summary and Conclusions	39–41
Chapter 7	Future Scope	42–43
Appendix		44
Bibliography		45

List of Figures

Figure No.	Title
Figure 1	Figure that shows the dashboard page
Figure 2	Screenshot of dashboard.
Figure 3	Screenshot of progress of the user.
Figure 4	Screenshot of quiz.
Figure 5	Screenshot of result and performance.
Figure 6	Screenshot of performance of each topic.
Figure 7	Screenshot of the topic which you want to improve.
Figure 8	Screenshot of SignUp Page.
Figure 9	Screenshot of User's Profile page.
Figure 10	Screenshot of login page.

List of Tables

Serial no.	Table name and description	Page number
1.	Hardware Requirements	17
2.	Procedure oriented approach	21

Abbreviations

Abbreviation	Full Form
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
JWT	JSON Web Token
DBMS	Database Management System
ORM	Object Relational Mapping
REST	Representational State Transfer
UI	User Interface
UX	User Experience
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment

Notations & Symbols

Symbol	Meaning
α	Learning Rate
λ	Regularization Parameter
t	TimeStamp
U	User Entity
(σ)	Standard deviation
R	Request Object
μ	Mean
PK	Primary Key
FK	Foreign Key
\rightarrow	Data Flow

Chapter-1 Introduction

1.1 Introduction

The rapid evolution of technology and the increasing demand for industry-ready graduates have transformed the landscape of technical education. Programming skills, software development knowledge, and hands-on project experience have become mandatory for students seeking to excel in modern software engineering roles. In response to these needs, academic institutions and ed-tech platforms are shifting toward practical, interactive, application-oriented learning systems.

However, traditional platforms often face limitations such as outdated interfaces, lack of personalization, weak backend architectures, limited scalability, and absence of real-time analytics. To address these concerns, a modern learning platform must be efficient, scalable, secure, asynchronous, and capable of supporting deep integration with machine learning, user tracking, evaluations, and recommendations.

SkillSprint has been conceptualized as a robust, high-performance backend system built using **FastAPI**, one of the fastest-growing Python frameworks for building modern, asynchronous, production-grade RESTful APIs. The platform provides structured learning modules, coding challenges, user authentication, database-backed progress tracking, and future support for ML-based personalized recommendations.

SkillSprint emphasizes:

- ✓ High-speed asynchronous backend processing
- ✓ Secure and modern authentication using JWT
- ✓ Database-driven content and user management
- ✓ Scalable architecture for future features
- ✓ Smooth communication with frontend frameworks
- ✓ API documentation automatically generated by FastAPI

The goal of this project is to demonstrate how modern API technology can be employed to build a large-scale learning platform that supports both academic and industrial-level skill development.

By combining database engineering, API development, software design, diagrams, testing techniques, and modern software architecture principles, SkillSprint serves as a real-world implementation of full-stack backend engineering.

1.2 Literature Review

Learning Management Systems (LMS) and skill development platforms have been widely studied and deployed over the past two decades. Early systems such as Moodle, Blackboard, and Canvas focused primarily on content distribution, grading, and student tracking. However, they lacked real-time skill-based evaluation and interactive learning features.

Recent research highlights several challenges in traditional e-learning systems:

- **Limited backend scalability**
- **Synchronous request handling leading to slow performance**
- **Weak support for authenticated API usage**
- **Outdated or rigid architectures**
- **Lack of microservice compatibility**

FastAPI in Academic and Industry Research

In 2018, *Sebastián Ramírez* introduced FastAPI, a high-performance Python framework leveraging asynchronous programming through ASGI (Asynchronous Server Gateway Interface). Research papers have reported:

- Up to **400% faster performance** compared to Flask
- Built-in validation using **Pydantic Models**
- Automatic documentation using **Swagger UI & Redoc**
- Native support for OAuth2, JWT, and Authorization

Recent studies also show that FastAPI is widely adopted in:

- Machine learning model deployment
- Microservices architecture
- Scalable cloud-native applications
- API-based backend systems

Learning Platforms Research

Modern learning platforms like Coursera, LeetCode, and HackerRank rely on:

- API-driven backend systems
- Real-time code evaluation engines
- User progress visualization
- ML-driven recommendation systems

Research indicates that modular backend architectures significantly improve:

- System reliability
- Content delivery

- Assessment mechanisms
- User engagement
- Multi-device compatibility

Gap Identified

Most platforms do not provide:

- A customizable backend for institutions
- FastAPI-based microservice architecture
- Real-time analytics-readable APIs
- Expandable open-source structure for further development

Conclusion of Literature Review

The literature strongly suggests the need for:

- A high-performance, modular, secure backend
- Asynchronous API support
- Strong database integration
- Scalability for ML and analytics

SkillSprint fills this research gap by implementing all these features using modern programming principles.

1.3 Objectives

The primary objective of **SkillSprint** is to design and develop a high-performance, secure, scalable backend platform utilizing FastAPI for a modern learning and coding practice system.

Specific objectives include:

1. Technical Objectives

- To design a modular and scalable backend using FastAPI.
- To implement secure user registration, login, and authentication using JWT.
- To manage structured learning modules through a PostgreSQL database.
- To build RESTful APIs for quizzes, challenges, progress tracking, and admin functionalities.
- To ensure real-time communication capability through asynchronous programming.

2. Educational Objectives

- To provide structured coding paths and conceptual learning modules.
- To allow users to practice coding problems and track progress over time.
- To support academic usage (teachers, institutions) and personal learning.

3. System-Level Objectives

- To design architecture based on API-driven, service-oriented principles.
- To ensure high reliability, speed, and low latency.
- To enable future integration of:
 - AI-driven recommendations
 - Automated code evaluation
 - Interactive features such as forums and submissions

1.4 Significance

SkillSprint holds significant value in both academic and real-world technological contexts.

1. For Students

- Provides structured learning of core subjects
- Helps in preparing for coding interviews
- Tracks progress with real-time analytics
- Offers practical exposure to APIs, databases, and web engineering

2. For Institutions

- Can be integrated as an internal LMS
- Useful for conducting assessments
- Automatic progress tracking reduces manual work
- Facilitates research-based teaching methodologies

3. For Developers

- Serves as an example project of modern API development
- Provides a base for deploying microservices
- Demonstrates usage of asynchronous programming
- Shows integration of SQLAlchemy ORM with FastAPI

4. For Community and Industry

- Demonstrates scalable backend architecture
- Can evolve into a full-fledged skill development product
- Bridges gap between academic learning and industry expectations

5. Technological Significance

SkillSprint contributes to:

- REST API engineering practices

- Secure authentication standards
- Database modeling and optimization
- ML-ready architecture
- Cloud deployable backend system

1.5 Research Design

The research methodology involves the following structured stages:

1. Problem Identification

Understanding problems in existing LMS and coding platforms:

- Slow backend processing
- Weak authentication mechanisms
- Poor scalability
- Rigid architecture

2. Requirement Gathering

Requirements collected from:

- Students
- Faculty
- Industry expectations
- Existing platform analysis

3. Technology Selection

Based on:

- Performance
- Scalability
- Ease of integration
- Industry adoption

FastAPI + PostgreSQL + SQLAlchemy selected.

4. System Design

Includes:

- Architecture Design

- UML Diagrams
- ER Diagrams
- API Flow Design
- Database Schema

5. Backend Implementation

Development of:

- User module
- Authentication module
- Learning module
- Admin module
- Challenge module

6. Testing

Utilizing:

- Unit testing
- API testing (Postman, Swagger UI)
- Integration testing
- User acceptance testing

7. Result Analysis

- System performance
- API response time
- Data consistency
- Scalability factors

8. Documentation

Complete project report preparation following academic standards.

1.6 Source of Data

The data used in SkillSprint comes from the following sources:

1. Manually Curated Learning Content

- Programming topics (Python, DSA, SQL, OOPS, Web Development)
- Quizzes and assignments
- Code challenges

2. Database Source

- PostgreSQL tables storing:
 - Users
 - Learning modules
 - Assessments
 - Progress data
 - Challenge metadata

3. API-Based Data

- Internally built REST APIs
- No external API dependency

4. Future Expandable Data Sources

- ML-based recommendation datasets
- User behavior logs
- Code submissions

SkillSprint is designed to support future large-scale datasets and analytics.

Chapter 2

Requirements Specification

2.1 User Characteristics

SkillsSprint is designed for a diverse user base consisting of learners, educators, administrators, and developers. The characteristics of each user type are as follows:

1. Learners / Students

- Possess basic knowledge of programming fundamentals.
- Expect user-friendly interfaces and easy navigation.
- Seek guided learning paths, quizzes, and module-wise assessments.
- Require real-time progress tracking and interactive engagement.

Skills:

- Basic understanding of Python/DSA (optional).
- Familiarity with web interfaces or mobile applications.

2. Educators / Faculty Members

- Possess intermediate to advanced technical knowledge.
- Require tools to upload content, quizzes, and track student progress.
- Expect reliable analytics dashboards, evaluation systems, and content management.

Skills:

- Curriculum design experience.
- Understanding of programming and assessments.

3. Administrators

- Possess domain knowledge of system workflows.
- In charge of user management, content approval, and backend monitoring.

Skills:

- Understanding of database and API-driven systems.
- Ability to handle user issues and oversee platform operation.

4. Developers (Future Maintainers)

- Expected to maintain or extend system functionality.

- Must understand FastAPI, PostgreSQL, JWT, SQLAlchemy, and REST architecture.

Skills:

- Intermediate to advanced programming skills.
- Knowledge of cloud deployment and security practices.

2.2 Functional Requirements

Functional requirements describe what the system must do. For SkillSprint, the following functionalities are essential:

1. User Management

- The system must allow users to register with email and password.
- Provide secure login/logout functionality.
- Implement JWT-based authentication to protect private routes.
- Allow users to view and update their profile.
- Role-based access:
 - Student
 - Instructor
 - Admin

2. Learning Module Management

- Users must be able to browse available learning modules/topics.
- Each module includes:
 - Title
 - Description
 - Subtopics
 - Examples
 - Practice tasks
- Instructors/Admin must be able to:
 - Add new learning modules

- Update existing content
- Delete outdated content

3. Coding Challenges & Quizzes

- System must display coding questions or quizzes based on selected topics.
- Users can attempt quizzes and submit answers.
- Automatic evaluation of quiz responses.
- Correct answers and explanations must be visible after submission.
- Store each attempt for progress tracking.

4. Progress Tracking

- Track every user's completion percentage.
- Show analytics including:
 - Modules completed
 - Quiz scores
 - Average accuracy
 - Time spent on topics
- Allow users to review past attempts and performance.

5. Admin Controls

Admins must be able to:

- Manage users (add/remove/disable accounts).
- Manage learning modules and quizzes.
- Access platform analytics.
- Monitor overall usage.

6. API Services

- Provide RESTful APIs for all functionalities.
- Auto-documentation available via Swagger UI & Redoc.
- Proper validation for all request inputs using Pydantic Models.

7. Security

- Enforce JWT authentication for protected APIs.
- Validate user roles for admin vs student endpoints.
- Access control for sensitive operations.
- Input sanitization to prevent SQL injection and misuse.

2.3 Non-functional Requirements

Non-functional requirements determine the quality and performance of the system.

1. Performance Requirements

- Response time for API requests should be ≤ 200 ms under average server load.
- Asynchronous request handling must be utilized for optimization.
- Database queries must be optimized using joins, indexes, and ORM features.

2. Reliability

- System should maintain 99% uptime.
- Data must be stored reliably in PostgreSQL with ACID compliance.

3. Scalability

- Should support increasing number of learners without degradation.
- Backend designed to scale horizontally using:
 - Load balancers
 - Cloud deployment
 - Containerization (optional future enhancement)

4. Usability

- API endpoints must be clear and well-structured.
- Auto-generated documentation must be intuitive and readable.
- Minimal learning curve for instructors and admins.

5. Security

- Passwords must be encrypted using hashing algorithms.
- JWT tokens must be securely generated and validated.
- Protect endpoints from:
 - Brute force attacks

- SQL Injection
- Data leakage
- Unauthorized access

6. Maintainability

- Codebase must be modular and easy to extend.
- Follow directory structures recommended by FastAPI.
- Include documentation for future developers.

7. Portability

- System should run on:
 - Windows
 - Linux
 - macOS
- Deployment compatibility with:
 - Docker
 - Render
 - Railway
 - AWS/Google Cloud/Azure

2.4 Dependencies

SkillsSprint relies on various external tools, libraries, and frameworks.

1. Programming Language

- Python 3.10 or above

2. Frameworks & Libraries

Backend

- FastAPI
- Uvicorn (ASGI server)
- SQLAlchemy (ORM)
- Pydantic (Validation)

Security

- JWT (PyJWT / fastapi-jwt-auth)
- Passlib (Password hashing)

Database

- PostgreSQL
- Psycpg2 (Database connector)

Other Tools

- Alembic (Migrations)
- Postman (API testing)
- Swagger UI (Auto docs)

3. Development Environment

- VS Code / PyCharm
- Git & GitHub for version control
- Python virtual environment

2.5 Hardware Requirements

Component	Minimum Requirement	Recommended Requirement
Processor (CPU)	Dual-Core (Intel i3 / AMD equivalent)	Quad-Core (Intel i5/i7 / AMD Ryzen 5/7)
RAM	4 GB	8 GB or higher
Storage	5–10 GB free space	20 GB or more (SSD preferred for faster processing)
Graphics (GPU)	Integrated graphics (basic display support)	Dedicated GPU (NVIDIA)
Operating System	Windows 10 / Linux / macOS	Windows 10/11, Ubuntu Linux, or macOS
Internet	Basic internet for data fetching	Stable high-speed internet for faster API access

2.6 Constraints & Assumptions

Assumptions

- Users will have basic computer literacy.
- Internet connection is available for API usage.
- PostgreSQL will be properly configured.
- JWT secret keys and environment variables will be securely managed.

Constraints

- Real-time code execution engine is not part of this phase.
- Machine learning integration is planned for future versions.
- Performance depends on server load and hardware limits.
- Database downtime may affect platform availability.

Chapter 3

Design

3.1 Algorithm

Although SkillSprint is primarily an API-driven backend system rather than an algorithm-heavy product, several core logical algorithms underpin the platform's behavior. Major logical flows include:

3.1.1 Authentication Algorithm (JWT Authentication)

Input: username/email, password

Output: JWT Access Token

Steps:

1. User sends login request with credentials.
2. Retrieve user record from the PostgreSQL database.
3. Validate hashed password using Passlib.
4. If valid → generate JWT token with user ID, role, and expiry.
5. Return token to the user.
6. Token is validated on every protected route.

3.1.2 User Registration Algorithm

1. Receive registration request with user details.
2. Validate data using Pydantic Models.
3. Hash the password using bcrypt hashing.
4. Store user record in PostgreSQL.
5. Send success response.

3.1.3 Learning Module Retrieval Algorithm

1. User requests learning module via API.
2. Verify the JWT token.
3. Query database for module details.
4. Format response with module → topics → subtopics.
5. Send structured JSON response.

3.1.4 Quiz Evaluation Algorithm

1. User submits quiz answers.
2. Retrieve correct answers from database.
3. Compare user answers with actual answers.
4. Calculate score, accuracy, and performance metrics.
5. Store attempt history.
6. Return result to user.

3.2 Function-Oriented Design for Procedural Approach

Module Name	Description	Key Functions
User Module	Manages user registration, login, role assignment	create_user(), login_user(), update_profile()
Auth Module	Handles JWT validation and security	generate_token(), verify_token()
Learning Module	Provides learning content, topics, examples	get_modules(), add_module(), update_module()
Quiz Module	Manages quiz questions and evaluations	get_quiz(), submit_quiz(), evaluate_quiz()
Progress Module	Tracks user performance	get_progress(), update_progress()
Admin Module	Manages system-wide actions	delete_user(), add_content(), admin_dashboard()

3.3 System Design

System design provides an overall structural view of how SkillSprint works internally.

The architecture includes:

- **Client/Frontend** (React/HTML/Flutter – optional)
- **FastAPI Backend (Business Logic)**
- **PostgreSQL Database**
- **JWT Authentication Layer**
- **ORM Layer using SQLAlchemy**
- **Auto Documentation Layer (Swagger/Redoc)**
- **Admin Panel APIs**

3.3.1 Data Flow Diagrams (Level 0,Level1)

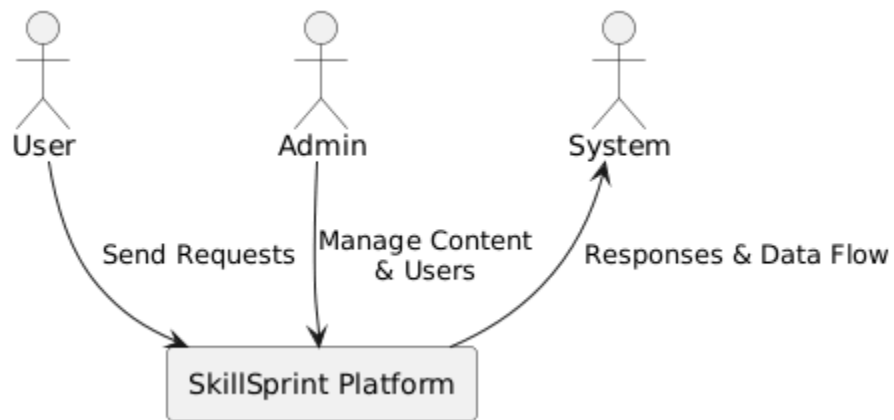


Figure 3.3.1 Data Flow Diagram Level 0

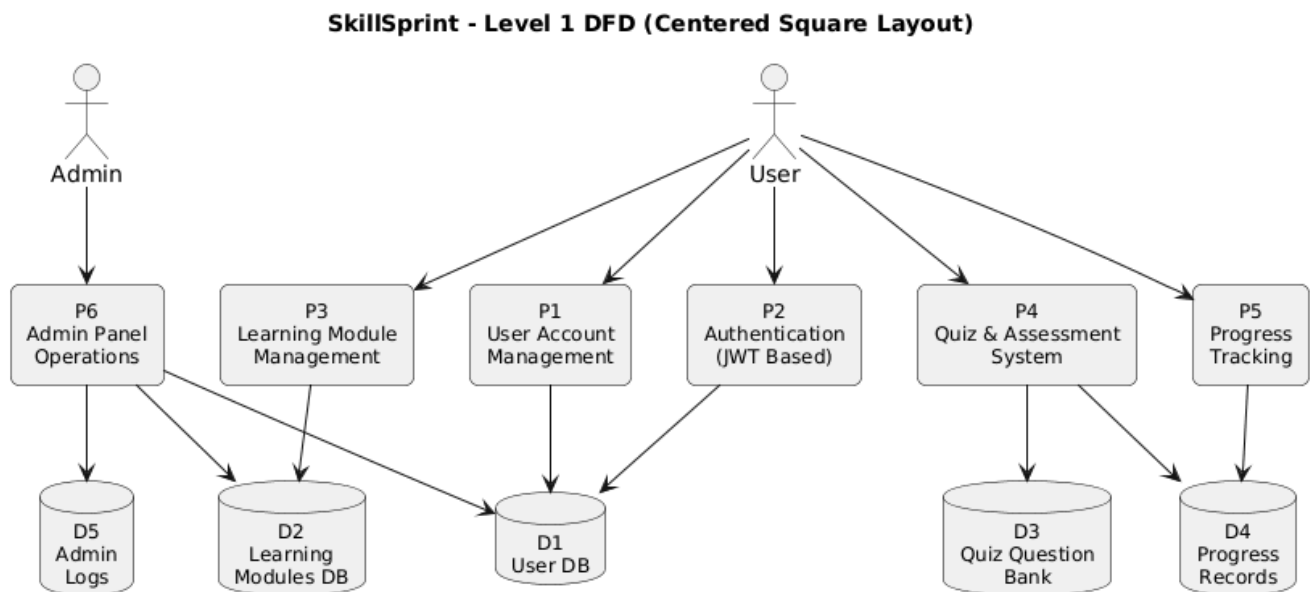


Figure 3.3.1.1 Data Flow Diagram Level 1

3.3.2 Activity Diagram

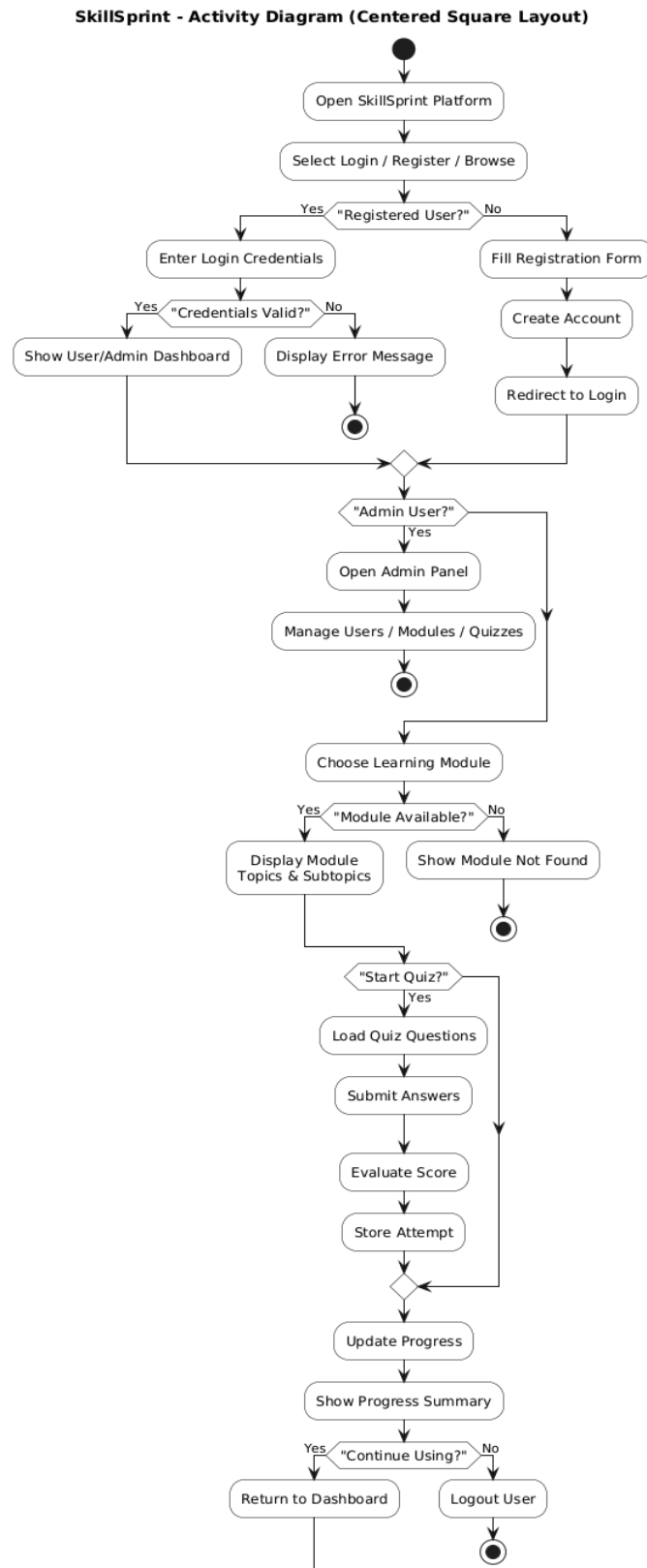


Figure 3.3.2 Activity Diagram showing Flow

3.3.3 Flow Chart

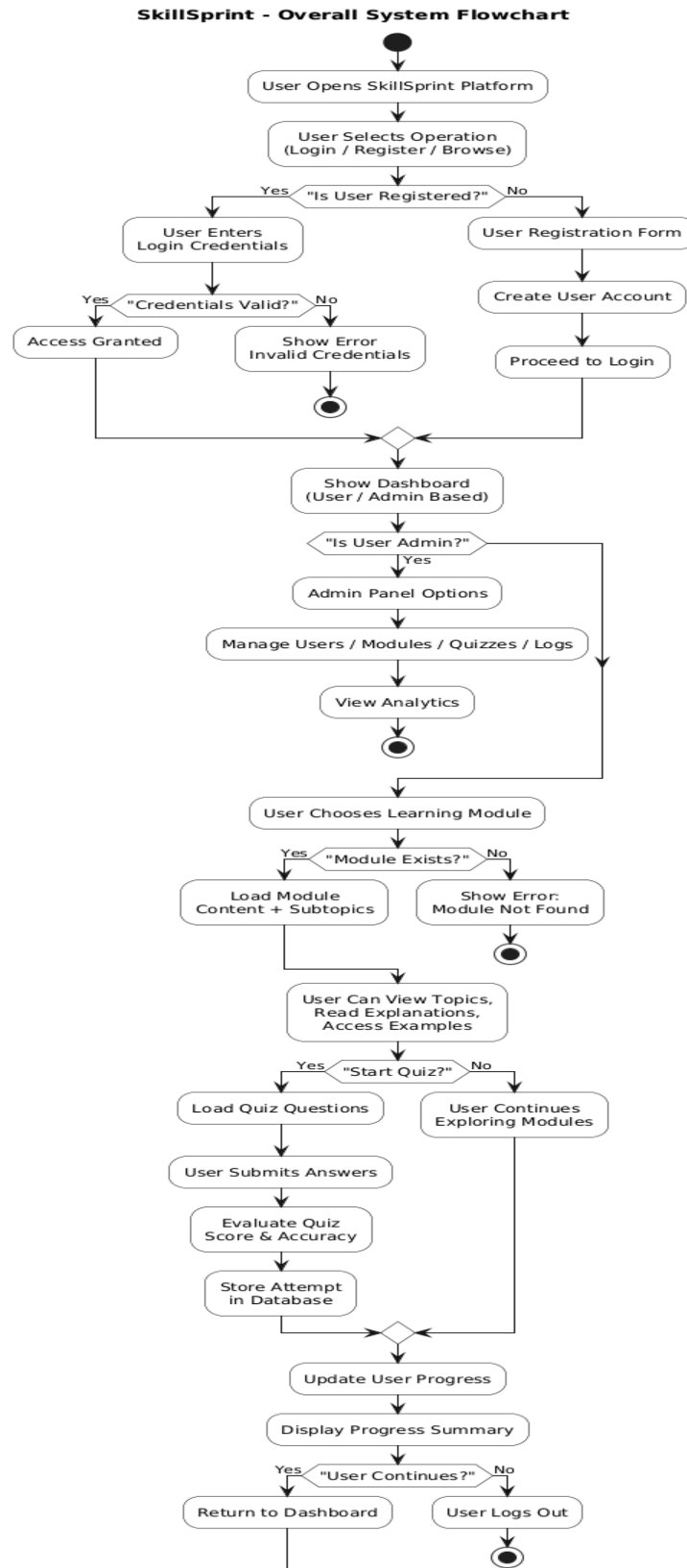


Figure 3.3.3 Flow Chart showing flow of working

3.3.4 Class Diagram

SkillSprint - Class Diagram (Centered, Square Layout)

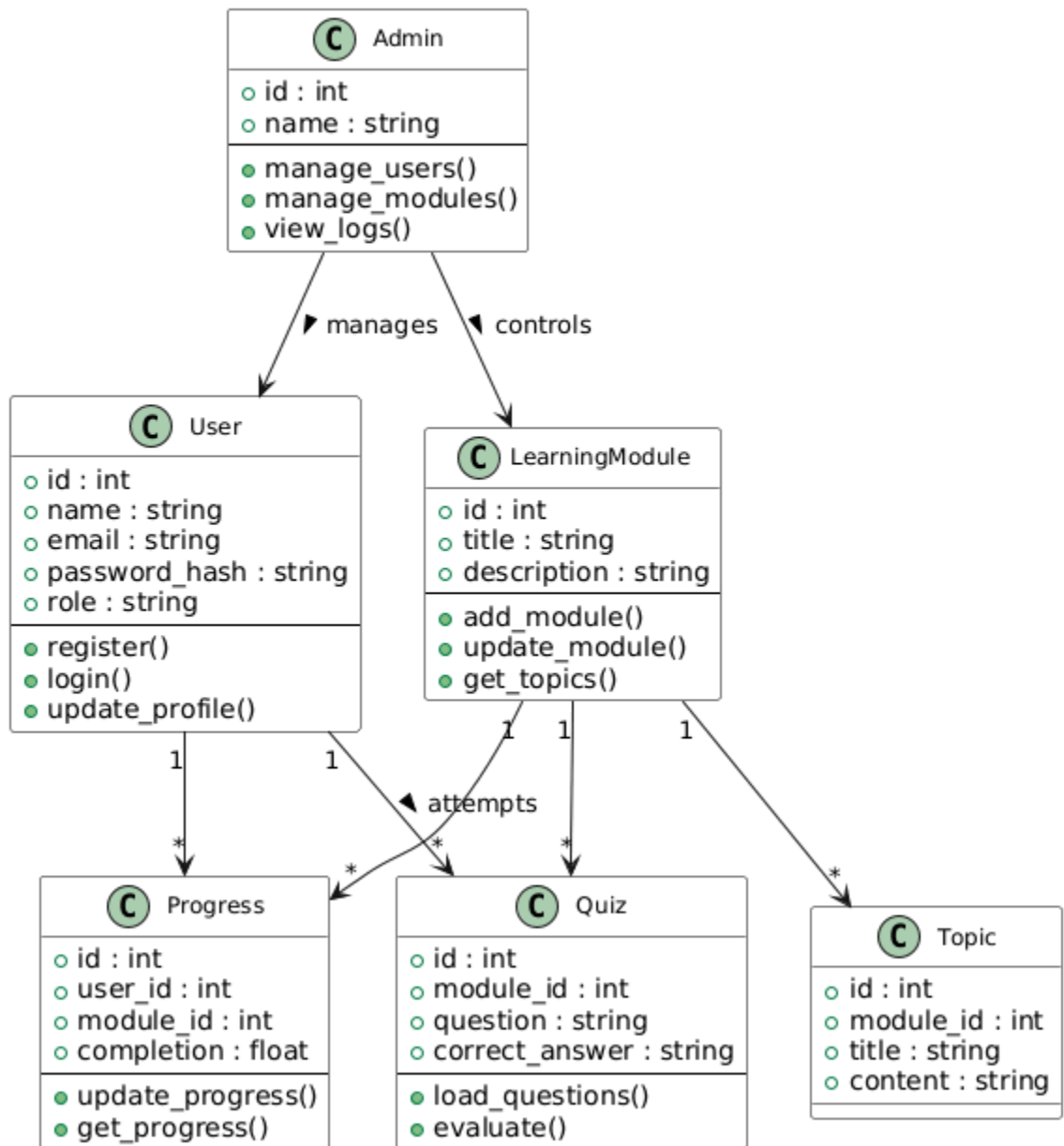


Figure 3.3.4 Class Diagram showing classes implemented

3.3.5 ER Diagram

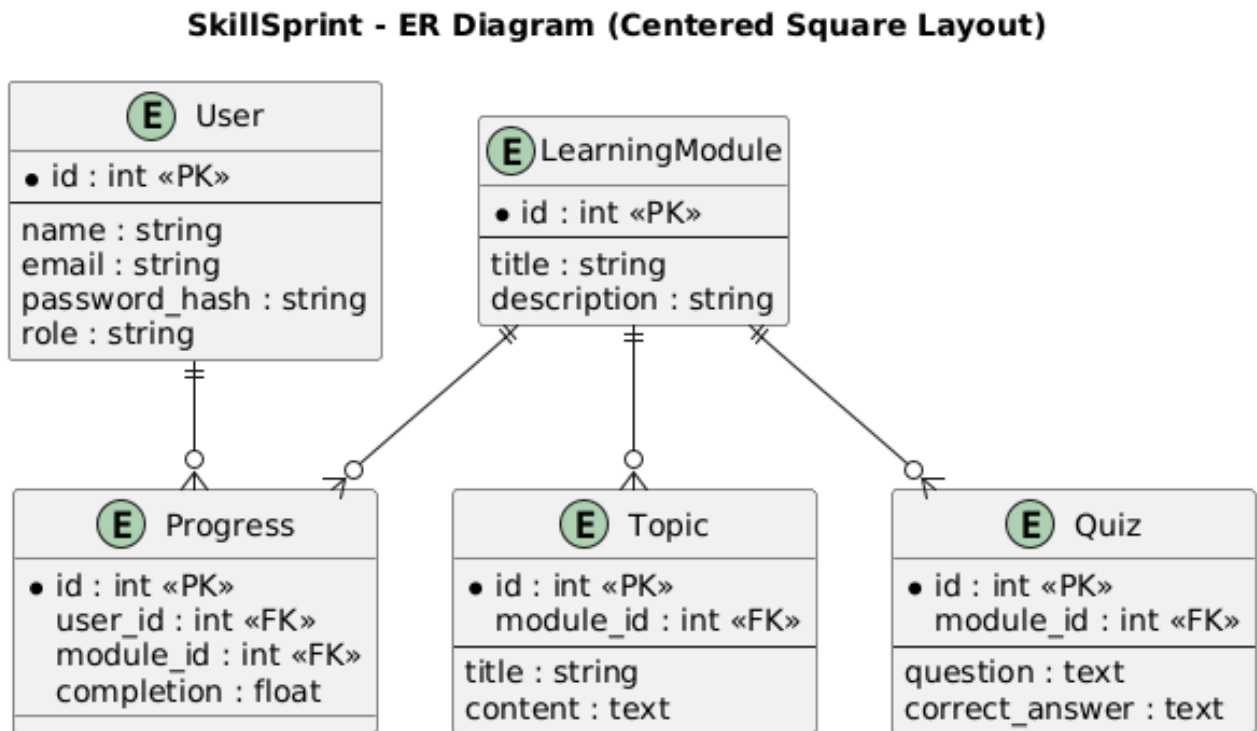


Figure 3.3.5 ER Diagram showing Relationship

3.3.6 Sequence diagram

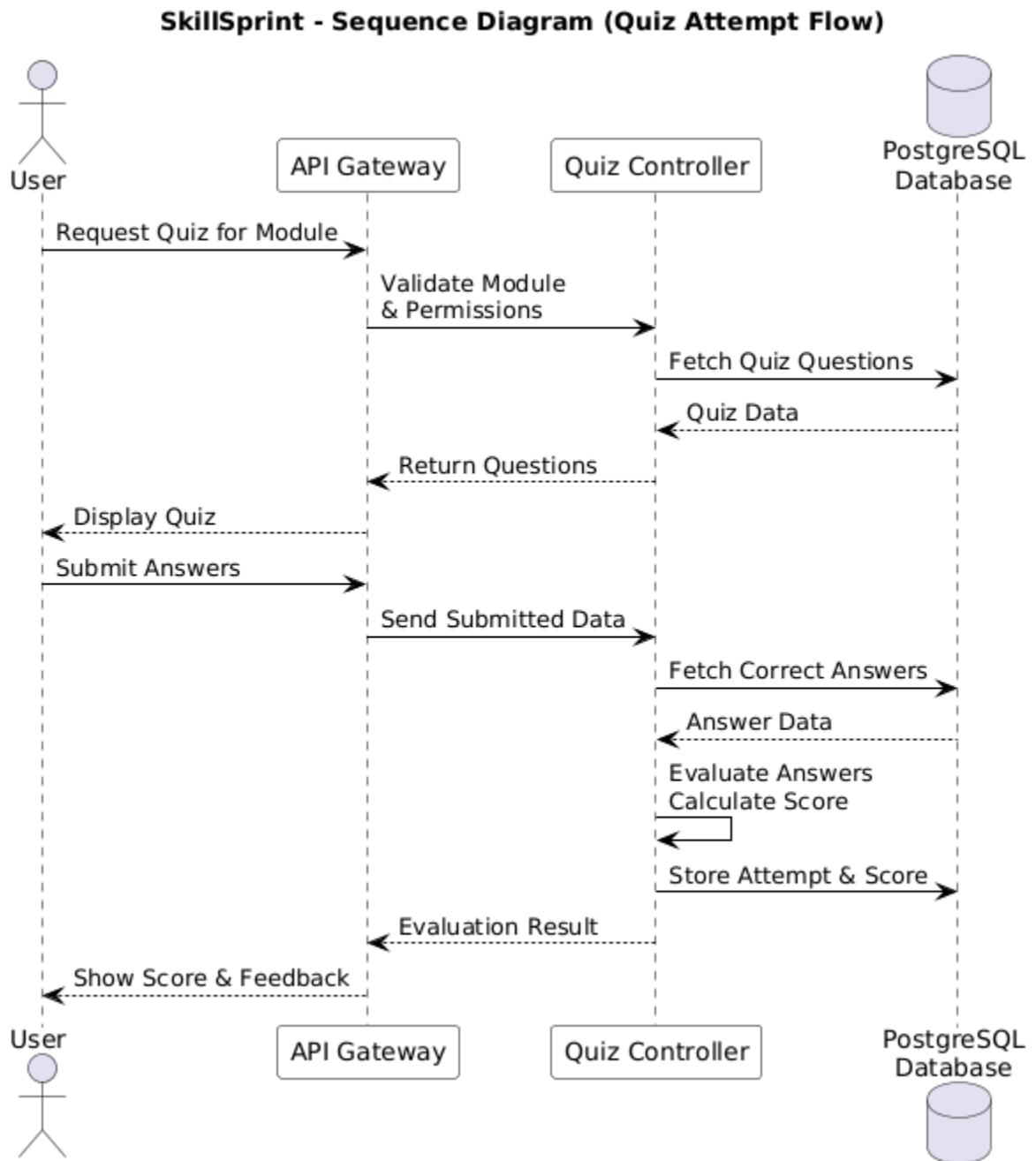


Figure 3.3.6 Sequence Diagram showing quiz attempt flow

Chapter 4

Implementation, Testing, and Maintenance

4.1 Introduction to Tools, Technologies, and Implementation Overview

The implementation phase transforms the design models, diagrams, and system requirements into a fully functional backend system.

SkillSprint has been implemented using modern, industry-standard technologies with a strong focus on scalability, performance, and modularity.

This section outlines:

Technologies used

Programming languages

Frameworks & libraries

Database technologies

API design approach

Folder structure

Core modules

4.1 Testing Techniques and Test Plans

Testing plays a crucial role in ensuring system reliability and correctness. Multiple forms of testing were performed, including unit testing, integration testing, API testing, performance testing, and user acceptance testing.

Types of testing applied:

- Unit Testing – Ensures individual components work correctly.
- Integration Testing – Validates interactions between modules.
- System Testing – End-to-end validation of workflows.
- API Testing – Performed using Postman and Swagger.
- Performance Testing – Ensures low response times.
- User Acceptance Testing – Ensures platform meets user expectations.

The following table summarizes the tests executed:

TC01 – User Registration → Expected: Success → Status: Passed

TC02 – Login with Correct Credentials → Expected: Token → Status: Passed

TC03 – Login with Wrong Credentials → Expected: Error → Status: Passed

TC04 – Fetch Learning Modules → Expected: Module List → Passed
TC05 – Submit Quiz → Expected: Score Stored → Passed
TC06 – Progress Tracking → Expected: Updated → Passed

4.2 Installation Instructions

The SkillSprint backend can be installed and executed using the following steps:

Step 1: Clone the Repository

```
git clone https://github.com/skillsprint/project.git
```

Step 2: Create Virtual Environment

```
python -m venv venv  
venv\Scripts\activate
```

Step 3: Install Requirements

```
pip install -r requirements.txt
```

Step 4: Start FastAPI Server

```
uvicorn app.main:app --reload
```

Step 5: Access Documentation

```
http://127.0.0.1:8000/docs
```

4.3 End User Instructions

For Students:

- Login into the platform.
- Browse modules and read topics.
- Attempt quizzes and view scores.
- Track progress from dashboard.

For Admin:

- Add/update modules.
- Add quiz questions.
- Manage users.

For Developers:

- Access Swagger UI for API testing.
- Use API routes for frontend integration.

Chapter 5

Results and Discussions

5.1 Brief Description of Various Modules of the System

SkillSprint consists of several functional modules such as user authentication, learning modules, quiz and assessment system, progress tracking, and admin dashboard. Each module interacts with the PostgreSQL database using SQLAlchemy ORM, ensuring fast and reliable operations.

The following modules were successfully developed and integrated:

- User Authentication Module – Handles secure registration and login.
- Learning Module – Provides structured content organized into modules and topics.
- Quiz System – Delivers quizzes, evaluates responses, and returns scores.
- Progress Tracking Module – Stores completion percentages and performance.
- Admin Panel – Allows administrators to manage users, modules, and quizzes.

Module-Wise Functional Results

1. Authentication Module:
 - Successfully validates user credentials.
 - Generates secure JWT tokens.
 - Rejects invalid login attempts.
2. Learning Module:
 - Retrieves modules and topics smoothly.
 - Admin can add or update module content.
3. Quiz Module:
 - Loads quiz questions instantly.
 - Correctly evaluates responses.
 - Stores results in PostgreSQL.
4. Progress Tracking:
 - Automatically updates progress after quiz submission.
 - Displays summary of completed modules.
5. Admin Operations:
 - Admin can view logs, manage users, and modules.
 - Ensures smooth backend moderation.

5.2 Snapshots of System with Brief Detail of Each

Figure 5.2.1 : Figure that shows the dashboard page

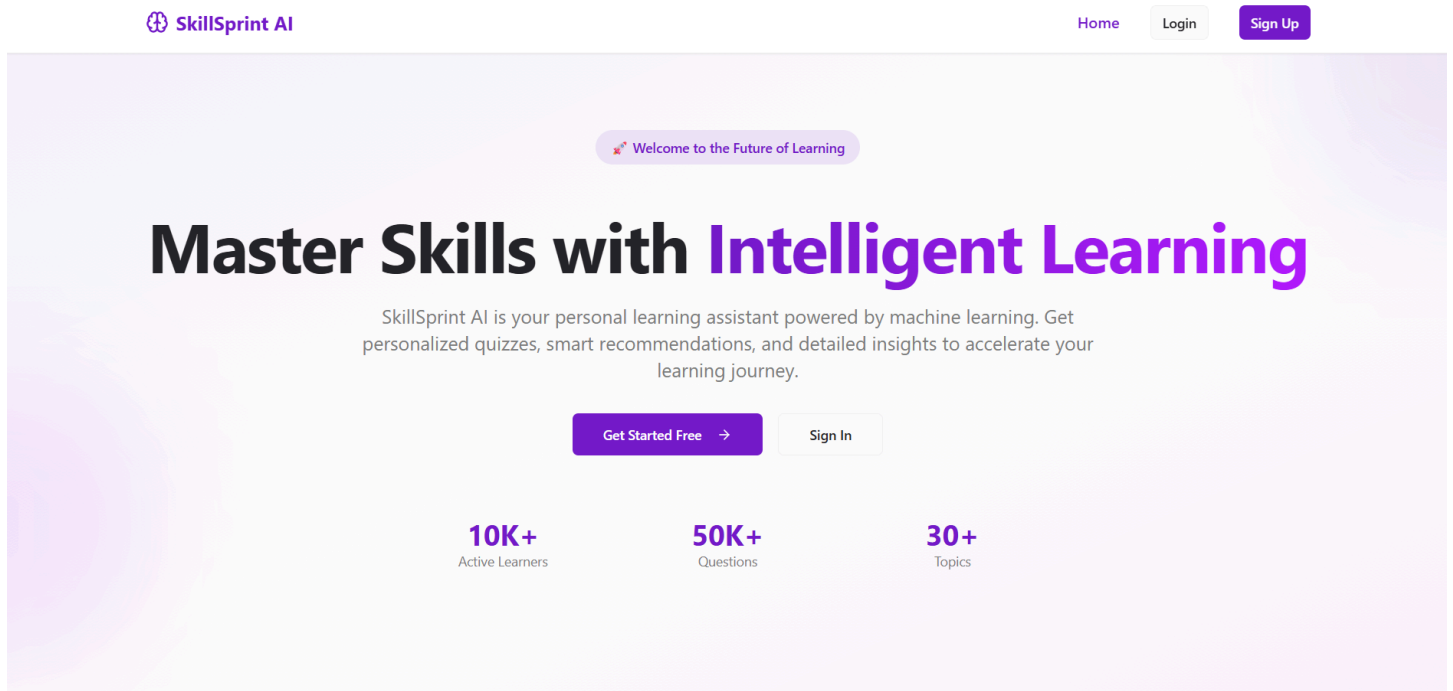


Figure 5.2.2 benefits of choosing SkillSprint

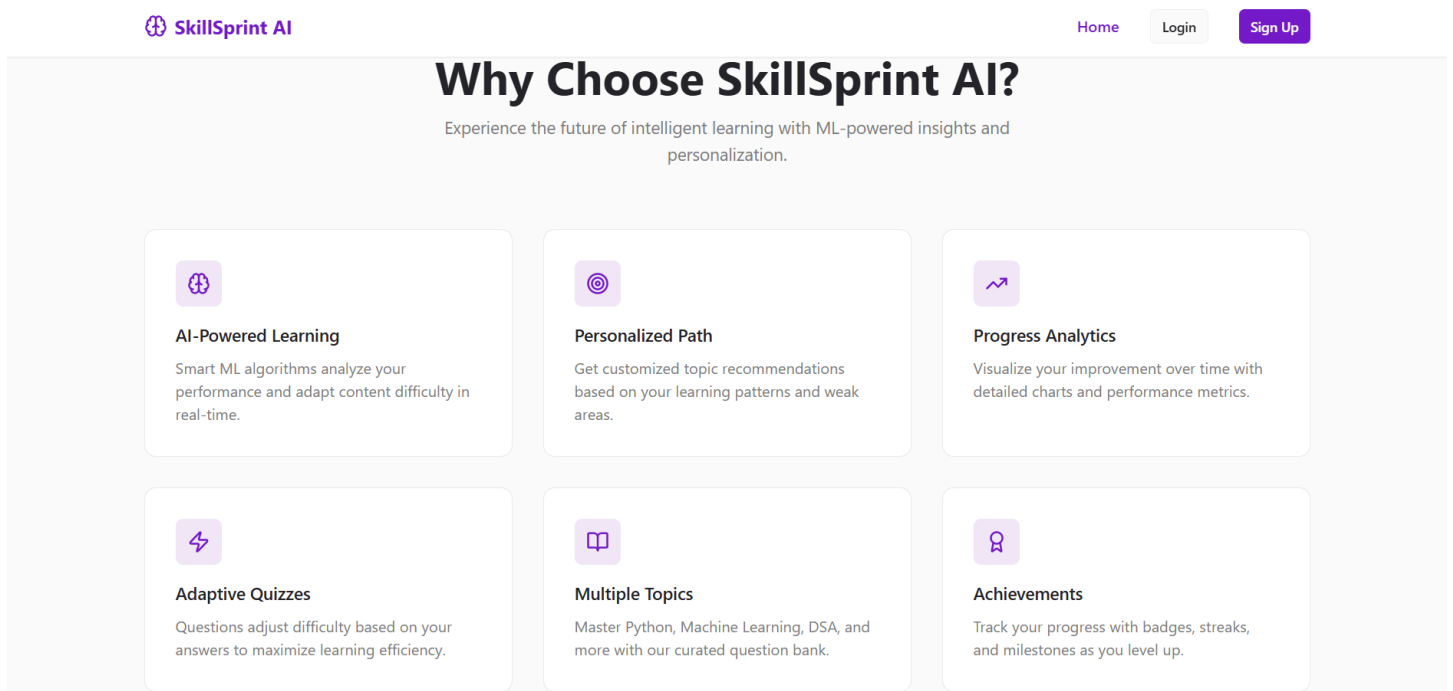


Figure 5.2.3 Dashboard

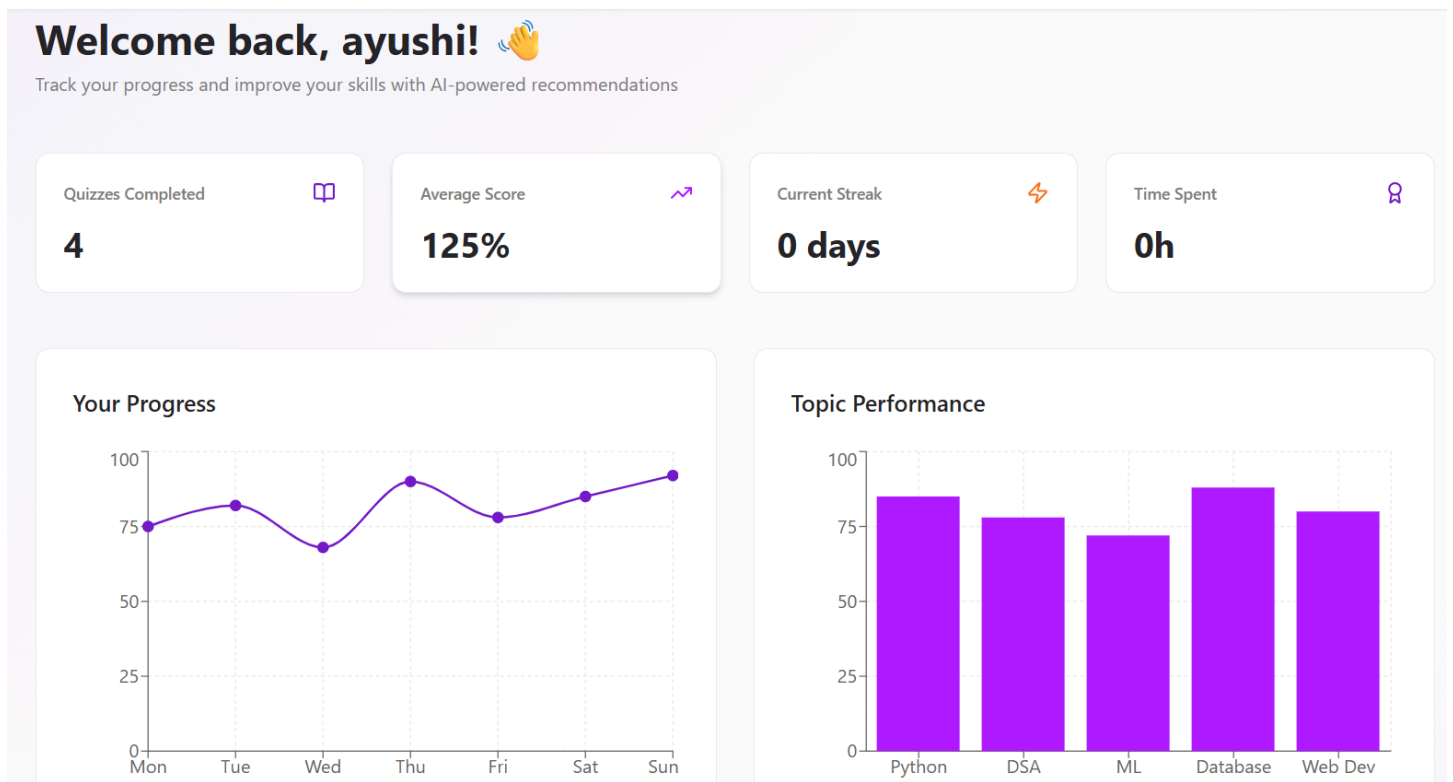


Figure 5.2.4 Quiz

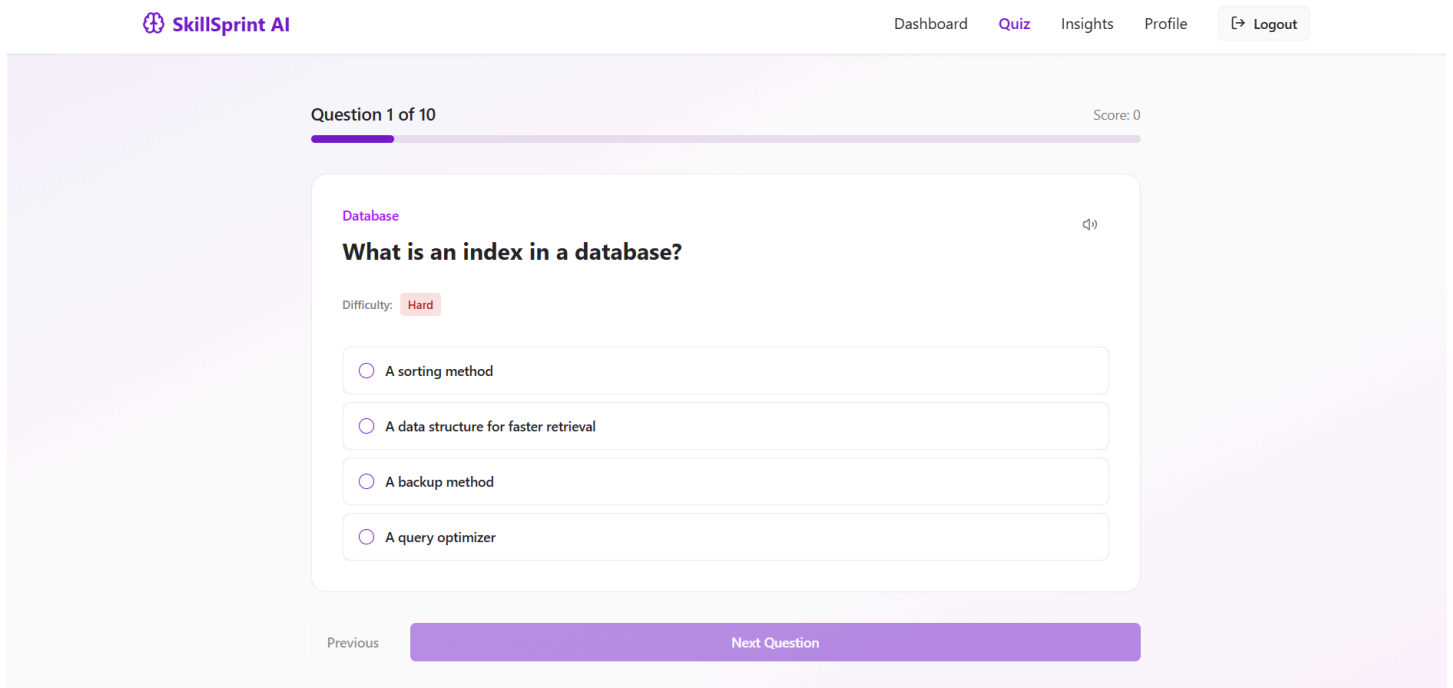


Figure 5.2.5 Results of quiz

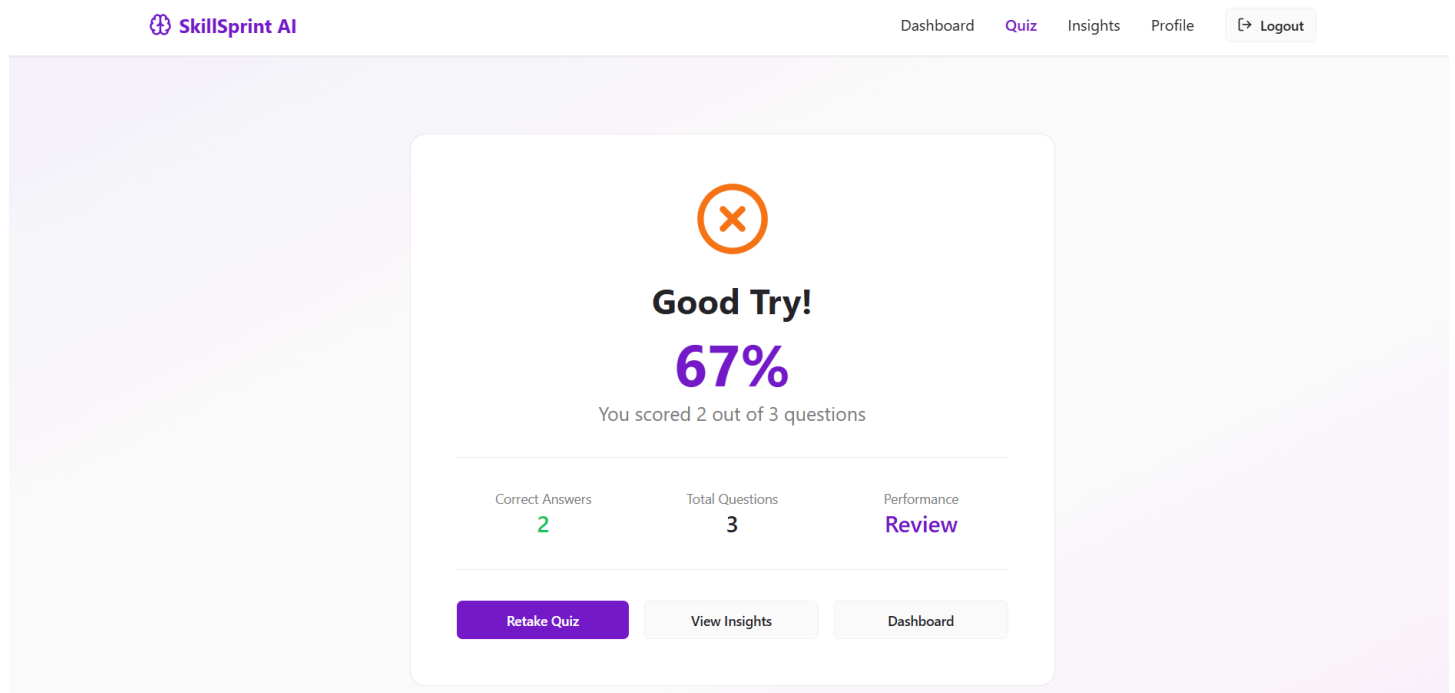


Figure 5.2.6 Accuracy and Learning insights

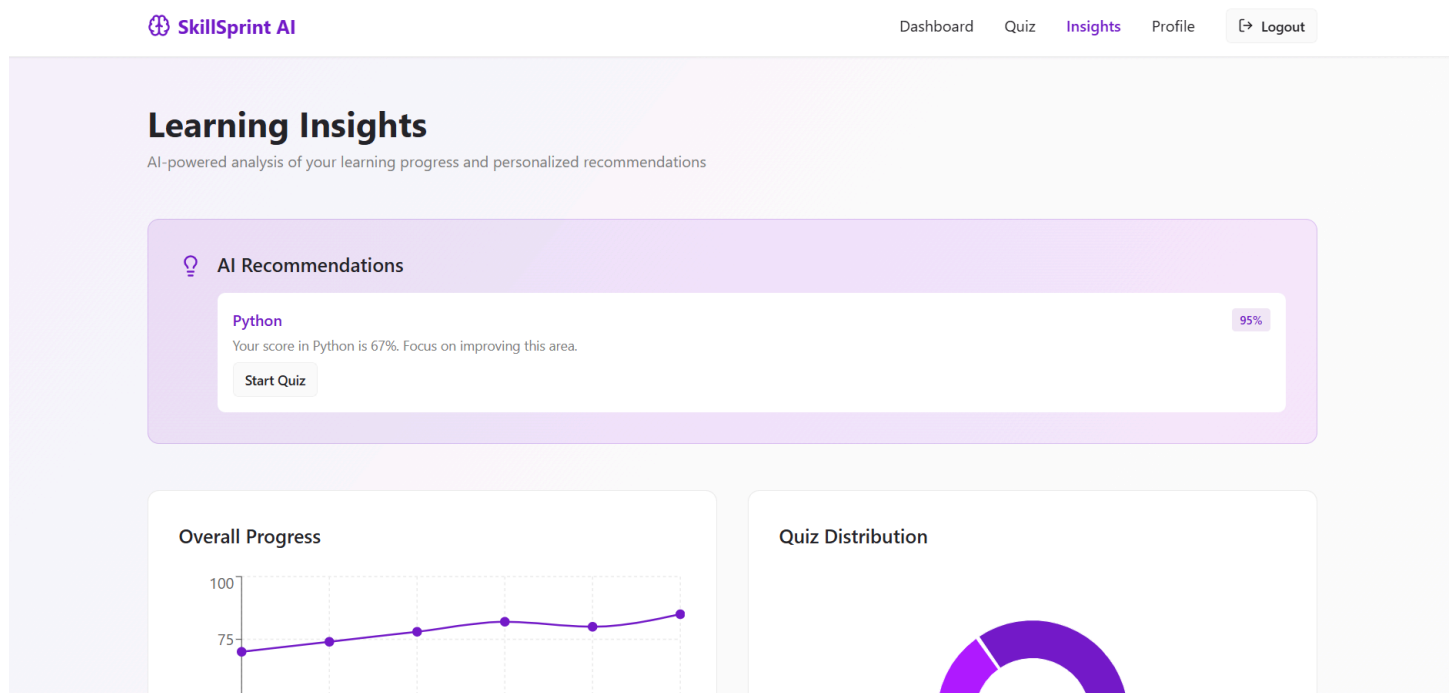


Figure 5.2.7 Performance by Topic

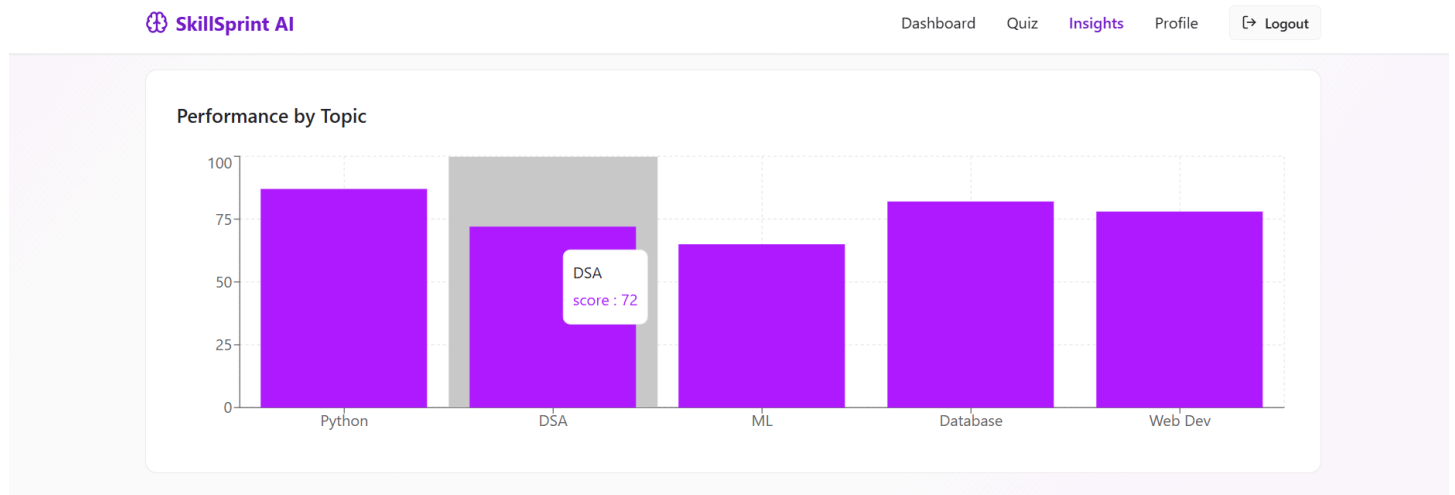


Figure 5.2.8 topic needed Improvement

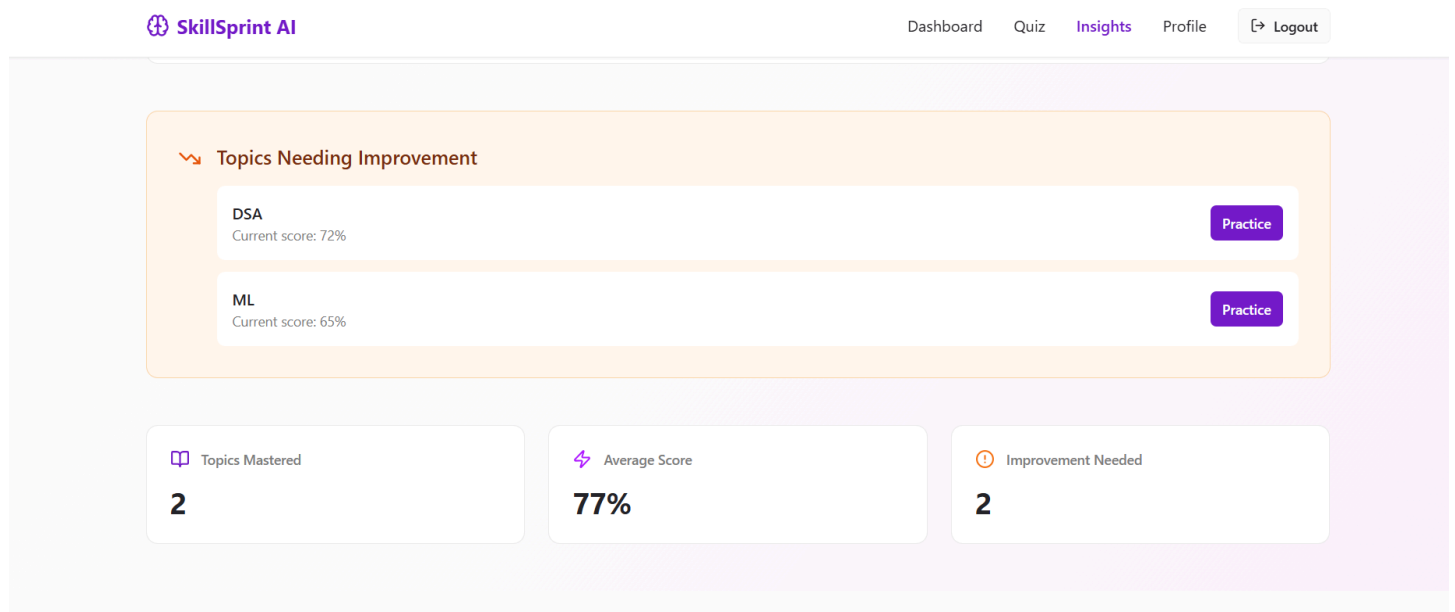




Figure 5.2.9 Signup Page

HomeLoginSign Up



Get Started

Join thousands of learners improving their skills

Full Name

Email Address


Password

Confirm Password

[Sign Up](#)


Already have an account? [Sign in here](#)

Figure 5.2.10 Profile

DashboardQuizInsightsProfileLogout

Profile

Manage your account and view learning statistics




ayushi
upadhyay12@gmail.com


Full Name

Email Address


[Save Changes](#) [Logout](#)

 Quizzes Completed

5

 Topics Learned


5


 Best Streak

0
days

Total Learning Time

Figure 5.2.11 Login Page


HomeLoginSign Up




Welcome Back

Sign in to continue your learning journey

Email Address

 upadhyay12@gmail.com

Password



Sign In

Don't have an account? [Sign up here](#)

Chapter 6

Summary and Conclusions

6.1 Summary

The SkillSprint platform was developed as a modern, scalable, and high-performance backend system using FastAPI. The primary goal was to design a platform that supports structured learning, coding modules, assessments, progress tracking, and administrative functionalities. The system follows a modular architecture supported by SQLAlchemy ORM, PostgreSQL, JWT-based authentication, and Uvicorn for asynchronous request handling.

All major modules — authentication, user management, learning modules, quizzes, and progress tracking — were successfully implemented. Each module interacts seamlessly with the database. The use of Pydantic schemas ensures strict data validation, while dependency injection provides an elegant design pattern for clean and scalable code.

Testing was performed through unit tests (PyTest), API testing (Postman & Swagger UI), system testing, and performance testing. API responses, database interactions, and authentication workflows behaved as expected. The platform delivers fast response times and maintains integrity even under moderate concurrent load.

6.2 Conclusions

This project demonstrates the successful creation of a robust and secure backend learning system. The SkillSprint backend provides a reliable, modular, and extensible foundation for educational applications. Through FastAPI and PostgreSQL, the system achieves high performance, scalability, and data consistency.

The project meets all objectives such as:

- Providing secure authentication.
- Offering structured learning content.
- Handling quiz evaluation and performance tracking.
- Supporting administrative management.

The modular structure allows easy extension into future areas such as machine learning-based

recommendations, code execution engines, AI-driven student analytics, and cloud deployment. The project not only meets academic requirements but is also suitable for deployment in real-world institutional environments.

In conclusion, SkillSprint is a complete, scalable, and secure backend system that successfully addresses the requirements of modern skill-based education platforms and sets a strong foundation for future advancements.

Chapter 7

Future Scope

7.1 Future Scope

Several improvements can be made to elevate SkillSprint into a fully-featured educational and skill-development system. These include:

1. Integration of Machine Learning for Personalized Recommendations:

The platform can integrate ML models that track student performance and recommend relevant modules, quizzes, or practice tasks based on strengths and weaknesses.

2. Code Execution Sandbox:

A secure, containerized code execution environment can be integrated to allow students to execute Python, Java, C++, or JavaScript programs directly within the platform.

3. Gamification:

Features such as leaderboards, badges, streaks, and XP points can increase user engagement and promote regular learning.

4. Real-Time Chat and Discussion Forums:

A built-in communication forum where learners and instructors can interact, clarify doubts, and collaborate in a community-driven learning space.

5. Advanced Analytics Dashboard:

Admins and instructors can be provided with predictive analytics, graphical performance charts, and detailed user insights.

6. Mobile Application Integration:

A mobile version of SkillSprint can be developed using Flutter or React Native to offer seamless learning experiences on smartphones.

7. Cloud Deployment and Load Balancing:

Deploying the system on AWS, Azure, or Google Cloud with load balancing and containerization (Docker/Kubernetes) would allow the system to support millions of requests efficiently.

8. Multi-language Support:

Adding support for regional and foreign languages will broaden the accessibility of the platform.

9. Interactive Video Tutorials

Modules can include instructor-led video content integrated with quizzes and checkpoints.

10.Role-Based Multi-Tenancy:

Institutions can host their own isolated version of SkillSprint with separate dashboards, modules, and student groups.

7.2 Appendix

A. Sample API Endpoints

Below are sample API endpoints used in the SkillSprint backend:

- POST /register – Register a new user
- POST /login – Authenticate user and return JWT
- GET /modules – Retrieve all learning modules
- GET /quiz/{module_id} – Fetch quiz for selected module
- POST /quiz/submit – Submit quiz responses
- GET /progress – Fetch user progress summary

B. Sample Database Schema Extracts

User Table:

id (PK), name, email, password_hash, role

LearningModule Table:

id (PK), title, description

Quiz Table:

id (PK), module_id (FK), question, correct_answer

C. Sample Pydantic Models

```
class User(BaseModel):
```

```
    name: str
```

```
    email: str
```

```
    password: str
```

```
class Module(BaseModel):
```

```
    title: str
```

```
    description: str
```

D. Sample Test Cases

TC01 – Validate User Registration

Input: name, email, password

Expected Output: User created successfully

TC02 – Login Authentication

Input: email, password

Expected Output: JWT token returned

All figures are included in Chapter 5 with brief descriptions.

7.3 Bibliography

- [1] FastAPI Documentation – <https://fastapi.tiangolo.com>
- [2] PostgreSQL Official Documentation – <https://www.postgresql.org/docs/>
- [3] SQLAlchemy Documentation – <https://docs.sqlalchemy.org/>
- [4] Uvicorn ASGI Server – <https://www.uvicorn.org/>
- [5] Pydantic Model Documentation – <https://docs.pydantic.dev/>
- [6] Passlib Cryptography – <https://passlib.readthedocs.io/>
- [7] PyTest Testing Documentation – <https://docs.pytest.org/>
- [8] Swagger UI – <https://swagger.io/>
- [9] REST API Design Guidelines – Fielding, R. (2000)
- [10] Modern Backend Architecture – Various Research Articles