

SKILLVOYAGE

Mini Project Report

Submitted by

Biya Aby

Reg. No.: AJC20MCA-I027

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

(INMCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



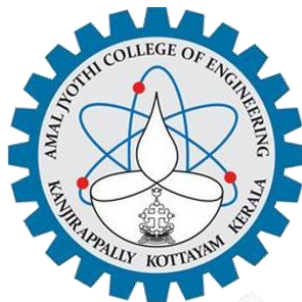
AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2024-2025

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Mini Project report, “**SkillVoyage**” is the bona fide work of **BIYA ABY (Regno: AJC20MCA-I027)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2024-25.

Ms. Merin Chacko

Internal Guide

Mr. Binumon Joseph

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the Mini project report “**SkillVoyage**” is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the **Integrated Master of Computer Applications (INMCA)** from **APJ Abdul Kalam Technological University**, during the academic year 2024-2025.

Date:

BIYA ABY

KANJIRAPPALLY

Reg: AJC20MCA-I027

ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Mr. Binumon Joseph, Assistant Professor** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Merin Chacko, Assistant Professor** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

BIYA ABY

ABSTRACT

The SkillVoyage project is an innovative online platform designed to empower individuals in discovering and developing diverse artistic talents through structured courses led by skilled instructors. Aiming to bridge learners with guides across various art forms such as dance, music, drawing, and martial arts, SkillVoyage offers a seamless, immersive learning experience that incorporates personalized instruction, interactive feedback, and an integrated marketplace for art-related products. With dedicated modules for administrators, guides, and students, the platform ensures a streamlined, community-oriented environment for skill development.

At the core of SkillVoyage is a dynamic, user-friendly interface that accommodates both students and guides. Students can enroll in courses, attend sessions, purchase related products, and leave feedback, allowing them to engage meaningfully with the art form and track their progress. Guides, on the other hand, can showcase their qualifications, select specific art categories and subcategories, create comprehensive courses with detailed lessons, and view student feedback, fostering a collaborative, growth-centered environment. The platform also integrates a secure payment gateway, enabling students to complete course enrollments and product purchases with ease.

The administration module of SkillVoyage plays a vital role in maintaining the platform's quality and functionality. Admins oversee content management by adding new categories and subcategories, approving guides, publishing courses, and monitoring the marketplace, ensuring a well-regulated, high-quality learning ecosystem. Additionally, the platform's marketplace feature allows students to explore and purchase curated art products, further enriching their learning journey and offering guides an avenue to reach a broader audience with their creations.

SkillVoyage stands as a comprehensive ecosystem dedicated to nurturing artistic growth, combining accessibility, community, and commerce. By integrating tailored features for learners, instructors, and administrators, the platform redefines the online learning experience in the arts, promoting individual skill development while fostering connections within the creative community. Through this holistic approach, SkillVoyage contributes meaningfully to the cultural and personal growth of its users, supporting both established artists and emerging enthusiasts alike.

CONTENT

SL. NO	TOPIC	PAGE NO
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	3
2.1	INTRODUCTION	4
2.2	LITERATURE REVIEW	4
2.3	PROPOSED SYSTEM	6
2.4	ADVANTAGES OF PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	7
3.1	FEASIBILITY STUDY	8
3.1.1	ECONOMICAL FEASIBILITY	8
3.1.2	TECHNICAL FEASIBILITY	8
3.1.3	BEHAVIORAL FEASIBILITY	9
3.1.4	FEASIBILITY STUDY QUESTIONNAIRE	9
3.2	SYSTEM SPECIFICATION	13
3.2.1	HARDWARE SPECIFICATION	13
3.2.2	SOFTWARE SPECIFICATION	13
3.3	SOFTWARE DESCRIPTION	13
3.3.1	DJANGO	13
3.3.2	SQLITE	14
4	SYSTEM DESIGN	16
4.1	INTRODUCTION	17
4.2	UML DIAGRAM	17
4.2.1	USE CASE DIAGRAM	18
4.2.2	SEQUENCE DIAGRAM	19
4.2.3	ACTIVITY DIAGRAM	21
4.2.4	CLASS DIAGRAM	23
4.2.5	OBJECT DIAGRAM	24
4.3	USER INTERFACE DESIGN USING FIGMA	25
4.4	DATA BASE DESIGN	29
4.4.1	RELATIONAL DATABASE MANAGEMENT SYSTEM	29

4.4.2	NORMALIZATION	29
4.4.3	SANITIZATION	31
4.4.4	INDEXING	31
4.5	TABLE DESIGN	32
5	SYSTEM TESTING	43
5.1	INTRODUCTION	44
5.2	TEST PLAN	45
5.2.1	UNIT TESTING	45
5.2.2	INTEGRATION TESTING	46
5.2.3	USER ACCEPTANCE TESTING	46
5.2.4	AUTOMATION TESTING	47
5.2.5	SELENIUM TESTING	47
6	IMPLEMENTATION	56
6.1	INTRODUCTION	57
6.2	IMPLEMENTATION PROCEDURE	57
6.2.1	USER TRAINING	58
6.2.2	TRAINING ON APPLICATION SOFTWARE	58
6.2.3	SYSTEM MAINTENANCE	58
6.2.4	HOSTING	59
7	CONCLUSION & FUTURE SCOPE	62
7.1	CONCLUSION	63
7.2	FUTURE SCOPE	63
8	BIBLIOGRAPHY	64
9	APPENDIX	66
9.1	SAMPLE CODE	67
9.2	SCREEN SHOTS	106
9.3	GIT LOGS	114

List of Abbreviation

- SkillVoyage - Online Platform for Artistic Skill Development
- IDE - Integrated Development Environment
- CNN - Convolutional Neural Network
- CSV - Comma-Separated Values
- HTML - HyperText Markup Language
- CSS - Cascading Style Sheets
- UI - User Interface
- UX - User Experience
- SQL - Structured Query Language
- SQLite - SQL-based Lightweight Database Engine

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

SkillVoyage is an online platform designed to support individuals in discovering and developing artistic skills such as dance, music, drawing, and martial arts. By providing dedicated modules for students, guides, and administrators, SkillVoyage creates a structured, user-friendly environment for skill development. Students can enroll in courses, make payments, access materials, and provide feedback, while guides showcase their qualifications, create courses, and engage with students. Administrators manage categories, approve guides, and publish courses, ensuring quality across the platform. Built on a scalable tech stack with Python, Django, and SQLite, SkillVoyage fosters a vibrant learning community, supporting personal growth and cultural connection.

1.2 PROJECT SPECIFICATION

SkillVoyage is a comprehensive online platform designed to facilitate artistic skill development by connecting students with skilled guides in various art forms. The platform features three user roles:

- **Student:** Registers, enrolls in courses, makes payments, attends lessons, and provides feedback. Students can also purchase art-related products and manage their learning experience through an intuitive interface.
- **Guide:** Submits qualifications for approval, selects relevant art categories, creates courses and lesson plans, and engages with enrolled students. Guides can view feedback and track student progress, helping them refine their teaching approach.
- **Admin:** Oversees platform operations, including adding categories, approving guide requests, publishing courses, and managing product listings. The admin ensures quality and security across all interactions, creating a well-regulated learning environment.

SkillVoyage aims to empower learners, support guides, and foster a vibrant community of artistic growth and cultural connection through a secure, user-friendly digital ecosystem.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

SkillVoyage is designed to transform the way individuals explore and develop artistic skills, connecting aspiring students with qualified guides across diverse art forms like dance, music, drawing, and martial arts. Many learners and instructors face challenges in accessing structured, interactive learning environments tailored for the arts. SkillVoyage addresses these gaps by offering a comprehensive digital platform that empowers students to enroll in courses, purchase materials, and receive feedback, all in a secure and user-friendly interface.

The platform supports guides by enabling them to showcase their qualifications, create courses, and engage with students, allowing them to expand their reach and enhance their teaching impact. A robust admin module further ensures platform quality by overseeing content management, guide approvals, and course publication.

By incorporating secure payment processing, an organized marketplace, and streamlined profile management, SkillVoyage fosters a vibrant ecosystem for artistic learning. This holistic approach not only simplifies skill acquisition and knowledge-sharing but also cultivates a community where learners and instructors can thrive, fostering creativity, cultural exchange, and personal growth in an accessible digital landscape.

2.2 LITERATURE REVIEW

Here's a structured literature review tailored for your SkillVoyage project:

- **Deep Learning for Art Style Recognition and Description Generation**

Recent advancements in deep learning have significantly impacted the field of art recognition, with various models being applied to classify art styles and generate corresponding descriptions [2]. Zhang et al. (2023) conducted a comprehensive review that covers different deep learning approaches, such as Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), and Vision Transformers (ViTs) [2]. The review emphasized the advantages of pre-trained models like ResNet and VGG in extracting meaningful features from art images [2]. Moreover, it explored the use of Natural Language Processing (NLP) techniques to generate descriptive captions based on visual content [3]. Challenges in recognizing modern, abstract, and mixed-media artworks due to their subjective nature and lack of labeled datasets were also discussed [2, 3].

- **Machine Learning Techniques for Automated Art Description**

Singh and Patel (2024) focused on using machine learning for automating the recognition and description of artworks [3]. Techniques like Transfer Learning, Fine-Tuning, and few-shot learning were highlighted for their ability to improve model accuracy with limited data [3]. While recent algorithms perform well on structured datasets, real-world applications face issues related to variability in art styles, lighting conditions, and occlusions [3].

- **Multimodal Approaches in Art Recognition**

Garcia et al. (2023) reviewed integrating multimodal approaches, combining image data with text, for art recognition tasks [4]. The study explored combining visual features from CNNs with textual data from descriptive metadata for more accurate classification and meaningful description generation [4]. Techniques like Visual Question Answering (VQA) and Cross-Modal Learning were emphasized for their potential to improve understanding of artworks [4].

- **Advanced Reinforcement Learning for Art Curation and Personalization**

Garcia et al. (2023) examined the use of reinforcement learning in the field of art curation and personalized art recommendations, highlighting how agents trained through reinforcement learning can adapt to user preferences, style inclinations, and viewing habits [6]. Their study emphasized the potential of using reinforcement learning in systems that recommend art pieces in virtual galleries or digital art platforms [6]. Techniques like Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) were explored for dynamically adjusting recommendations based on real-time user interaction data, with promising results in enhancing engagement and user satisfaction in art-viewing applications [6].

- **Explainable AI (XAI) for Interpreting Art Recognition Models**

Martinez et al. (2024) reviewed the emerging use of Explainable AI (XAI) in making art recognition models more interpretable for researchers and users. The review focused on methods like Gradient-weighted Class Activation Mapping (Grad-CAM) and SHAP (SHapley Additive exPlanations), which offer insights into the decision-making process of deep learning models. By visualizing which aspects of an artwork influence model predictions, XAI tools help in understanding complex patterns in abstract and modern art that might otherwise be challenging to interpret. The review also noted how XAI can help validate model accuracy and reliability in diverse art genres, ensuring that predictions align with human understanding and expert analysis.

2.3 PROPOSED SYSTEM

The proposed SkillVoyage platform is designed to revolutionize online learning for various art forms, offering a comprehensive and interactive solution for students, instructors (guides), and administrators. This user-friendly system enables students to enroll in courses tailored to their interests, complete secure payments, and provide feedback on their learning experiences. Instructors can submit their qualifications, create courses, upload lessons, and track enrolled students' progress and feedback, thus creating an engaging, supportive learning environment. The admin module allows for efficient management of categories, subcategories, and instructor approvals, along with course publishing and product management for e-commerce functionality. SkillVoyage integrates effective course organization, e-commerce features, and enhanced user interaction, ultimately making art education accessible and engaging.

2.4 ADVANTAGES OF PROPOSED SYSTEM

- Comprehensive Course Management for Art Learning
- Efficient and Secure Payment Processing
- Streamlined Instructor Approval and Content Creation
- User-Friendly Interface for Students and Instructors
- Dynamic E-commerce and Cart Functionality
- Enhanced Administrative Oversight and Flexibility

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility study assesses the viability of the SkillVoyage platform by examining its technical, economic, and operational aspects. This analysis evaluates whether the proposed technologies (Django for the backend, CSS/Bootstrap for the frontend, and a relational database for data management) meet project requirements. The economic feasibility addresses development costs, potential financial gains, and long-term investment returns. Operational feasibility examines how the platform will address user needs, integrate smoothly into the current organizational structure, and enhance user engagement. This comprehensive analysis aims to confirm the project's feasibility and alignment with educational technology trends.

3.1.1 Economical Feasibility

The economic feasibility of SkillVoyage focuses on assessing the project's financial viability, considering costs, potential revenue, and return on investment (ROI). The main costs include development, infrastructure, and ongoing maintenance. Revenue is expected from course enrollments, potential partnerships, and advertising.

Key components include:

- **Market Demand:** Assessing demand for online skill-learning platforms and identifying potential users, including students, hobbyists, and professional instructors.
- **Cost-Benefit Analysis:** Comparing development and operational costs with anticipated financial and educational benefits.
- **Scalability:** Evaluating the potential for service expansion with additional courses and features.
- **Risk Assessment:** Identifying financial risks, including market competition and user adoption challenges.
- **Funding Sources:** Exploring grants, partnerships, or investment opportunities to support development and expansion.

3.1.2 Technical Feasibility

The technical feasibility study evaluates *SkillVoyage*'s technological requirements and assesses the capabilities needed to implement and maintain the platform effectively. The analysis covers the required software, hardware, and infrastructure to ensure the platform's success.

Key components include:

- **Technology Stack:** Leveraging Django for backend development, CSS/Bootstrap for frontend design, and a relational database for user and course management.
- **System Integration:** Ensuring seamless interaction among modules, including authentication, course management, payment processing, and talent showcasing.
- **Scalability:** Assessing the system's ability to support growing numbers of users, courses, and media content as *SkillVoyage* expands.
- **Security Measures:** Implementing robust security protocols to protect user data, transactions, and intellectual property.
- **User Support:** Providing necessary documentation and support for effective platform utilization by students, instructors, and administrators.

3.1.3 Behavioral Feasibility

The operational feasibility study assesses the extent to which SkillVoyage will meet user needs, align with the organization's objectives, and provide value to students, instructors, and administrators.

Key components include:

- **User Requirements:** Identifying essential user needs, such as course enrollment, interactive learning, talent showcasing, and secure transactions.
- **Solution Acceptability:** Validating that SkillVoyage meets educational and organizational objectives, offering a user-friendly and supportive environment.
- **User Adaptability:** Assessing adaptability among users, including beta testing to gather feedback and refine platform functionality.
- **Alternative Solutions:** Exploring other solutions to ensure that SkillVoyage offers the most efficient and effective platform.

3.1.4 Questionnaire

1. Do you currently have a platform for your talent teaching services?

- Mini Gopi, Dance Teacher: "No, I don't have a dedicated online platform for my dance classes. Everything is managed through direct contact with students."
- Aswin, Music Instrument Coach: "No, I primarily rely on in-person lessons and some occasional remote sessions through Zoom or Google meet, but no formal platform."
- Aby George, Taekwondo Trainer: "No, all my classes and student interactions are handled

offline or through direct phone communication."

2. If you were to get a platform, what essential features would you like it to have?

- Mini Gopi, Dance Teacher: "I would love to have easy tools for creating and managing dance courses, high-quality video tutorials for students to follow, and a section where they can upload their practice videos for feedback."
- Aswin, Music Instrument Coach: "For me, it's important to have a robust course creation tool, high-quality video streaming and downloads for tutorials, and a feature where students can upload their practice recordings for review."
- Aby George, Taekwondo Trainer: "A platform that offers structured course management, high-quality video lessons, and a way for students to upload their training videos for feedback would be essential."

3. How do you currently manage your student enrollments and schedules?

- Mini Gopi, Dance Teacher: "At the moment, students usually contact me through phone calls or WhatsApp to enroll. I keep track of schedules and student information manually in a notebook."
- Aswin, Music Instrument Coach: "I handle enrollments through phonecalls or directly and maintain it to keep track of class schedules and student progress. "
- Aby George, Taekwondo Trainer: "I manage everything through direct phone calls and messages. I also use a physical calendar to keep track of classes and appointments, which can get quite overwhelming."

4. What type of interaction would you like to have with your students on the platform?

- Mini Gopi, Dance Teacher: "It would be great to have a forum where students can ask questions and share videos of their practice. Real-time chat would also be useful for immediate feedback."
- Aswin, Music Instrument Coach: "I would like to have a messaging system where I can communicate with students individually or in groups, and also a feedback feature where they can submit their practice recordings for review."
- Aby George, Taekwondo Trainer: "A direct messaging system would be helpful, along with the ability to create group discussions for class announcements and interactive sessions."

5. What challenges do you currently face with equipment for your talent classes?

- Mini Gopi, Dance Teacher: "Finding and recommending the right dance accessories for students can be challenging. A platform that offers a marketplace for these items would be very beneficial."
- Aswin, Music Instrument Coach: "Some students struggle to find quality instruments for practice. Having a rental or purchase option within the platform would be a huge help."
- Aby George, Taekwondo Trainer: "It's difficult for students to source proper training gear. An integrated shop on the platform would simplify this process a lot."

6. How do you handle feedback and assessment for your students?

- Mini Gopi, Dance Teacher: "Currently, I give feedback verbally after classes. An online system where students can upload their practice videos for review would make this process more efficient."
- Aswin, Music Instrument Coach: "Feedback is usually given during class or via email. An integrated feedback and assessment system would streamline this and allow for more detailed and structured critiques."
- Aby George, Taekwondo Trainer: "I provide feedback during training sessions. An online assessment tool where students can receive detailed feedback and track their progress would be very useful."

7. What kind of community features would enhance your teaching experience?

- Mini Gopi, Dance Teacher: "A community forum where students can share their practice videos, get feedback from peers, and participate in dance challenges would be fantastic."
- Aswin, Music Instrument Coach: "A space where students can post recordings of their performances and receive constructive criticism from both peers and instructors would be very valuable."
- Aby George, Taekwondo Trainer: "Community challenges and events, where students can compete and showcase their skills, would greatly enhance motivation and engagement."

8. How important is it for you to have analytics and reporting tools on the platform?

- Mini Gopi, Dance Teacher: "It's very important. I need to track student progress and identify areas where they might need additional help or encouragement."

- Aswin, Music Instrument Coach: "Analytics would help me understand which areas students are finding difficult, so I can adjust my teaching methods and provide more targeted support."
- Aby George, Taekwondo Trainer: "Essential. Detailed reports on attendance, progress, and performance would help me manage my classes more effectively and ensure every student is progressing well."

9. What type of certifications do you currently offer, and how are they managed?

- Mini Gopi, Dance Teacher: "I issue paper certificates at the end of each course. Moving to an automated digital certification system would make this process much simpler and more professional."
- Aswin, Music Instrument Coach: "Certificates are currently issued manually after assessments. Automating this process on the platform would save a lot of time and reduce errors."
- Aby George, Taekwondo Trainer: "I provide paper certificates upon course completion. An online system for issuing and tracking certificates would streamline this and make it easier for students to access their achievements."

10. How do you envision the equipment rental/purchase feature working on the platform?

- Mini Gopi, Dance Teacher: "Students should be able to browse a catalog of dance accessories, see prices, and either rent or buy items directly from the platform."
- Aswin, Music Instrument Coach: "A well-organized catalog with detailed descriptions and reviews of instruments, along with easy options to rent or purchase, would be ideal."
- Aby George, Taekwondo Trainer: "A marketplace with clear images and descriptions of training gear, where students can easily rent or buy what they need, would be very helpful."

11. How would you like the platform to handle event notifications?

- Mini Gopi, Dance Teacher: "Students should receive notifications about upcoming classes and events on their dashboard and via email or push notifications."
- Aswin, Music Instrument Coach: "Integration with a calendar and automatic reminders for upcoming sessions and events would ensure students never miss anything important."
- Aby George, Taekwondo Trainer: "Push notifications and email reminders for upcoming classes, competitions, and events would keep everyone informed and prepared."

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor: Intel Core i5

RAM: 8 GB or higher

Hard Disk: 512 GB SSD or higher

3.2.2 Software Specification

Front End - HTML, CSS, JavaScript

Backend - Django

Client on PC - Windows 7 and above.

Technologies used - PYTHON, JS, HTML5, AJAX, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 Eg. Django

Django is a robust, Python-based web framework designed to handle the complexities of data-driven applications with ease, making it a popular choice for developers building dynamic, scalable web applications. Following the Model-View-Template (MVT) architectural pattern, Django promotes a clean separation of business logic, data handling, and presentation, which simplifies both development and maintenance. Models represent the data structure and use Django's built-in ORM (Object-Relational Mapping) to connect to various databases like SQLite, PostgreSQL, or MySQL, providing a flexible way to interact with databases without extensive SQL. Views handle business logic and manage the flow of information between models and templates, while templates manage the presentation layer, ensuring that HTML and other presentation code remain separate from the business logic. Django is known for its "batteries-included" philosophy, meaning it comes with essential tools for web development right out of the box. These include a powerful built-in admin interface for content and user management, which reduces the need for third-party tools in many cases. It also includes robust tools for handling user authentication, form processing, session management, and more, which allow for secure and efficient management of users, roles, and permissions—a vital feature for applications like SkillVoyage, which require fine-grained access control. Security is a core focus, and Django provides built-in defenses against common vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), making it well-suited for applications that

handle sensitive information. With its scalability, comprehensive documentation, and extensive community support, Django allows developers to build reliable, performant applications, even for high-demand projects, while minimizing the need for additional components. Django's modular design and vast ecosystem of reusable libraries further enhance its flexibility, enabling developers to integrate various functionalities or extend the platform as needed. These features make Django an ideal framework for complex applications that require secure, multi-user environments and real-time interactions, like the SkillVoyage project.

3.3.2Eg. SQLite

SQLite is a highly efficient, self-contained relational database management system (RDBMS) that offers a lightweight, serverless solution for storing data, making it ideal for small to medium-sized applications and development environments. Unlike traditional databases that require a dedicated server, SQLite operates as an embedded database engine, with the entire database stored in a single file on the local file system. This architecture makes SQLite extremely easy to set up, as there are no server configurations, network permissions, or service management tasks required. Developers can integrate SQLite directly into an application without the need for additional software or complex setup, which simplifies deployment and reduces overall application overhead.

Despite its compact design, SQLite supports a wide range of SQL features found in more extensive database systems. It includes robust support for transactions, which ensures data integrity even in case of unexpected application closures. Its indexing capabilities allow for faster data retrieval, while support for complex queries and foreign key constraints provides flexibility in managing relational data. Additionally, SQLite is compatible with various data types and is capable of handling a significant amount of data, with a maximum database size of 140 terabytes, which far exceeds the needs of many applications.

For applications like SkillVoyage, SQLite is particularly advantageous during development phases. Its simplicity and low overhead make it easy to test database-driven functionality without the need for a heavyweight database server, facilitating rapid iteration and debugging. Furthermore, SQLite's reliability and durability are essential for applications handling user-generated content, providing a stable platform for managing users, courses, interactions, and content uploads. SQLite's atomic commit and rollback capabilities make it safe and reliable, even under heavy read/write operations.

As SkillVoyage grows, SQLite may serve as an efficient choice for local data storage or for single-user setups where remote access is unnecessary. However, if SkillVoyage's data needs expand significantly,

migrating to a more scalable RDBMS like PostgreSQL or MySQL is straightforward since SQLite is fully compatible with standard SQL syntax. In summary, SQLite's portability, ease of integration, and robust support for SQL features make it an excellent choice for applications that prioritize simplicity, reliability, and minimal administrative overhead, particularly during early-stage development and deployment for SkillVoyage.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process, or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance, and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram

- Object diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A Use Case Diagram is a visual representation of a system's functionality, highlighting the interactions between users (actors) and the system itself. It serves as a high-level overview, showing how users achieve their goals by utilizing the system's various functions, represented as "use cases." These diagrams are commonly used in the early stages of software design to communicate system requirements, focusing on what the system does rather than how it does it.

In a Use Case Diagram:

- **Actors:** These represent the users or external systems that interact with the SkillVoyage platform. Examples of actors include administrators, teachers, students, and third-party services that may interface with the application.
- **Use Cases:** These encapsulate the specific functionalities or actions the system performs. Common use cases for SkillVoyage might include "Login," "Register," "Submit Course," "View Course Catalog," and "Manage Teacher Requests."
- **Associations:** The lines connecting actors to use cases indicate the interactions or relationships between them. These associations clarify who can perform which actions within the system.
- **System Boundary:** Represented by a rectangle, this boundary encapsulates all the use cases and defines the scope of the SkillVoyage system, helping stakeholders understand which functionalities are included within the system.

Use Case Diagrams are instrumental in grasping user requirements, identifying essential functionalities, and defining roles within the SkillVoyage platform. They provide a foundation for further detailed analysis and design, ensuring that the system is aligned with user needs and expectations. By visually representing these interactions, stakeholders can easily comprehend how different users will engage with the system and what actions are available to them.

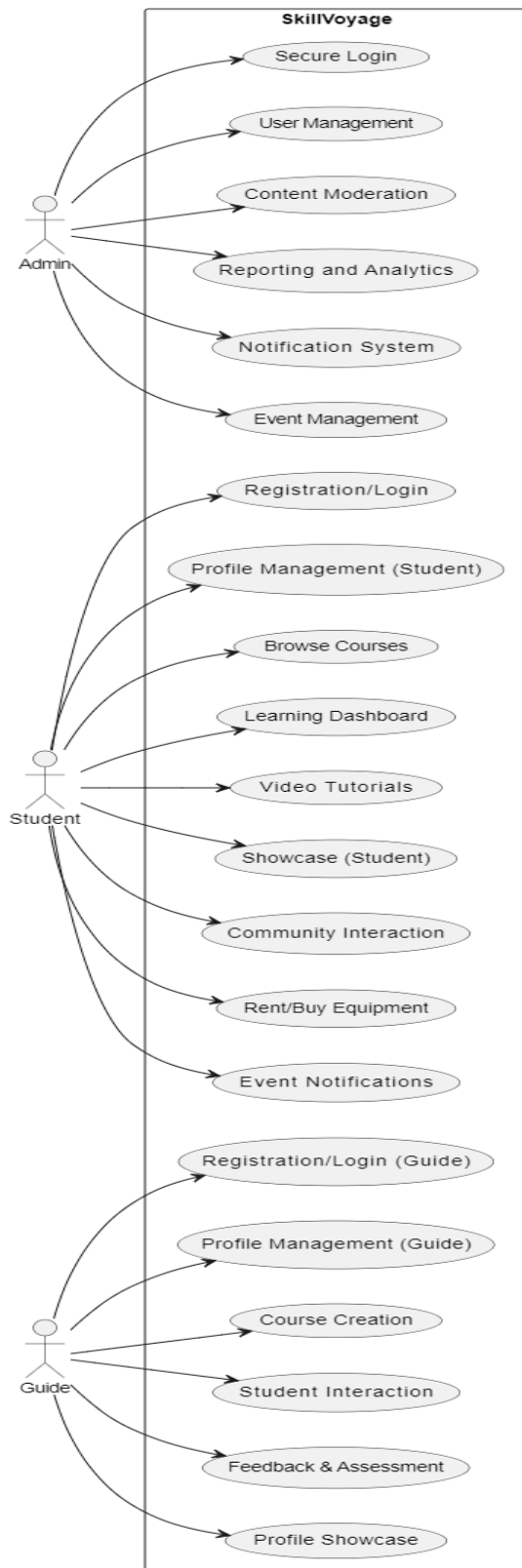


Fig 1.use case diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram in Unified Modeling Language (UML) that

shows how objects interact in a particular sequence within a system. It visually represents the flow of messages or events between objects over time, illustrating how and in what order operations are carried out.

Key components of a sequence diagram include:

Actors: Represent users or external systems interacting with the system.

Objects or Classes: Represent entities within the system that perform actions.

Lifelines: Vertical lines showing the life span of an object during the sequence.

Activation Bars: Rectangles on the lifelines indicating the period when an object is active.

Messages: Arrows between objects indicating communication; they can be synchronous or asynchronous.

Return Messages: Dashed lines that show the return or response after a message is processed.

Use and Advantages:

Sequence diagrams are integral to modelling the dynamic aspects of systems and are especially useful during the design phase for:

- **Clear Communication:** They provide a visual, step-by-step breakdown of processes, improving communication among team members.
- **Requirement Validation:** Diagrams help validate requirements by demonstrating how processes should flow and helping stakeholders confirm intended functionality.
- **System Design and Debugging:** Developers use sequence diagrams to refine system architecture, identify bottlenecks, and troubleshoot issues by understanding exact message flow.

Sequence diagrams support collaboration by showing the workflow in a way that is easy to understand and highly detailed. They help developers, stakeholders, and testers confirm requirements and understand how different components should interact, assisting in debugging, performance optimization, and overall design clarity.

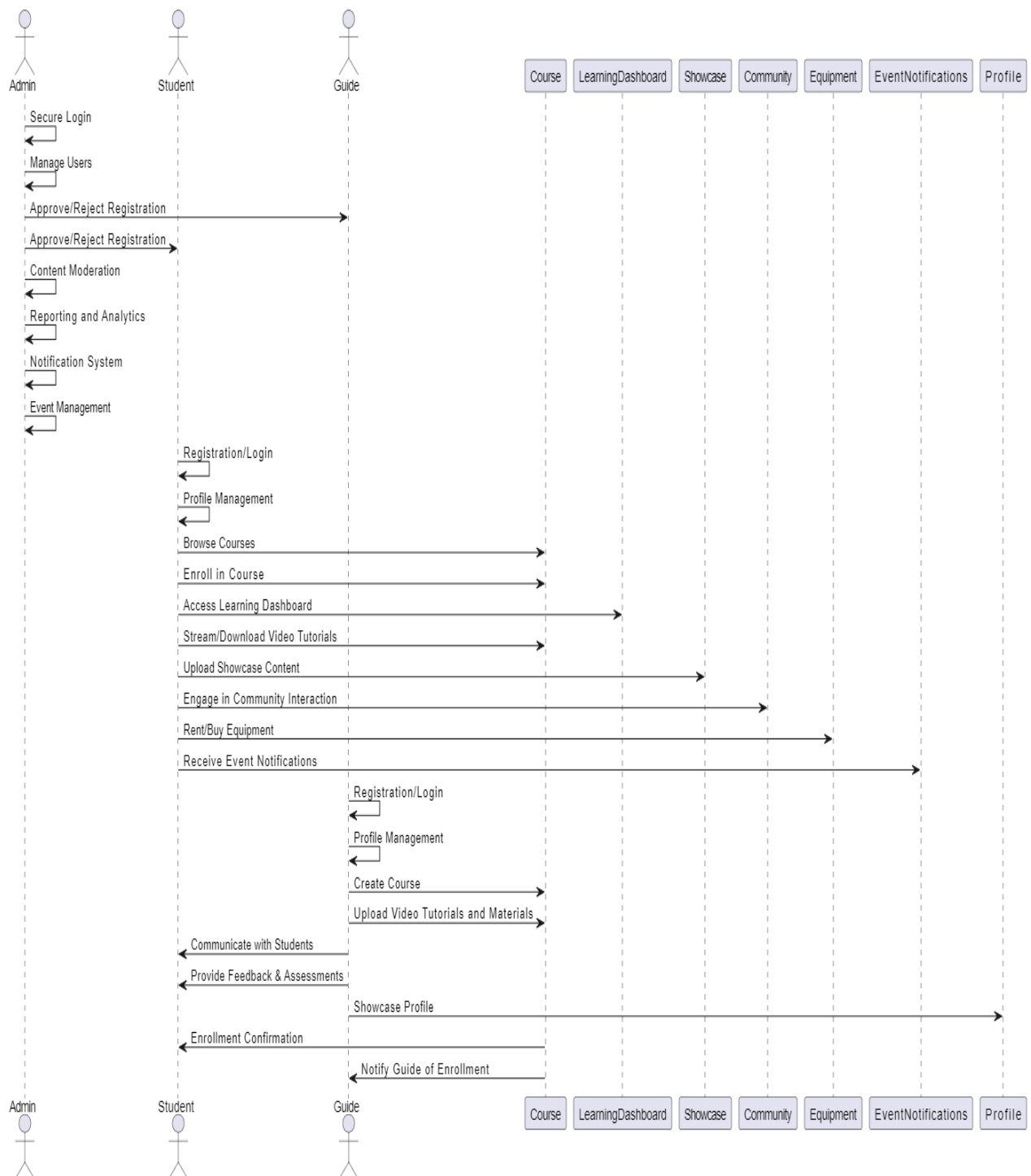


Fig 2.sequence diagram

4.2.3 Activity Diagram

An activity diagram is a behavioral diagram in Unified Modeling Language (UML) that illustrates the flow of activities or actions within a system, highlighting the dynamic aspects of processes. It features key elements such as activities, represented by rounded rectangles, which indicate tasks

performed within the system. The diagram begins with a start node, depicted as a filled circle, and concludes with an end node, a circle encased within another circle, marking the process's completion. Transitions, represented by arrows, connect activities to demonstrate the flow from one task to another. Decision nodes, illustrated as diamonds, indicate points where the flow can diverge based on specific conditions, while forks and joins (depicted as bars) represent parallel activities. Activity diagrams are particularly valuable for modeling complex workflows, use case scenarios, and business processes, as they clarify the sequence of operations, identify potential bottlenecks, and facilitate communication among stakeholders by visualizing activities and their interdependencies.



Fig 3.activity diagram

4.2.4 Class Diagram

A class diagram is a static structure diagram in Unified Modeling Language (UML) that represents the system's classes, their attributes, methods, and the relationships among them. It provides a visual blueprint of the system's architecture, highlighting how different classes interact and how data is organized. Each class is depicted as a rectangle divided into three sections: the top section contains the class name, the middle section lists its attributes, and the bottom section shows its methods (or operations). Relationships between classes are illustrated through lines, with various notations to indicate the type of relationship, such as inheritance (generalization), associations, aggregations, and compositions. Class diagrams are essential for object-oriented design, as they help in understanding the system's structure, guiding the implementation of classes, and ensuring that all components interact correctly. They also serve as a communication tool among developers, designers, and stakeholders by providing a clear representation of the system's data model and its relationships.

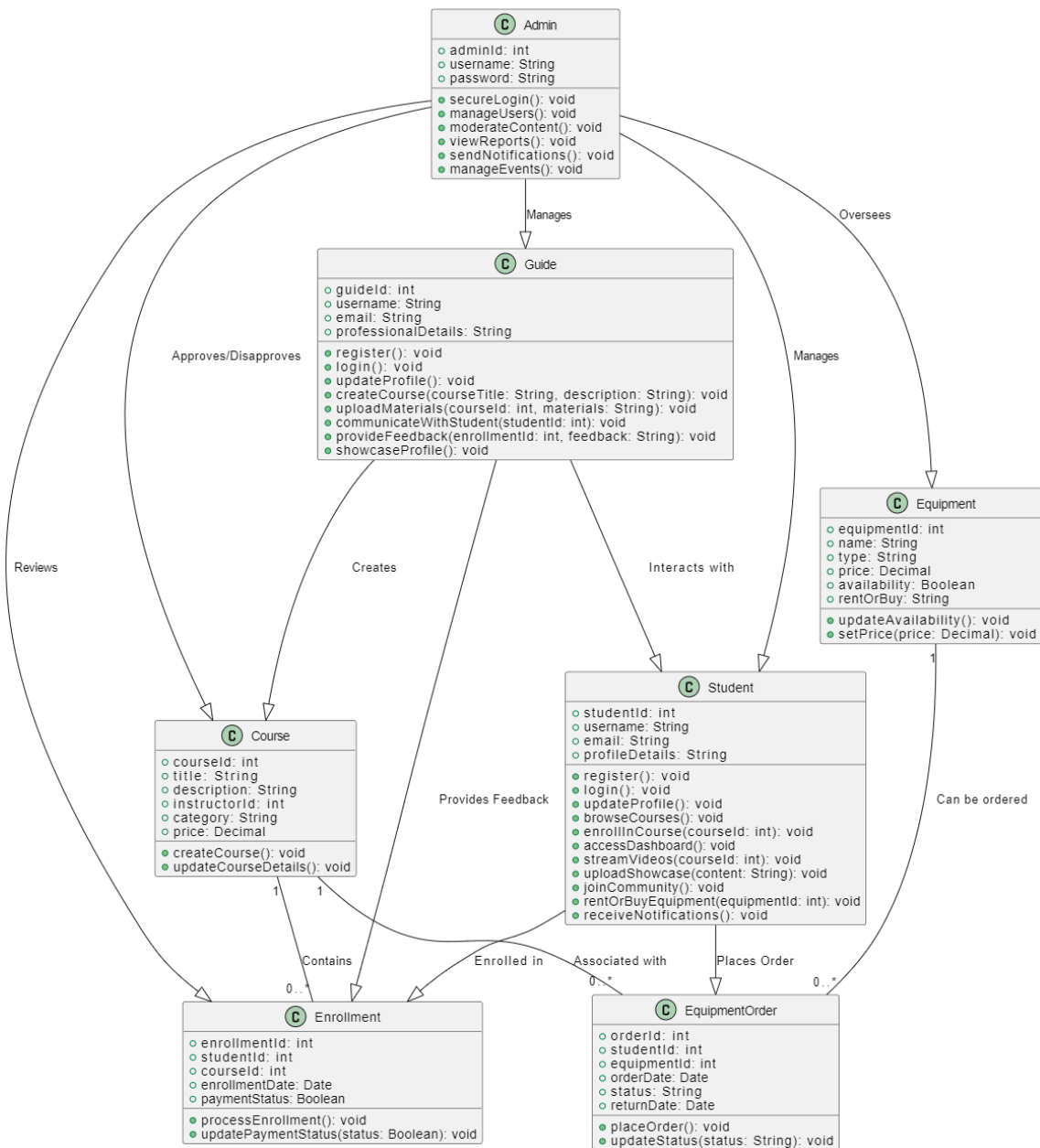


Fig 4.class diagram

4.2.5 Object Diagram

An object diagram is a type of static structure diagram in Unified Modeling Language (UML) that provides a snapshot of the instances of classes (objects) and their relationships at a specific moment in time. Unlike class diagrams, which focus on the blueprint of classes and their relationships, object diagrams emphasize the concrete examples of those classes in a particular state.

Each object is represented by a rectangle, similar to a class in a class diagram, but it includes the object's name and its current state or attribute values. Lines connecting the objects illustrate their relationships, such as associations, aggregations, or compositions. Object diagrams are particularly useful for visualizing complex relationships and interactions among objects in a system,

demonstrating how they collaborate to perform tasks or represent data.

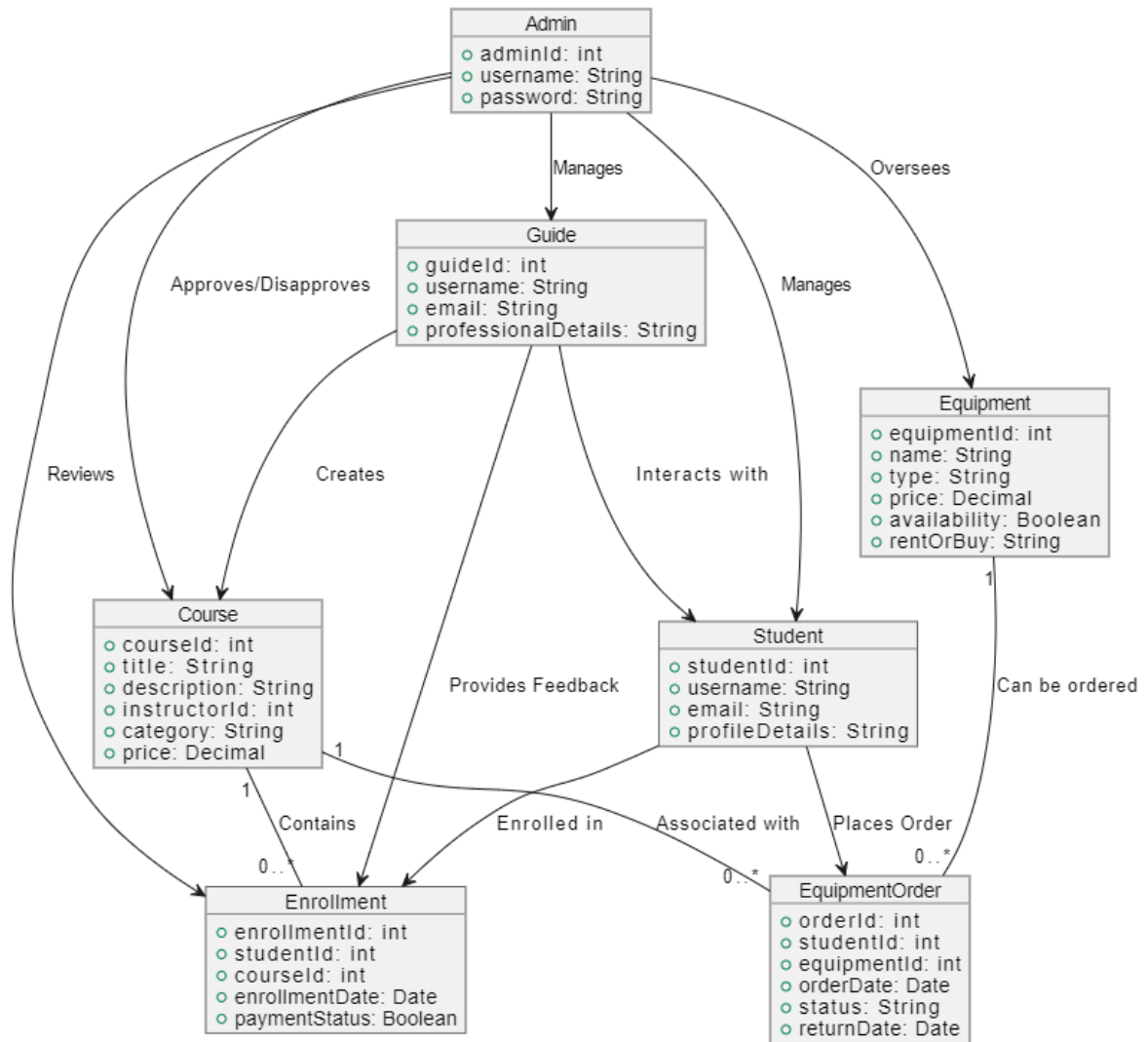


Fig 5.object diagram

4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: Index

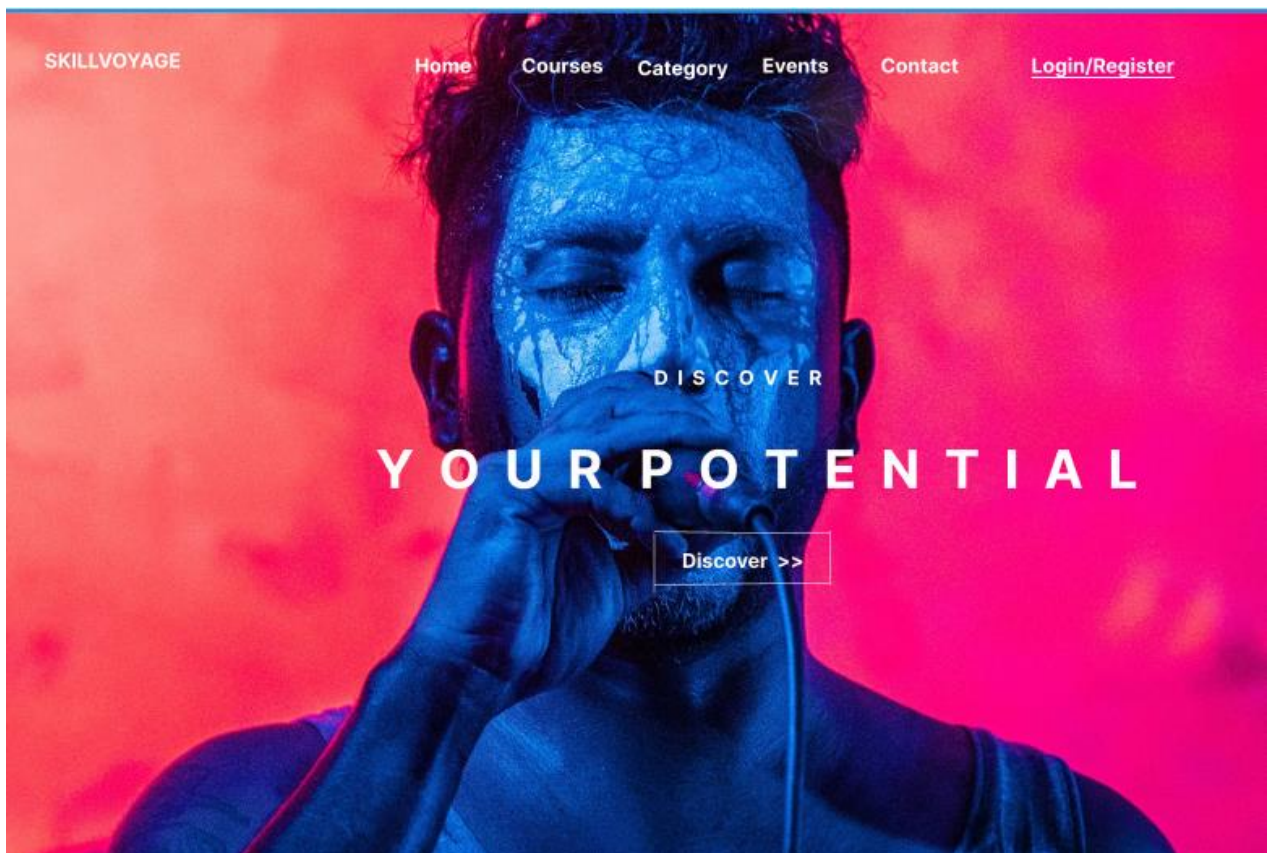


Fig 6. Home page

Form Name: LoginThe image displays the login form of the SkillVoyage website. At the top, there is a dark navigation bar with the logo 'SKILLVOYAGE' and links for 'Home', 'Courses', 'Category', 'Events', 'Contact', and 'Login/Register'. Below this, the word 'LOGIN' is centered in a light blue box. The main form area has a light blue background. It starts with the heading 'Welcome Back'. Below this, there are two input fields: 'Email Address' with a placeholder 'Enter e-mail' and 'Password' with a placeholder 'Enter password'. To the right of the password field is a link that says 'Forget Password'. At the bottom of the form is a button labeled 'LOGIN' and a link that says 'Not yet registered sign up'.

Fig 7. Login form

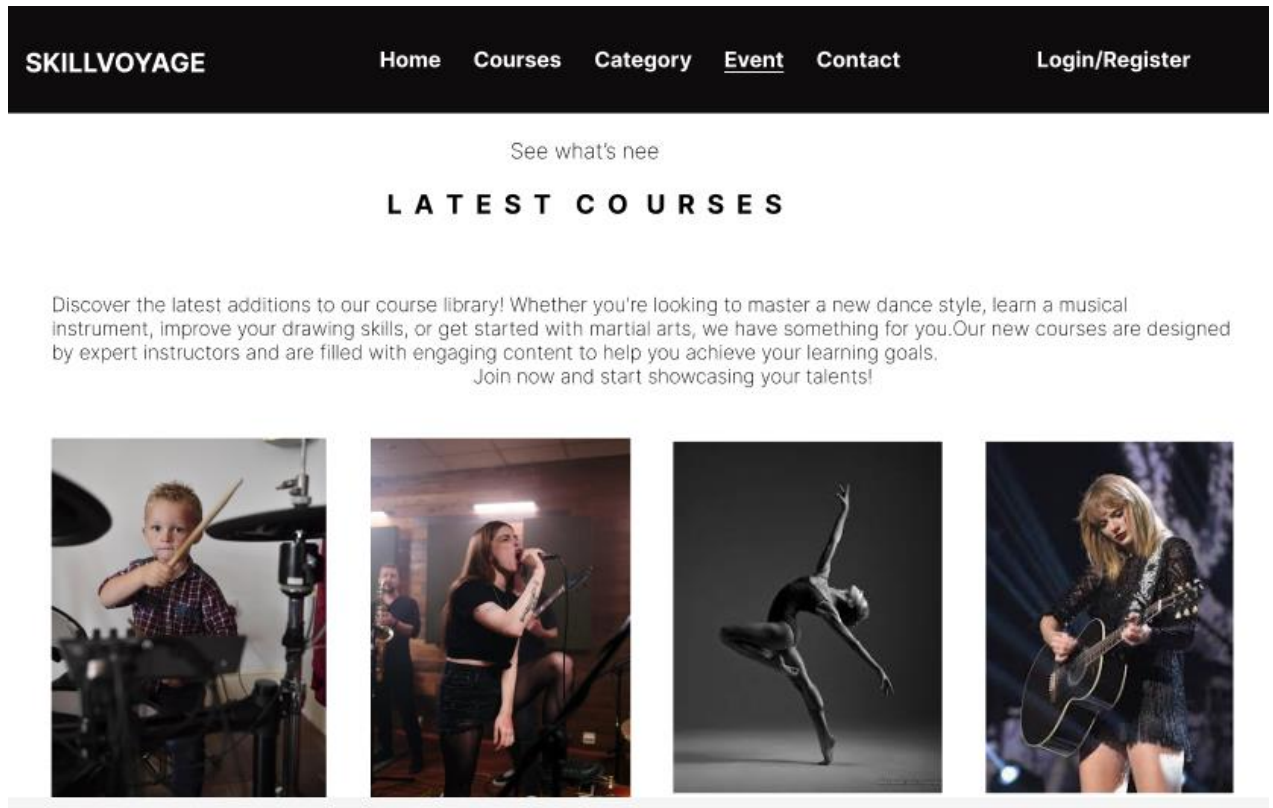
Form Name: Home Page

Fig 8. Home page

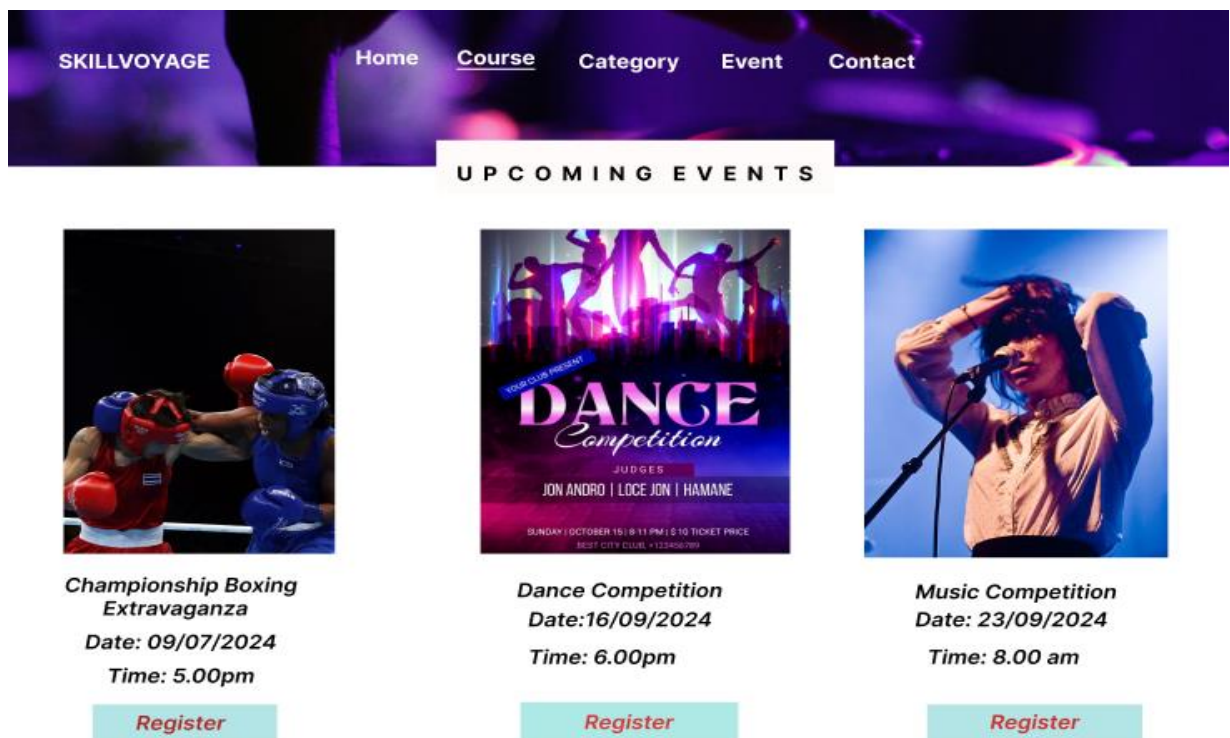
Form Name: Event Page

Fig 9. Event page

Form Name: Course Page**SKILLVOYAGE**[Home](#)[Course](#)[Category](#)[Event](#)[Contact](#)**P R I C I N G P L A N**

"Unlock your potential with our flexible and affordable pricing plans. Choose the perfect plan to access a wide range of high-quality courses and start mastering new skills today. Enroll now and embark on your journey of learning and showcasing your talents!"

Guitar

Master the guitar with our easy-to-follow lessons. Ideal for beginners and improvers, covering chords, strumming, and music theory. Start playing your favorite songs with expert guidance. Enroll today!

Price: \$99

[Enroll Now](#)**Bharatanatyam**

Explore the elegance of Bharathanatyam with our comprehensive classes. Learn classical steps, expressive storytelling, and traditional techniques. Perfect for all levels. Begin your dance journey today!

Price: \$45

[Enroll Now](#)**Drumming**

Master the art of drumming with our dynamic courses. From basic rhythms to advanced techniques, develop your drumming skills with expert guidance. Join now and start your rhythmic journey!

Price: \$50

[Enroll Now](#)

Fig 10. Course page

4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

In a relational database like SQLite, data is stored in structured tables, with each table representing a specific entity, and relationships are established between tables to maintain data integrity and consistency. Relational databases use a schema to define the structure of each table, including fields, data types, and constraints. SQLite is a lightweight RDBMS that supports SQL for managing relational data, making it a practical choice for smaller applications that still require structured data handling.

Relationships

- **Table Relationships:** Relationships in SQLite are created using keys. There are two key types that play a central role in maintaining relationships: **Primary Key** and **Foreign Key**.
- **Primary Key:** Uniquely identifies each record in a table. No two rows in a table can share the same primary key value, ensuring unique identification of each record.
- **Foreign Key:** References a Primary Key in another table, establishing a link between two tables. This link enables the database to enforce **Referential Integrity**, ensuring that each Foreign Key value corresponds to an existing Primary Key value in the related table.
- **Entity Integrity:** This principle enforces that the Primary Key of a table must always contain a unique and non-null value.
- **Referential Integrity:** This constraint ensures that the relationships between tables remain consistent. If a Foreign Key value is specified, it must match a valid Primary Key value in the referenced table.

4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys, and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in

a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

Ensures that the data in each column is atomic (indivisible), meaning that each field holds only a single value. In 1NF, each table has a Primary Key, and repeating groups or arrays are removed. This is achieved by creating separate tables for data that has a repeating nature.

Second Normal Form

Builds upon 1NF by ensuring that all non-key attributes are fully functionally dependent on the Primary Key. In cases where a Primary Key consists of multiple columns (composite key), each non-key attribute must depend on all parts of that composite key. This removes partial dependencies and creates separate tables for attributes that are only partially dependent on the Primary Key.

Third Normal Form

Extends 2NF by ensuring that no non-key attribute depends on another non-key attribute (i.e., no transitive dependencies). This form requires that non-key attributes rely directly and only on the Primary Key. The result is a structure in which all attributes are only dependent on the Primary Key, further reducing redundancy and potential anomalies.

Fourth Normal Form

Addresses multi-valued dependencies, where an attribute in a table can contain multiple independent values. By isolating these values into separate tables, 4NF reduces data redundancy and ensures each table represents a single entity or concept.

Fifth Normal Form

Also known as "Projection-Join Normal Form," 5NF removes any overlapping multi-valued dependencies between tables, decomposing complex relationships into smaller tables without data loss. This further reduces redundancy and improves data integrity, especially in cases with interdependent relationships across multiple entities.

4.4.3 Sanitization

An automated procedure called "sanitization" is used to get a value ready for use in a SQL query. This process typically involves checking the value for characters that have a special significance for the target database. To prevent a SQL injection attack, you must sanitize(filter) the input string while processing a SQL query based on user input. For instance, the user and password input is a typical scenario. In that scenario, the server response would provide access to the 'target user' account without requiring a password check.

4.4.4 Indexing

By reducing the number of disk accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes. The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order. Recall that the information may or may not be kept in sorted order.

4.5 TABLE DESIGN

1. Users Table

Field Name	Data Type (size)	Key Constraint	Description of the field
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the user
email	VARCHAR(255)	UNIQUE, NOT NULL	User's email address, serves as username for login
username	VARCHAR(255)	UNIQUE, NOT NULL	Username selected by the user
password	VARCHAR(255)	NOT NULL	Hashed password of the user
usertype	VARCHAR(10)	NOT NULL	Indicates whether the user is a "student" or "teacher"
is_staff	BOOLEAN	DEFAULT False	Indicates if the user is a staff member
is_superuser	BOOLEAN	DEFAULT False	Indicates if the user has superuser privileges
created_at	DATETIME	AUTO_GENERATED	Date and time the user was created

2. PasswordReset Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the password reset entry
user_id	INT	FOREIGN KEY (REFERENCES User(id))	ID of the user requesting password reset
token	VARCHAR(100)	UNIQUE, NOT NULL	Unique token for the password reset
created_at	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the password reset request was created

3. TeacherRequest Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the teacher request
user_id	INT	FOREIGN KEY (REFERENCES User(id))	ID of the user submitting the request
full_name	VARCHAR(100)	NOT NULL	Full name of the teacher applicant
email	VARCHAR(255)	NOT NULL	Email of the teacher applicant

phone_number	VARCHAR(20)	NOT NULL	Contact phone number of the teacher
qualification	VARCHAR(100)	NOT NULL	Qualification details of the teacher
experience_years	INT	NOT NULL	Years of experience of the teacher
skills	VARCHAR(255)	NOT NULL	Key skills possessed by the teacher
resume	FILE	NOT NULL	Resume file upload
status	VARCHAR(10)	DEFAULT 'Pending', NOT NULL	Status of the request (Pending, Accepted, Rejected)
created_at	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the request was created

4. Category Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the category

name	VARCHAR(255)	NOT NULL	Name of the category
description	TEXT	NULL	Description of the category

5. SubCategory Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the subcategory
name	VARCHAR(100)	NOT NULL	Name of the subcategory
category_id	INT	FOREIGN KEY (REFERENCES Category(id))	ID of the parent category
image	IMAGE (File Path)	NULL	Image for the subcategory

6. ContactUs Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the contact form entry
first_name	VARCHAR(255)	NOT NULL	First name of the person contacting

last_name	VARCHAR(255)	NOT NULL	Last name of the person contacting
email	VARCHAR(255)	NOT NULL	Email of the person contacting
phone_number	VARCHAR(20)	NULL	Phone number of the person contacting
message	TEXT	NOT NULL	Message content

7. Course Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the course
name	VARCHAR(255)	NOT NULL	Name of the course
short_description	VARCHAR(255)	NOT NULL	Short description of the course
description	TEXT	NOT NULL	Detailed description of the course
featured_image	IMAGE (File Path)	NOT NULL	Featured image for the course
sub_category_id	INT	FOREIGN KEY (REFERENCES SubCategory(id))	ID of the related subcategory
duration	DURATION	NOT NULL	Duration of the course

instructor_id	INT	FOREIGN KEY (REFERENCES User(id))	ID of the instructor (Teacher)
learning_outcome	TEXT	NOT NULL	Expected learning outcomes for the course
price	DECIMAL(8, 2)	NOT NULL	Price of the course
created_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the course was created
updated_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the course was last updated
status	VARCHAR(20)	DEFAULT 'draft', NOT NULL	Status of the course (Draft, Published, etc.)

8. CourseLesson Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the lesson
course_id	INT	FOREIGN KEY (REFERENCES Course(id))	ID of the related course
title	VARCHAR(255)	NOT NULL	Title of the lesson
description	TEXT	NULL	Description of the lesson

resource	FILE (File Path)	NULL	Resource file for the lesson (video, pdf, etc.)
created_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the lesson was created
updated_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the lesson was last updated

9. EnrolledCourse Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the enrolled course entry
course_id	INT	FOREIGN KEY (REFERENCES Course(id))	ID of the enrolled course
student_id	INT	FOREIGN KEY (REFERENCES User(id))	ID of the student who enrolled
created_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time of enrollment

10. Product Table

Field Name	Data Type (size)	Key Constraint	Description
------------	------------------	----------------	-------------

id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the product
name	VARCHAR(255)	NOT NULL	Name of the product
short_description	VARCHAR(255)	NOT NULL	Short description of the product
description	TEXT	NOT NULL	Detailed description of the product
price	DECIMAL(8, 2)	NOT NULL	Price of the product
image	IMAGE (File Path)	NULL	Image for the product
stock	INT	NOT NULL	Stock quantity of the product
status	VARCHAR(20)	DEFAULT 'active', NOT NULL	Status of the product (Active, Inactive)
category_id	INT	FOREIGN KEY (REFERENCES Category(id))	ID of the related category
created_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the product was created
updated_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the product was last updated

11. Cart Table

Field Name	Data Type (size)	Key Constraint	Description
id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for the cart entry
user_id	INT	FOREIGN KEY (REFERENCES User(id))	ID of the user who owns the cart
product_id	INT	FOREIGN KEY (REFERENCES Product(id))	ID of the product added to the cart
quantity	INT	NOT NULL	Quantity of the product added
created_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the cart entry was created
updated_on	DATETIME	AUTO_GENERATED, NOT NULL	Date and time the cart entry was last updated

12. Order Table

Field Name	Data Type (size)	Key Constraint	Description of the field
id	BigAutoField	Primary Key	Unique identifier for each order

user	ForeignKey (User)	Foreign Key, Cascade	Links to the user who made the order
created_on	DateTimeField	Auto add	Timestamp when the order was created
price	DecimalField (8,2)	None	Total price of the order
address	CharField (255)	None	Delivery address for the order
success	BooleanField	None	Status of order completion (True/False)

13. Order Item

Field Name	Data Type (size)	Key Constraint	Description of the field
id	BigAutoField	Primary Key	Unique identifier for each order item
order	ForeignKey (Order)	Foreign Key, Cascade	Links to the associated order
product	ForeignKey (Product)	Foreign Key, Cascade	Links to the product in the order item

quantity	PositiveIntegerField	None	Quantity of the product in the order
created_on	DateTimeField	Auto add	Timestamp when the order item was added

14. Feedback Table

Field Name	Data Type (size)	Key Constraint	Description of the field
user	ForeignKey (User)	Foreign Key, Cascade	Links to the user who gave the feedback
lesson	ForeignKey (CourseLesson)	Foreign Key, Cascade	Links to the lesson for feedback
rating	IntegerField (1-5 choices)	None	User's rating (1 to 5)
feedback	TextField	None	Feedback content by the user
created_at	DateTimeField	Auto add	Timestamp when the feedback was given

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met. There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation, and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is Whitebox oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at

its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is a systematic technique for constructing the program structure while simultaneously conducting tests to uncover errors associated with interfacing (Myers et al., 2011) [7]. The objective is to take unit-tested components and build a cohesive program structure as dictated by the design (Burnstein, 2003) [8]. The entire program is tested as a whole, though correcting issues can be challenging because isolating causes is complicated by the program's expanse (Fewster & Graham, 1999) [9]. After performing unit testing on individual modules, all components were integrated to check for inconsistencies in their interfaces (Myers et al., 2011) [7]. Differences in program structures were addressed, evolving toward a unified program structure [8].

This is the final step in testing. In this the entire system was tested with all forms, code, modules, and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.3 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs.
- Output Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.4 Automation Testing

A test case suite is executed using specialized automated testing software tools as part of the software testing technique known as automation testing (Fewster & Graham, 1999) [9]. While manual testing involves meticulously executing test stages by a human seated in front of a computer, automation testing leverages software to perform these tasks, reducing human intervention and enhancing efficiency (Raj & Kumar, 2019) [11]. Additionally, automation testing tools can generate detailed test reports, compare expected and actual outcomes, and automatically input test data into the System Under Test (SUT) (Molyneaux, 2009) [10]. However, effective software test automation requires significant financial and material investments, often justified by the long-term benefits of scalability and repeatability [9].

5.2.5 Selenium Testing

Selenium is a free and open-source tool for testing web applications across multiple browsers and operating systems. Selenium Test Scripts can be written in different programming languages, including Java, C#, JavaScript, Python, etc. Automation performed using the Selenium framework is referred to as Selenium Automation testing.

Example:

Test Case 1

Code

```
myProject > selenium_test > login selenium.py > ...
1  from selenium import webdriver
2  from selenium.webdriver.edge.service import Service
3  from webdriver_manager.microsoft import EdgeChromiumDriverManager # Use EdgeChromiumDriverManager
4  from selenium.webdriver.common.by import By
5  from selenium.webdriver.support.ui import WebDriverWait
6  from selenium.webdriver.support import expected_conditions as EC
7  import time
8  import easygui
9
10 # Initialize EdgeDriver using webdriver-manager
11 driver = webdriver.Edge(service=Service(EdgeChromiumDriverManager().install()))
12
13 # Open the Login page
14 driver.get("http://127.0.0.1:8000/login/")
15 print("Opened the login page.")
16
17 # Maximize the browser window
18 driver.maximize_window()
19
20 # Wait for the page to load completely
21 time.sleep(3)
22
23 # Wait for the email field to be visible and interact with it
24 wait = WebDriverWait(driver, 10)
25
26 # Step 1: Log in with valid credentials
27 email_field = wait.until(EC.visibility_of_element_located((By.NAME, "email")))
28 email_field.clear()
29 email_field.send_keys('admin@gmail.com') # Use the appropriate email
30 print("Entered email.")
31 time.sleep(1)
32
```

Eg.Screenshot



Fig 11. Login success testing

Eg. Test Report

Test Case 1	
Project Name: SkillVoyage	
Login Test Case	
Test Case ID: Test_1	Test Designed By: Biya Aby
Test Priority (Low/Medium/High):	Test Designed Date: 20/10/2024

Module Name: Login Page			Test Executed By: Merin Chacko		
Test Title: Successful Login to User Account			Test Execution Date: 22/10/2024		
Description: Test the successful login functionality with valid user credentials.					
Pre-Condition: User must have valid login credentials (username and password).					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to the Login Page	http://127.0.0.1:8000/login/	Login page should be displayed	Login page displayed	Pass
2	Enter email	“admin@gmail.com”	The email field should be filled with admin@gmail.com	Email entered	Pass
3	Enter password	“admin”	The password field should be filled with admin	Password entered	Pass
4	Click on the submit button	N/A	User should be redirected to the user dashboard page if login is successful	User redirected to dashboard	Pass
Post-Condition: The system verifies the email and password against stored values in the database. If they match, the login is successful.					

Test Case 2:

Code:

```

myProject > selenium_test > login selenium.py > ...
1  from selenium import webdriver
2  from selenium.webdriver.edge.service import Service
3  from webdriver_manager.microsoft import EdgeChromiumDriverManager # Use EdgeChromiumDriverManager
4  from selenium.webdriver.common.by import By
5  from selenium.webdriver.support.ui import WebDriverWait
6  from selenium.webdriver.support import expected_conditions as EC
7  import time
8  import easygui
9
10 # Initialize EdgeDriver using webdriver-manager
11 driver = webdriver.Edge(service=Service(EdgeChromiumDriverManager().install()))
12
13 # Open the Login page
14 driver.get("http://127.0.0.1:8000/login/")
15 print("Opened the login page.")
16
17 # Maximize the browser window
18 driver.maximize_window()
19
20 # Wait for the page to load completely
21 time.sleep(3)
22
23 # Wait for the email field to be visible and interact with it
24 wait = WebDriverWait(driver, 10)
25
26 # Step 1: Log in with valid credentials
27 email_field = wait.until(EC.visibility_of_element_located((By.NAME, "email")))
28 email_field.clear()
29 email_field.send_keys('admin@gmail.com') # Use the appropriate email
30 print("Entered email.")
31 time.sleep(1)
32

```

Screenshot:

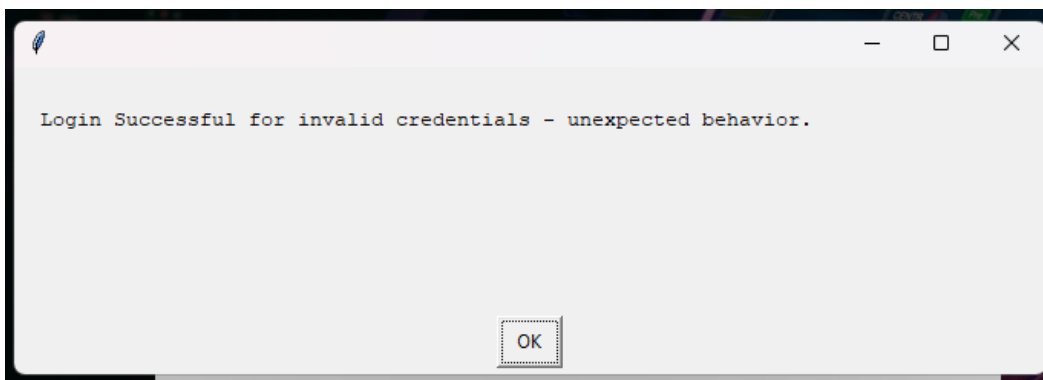


Fig 12. Login fail test

Test Case 2

Project Name: SkillVoyage

Login Test Case

Test Case ID: Test_2

Test Designed By: Biya Aby

Test Priority (Low/Medium/High):

Test Designed Date: 20/10/2024

Module Name: Login Page

Test Executed By: Merin Chacko

Test Title: Unsuccessful Login with Invalid Credentials			Test Execution Date: 22/10/2024		
Description: Test the login functionality with invalid user credentials.					
Pre-Condition: User attempts login with incorrect credentials.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to the login page	http://127.0.0.1:8000/login/	Login page should be displayed	Login page displayed	Pass
2	Enter invalid email	invalid_user@gmail.com	The email field should be filled with invalid_user@gmail.com	Invalid email entered	Pass
3	Enter invalid password	wrongpassword	The password field should be filled with wrongpassword	Invalid password entered	Pass
4	Click on the submit button	N/A	An error message should appear indicating login failure	Error message displayed	Pass
Post-Condition: The system checks the provided credentials against stored values. Since they do not match, the login fails and an error message is displayed.					

Test Case 3:

Code:

```

1  from selenium import webdriver
2  from selenium.webdriver.edge.service import Service
3  from webdriver_manager.microsoft import EdgeChromiumDriverManager
4  from selenium.webdriver.common.by import By
5  from selenium.webdriver.support.ui import WebDriverWait
6  from selenium.webdriver.support import expected_conditions as EC
7  import time
8  import easygui
9
10 # Initialize EdgeDriver with SSL bypass options
11 options = webdriver.EdgeOptions()
12 options.add_argument('--ignore-certificate-errors')
13 options.add_argument('--allow-insecure-localhost') # Allow insecure localhost connections
14 driver = webdriver.Edge(service=Service(EdgeChromiumDriverManager().install()), options=options)
15
16 # Open the login page
17 driver.get("http://127.0.0.1:8000/login/")
18 driver.maximize_window()
19
20 # Wait for the page to load completely
21 time.sleep(3)
22
23 # Log in with valid credentials
24 wait = WebDriverWait(driver, 10)
25 email_field = wait.until(EC.visibility_of_element_located((By.NAME, "email")))
26 email_field.clear()

```

Screenshot:

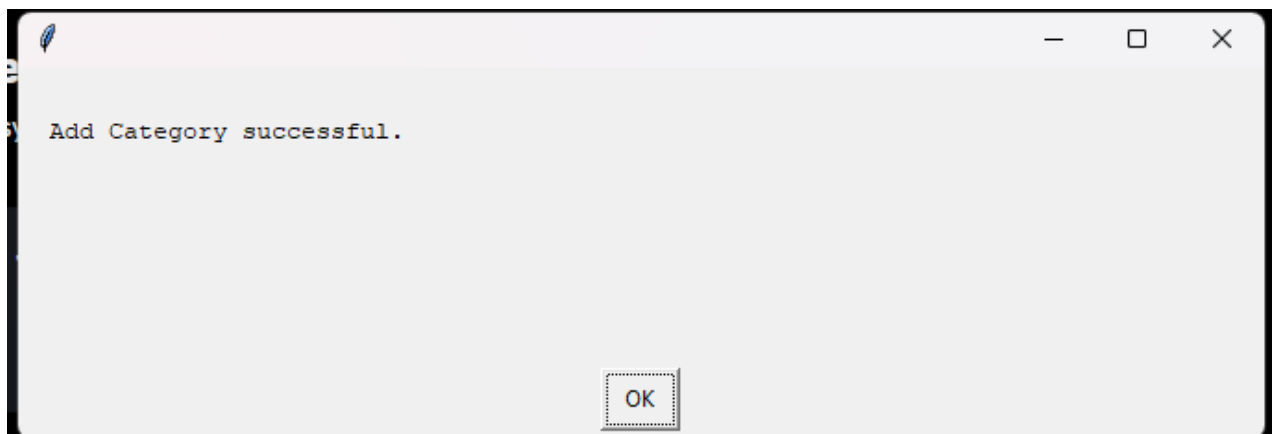


Fig 13. Category added

Test Case 3	
Project Name: SkillVoyage	
Category Management Test Case	
Test Case ID: Test_3	Test Designed By: Biya Aby
Test Priority (Low/Medium/High):	Test Designed Date: 20/10/2024
Module Name: Category Managemen	Test Executed By: Merin Chacko

Test Title: Add a new category			Test Execution Date: 22/10/2024		
Description: Test to add a new category in the application.					
Pre-Condition: User is logged in with valid credentials and has navigated to the Add Category page.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to the Add Category page	http://127.0.0.1:8000/add-category/	Add Category page should load successfully	Add Category page loaded	Pass
2	Enter category name	New Category	The category name field should be filled with New Category	Category name entered	Pass
3	Enter category description (if available)	This is a test category description.	The description field should be filled with This is a test category description.	Category description entered	Pass
4	Click the submit button	N/A	The category should be added and a success message should appear	Success message displayed	Pass
Post-Condition: The new category is added to the system, and a confirmation message is shown on the UI.					

Test Case 4:

Code:

```

try:
    description_field = driver.find_element(By.NAME, 'description') # Replace 'description' with the actual f
    description_field.clear()
    description_field.send_keys('This is a test category description.')
    print("Entered category description.")
    time.sleep(1)
except Exception as e:
    print("Description field not found:", e)

# Click the submit button
submit_button = wait.until(EC.element_to_be_clickable((By.CSS_SELECTOR, 'button[type="submit"]')))
submit_button.click()
print("Clicked the submit button to add the category.")

# Wait for processing and then check for the success message
time.sleep(3)

# Check for success message
try:
    success_message = wait.until(EC.visibility_of_element_located((By.CSS_SELECTOR, '.alert.alert-success')))
    if success_message.is_displayed():
        easygui.msgbox("Category added successfully!")
        print("Category added successfully.")
except Exception as e:
    print("Success message not found:", e)
    print("Page source at the time of failure:", driver.page_source)

except Exception as e:
    print("Error while trying to add a category:", e)
    print("Page source at the time of error:", driver.page_source)

```

Screenshot:

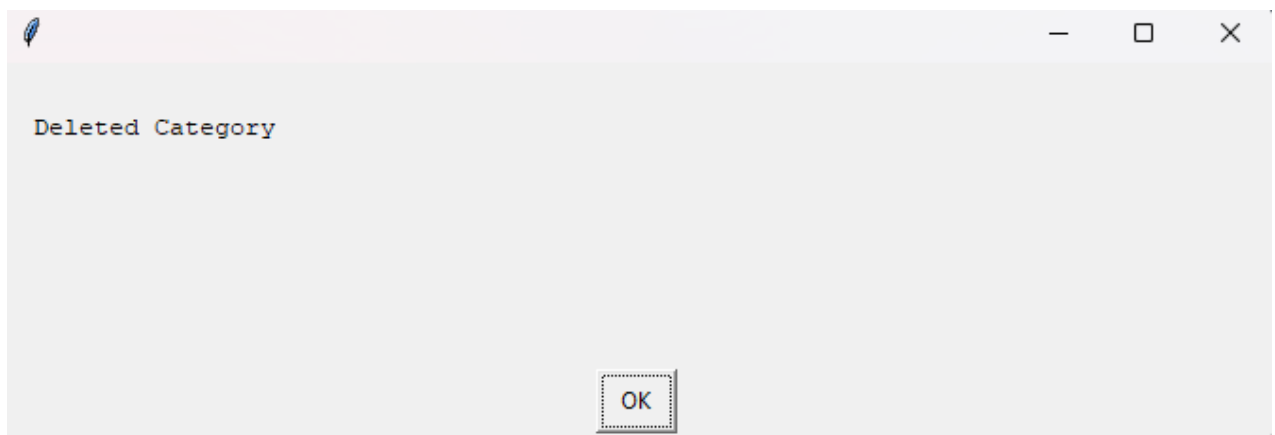


Fig 14. Category deleted

Test Case 4

Project Name: SkillVoyage

Category Management Test Case

Test Case ID: Test_4

Test Designed By: Biya Aby

Test Priority (Low/Medium/High):

Test Designed Date: 20/10/2024

Module Name: Category

Test Executed By: Merin Chacko

Management					
Test Title: Delete an existing category			Test Execution Date: 22/10/2024		
Description: Test to delete a specific category from the system.					
Pre-Condition: User is logged in with valid credentials, and the category to be deleted exists in the list of categories.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigate to the category management page	http://127.0.0.1:8000/categories /	Category list page should load successfully	Category list page loaded	Pass
2	Locate the delete button for the target category	New Category	Delete button for New Category should be visible	Delete button found	Pass
3	Click the delete button and confirm action	N/A	The category should be deleted, and a confirmation message should appear	Confirmation message shown	Pass
Post-Condition: The selected category is removed from the system, and a deletion confirmation message is displayed on the UI.					

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

- Careful planning
- Investigation of system and constraints
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we must ensure that the resistance does not build up, as one must make sure that:

- The active user must be aware of the benefits of using the new system. Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process will not take place

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database, and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.3 System Maintenance

System maintenance is an effective component such that System maintenance refers to the ongoing activities required to ensure that a system or application operates effectively and efficiently after it has been implemented. It involves regular updates, bug fixes, and performance optimizations to keep the system running smoothly and securely.

System maintenance is essential to ensure that a system remains operational and effective after implementation. By establishing maintenance procedures and following them consistently, project teams can ensure that the system operates smoothly, remains secure, and continues to meet the needs of the end-users.

6.2.4 Hosting

Hosting allows a website to be accessible online by storing its files on a server. Different hosting types cater to various needs: shared hosting is budget-friendly, VPS offers more control, dedicated hosting provides full server access, and cloud hosting improves scalability. Managed hosting handles technical tasks for you, while self-hosting allows complete control for tech-savvy users. Good hosting ensures website reliability, speed, and security, impacting user experience and SEO.

Render Hosting

Render hosting often includes features like server-side rendering (SSR), static site generation (SSG), and content delivery networks (CDNs) to optimize load times and improve scalability. By pre-rendering pages or using SSR, the server sends fully constructed HTML to users, boosting speed and SEO performance.

Procedure for hosting a website on 000Webhost:

Step 1: Set up a Render Account

- Visit Render's website and create an account or log in if you already have one.
- Once logged in, go to the Dashboard.

Step 2: Link GitHub Repository

- In the Render Dashboard, click on the New button and select Web Service.
- Connect your GitHub account to Render if you haven't already. This will allow Render to access your repositories.
- After linking, select the SkillVoyage repository from the list of repositories.

Step 3: Configure Deployment Settings

- Branch Selection: Choose the branch you want to deploy (usually main or master).
- Environment Settings: Set up the following configurations:
 1. Runtime: Choose Python 3.x.
 2. Build Command: Set it to `pip install -r requirements.txt` to ensure all dependencies are installed.
 3. Start Command: Use `gunicorn SkillVoyage.wsgi:application` (replace SkillVoyage with the name of your Django project directory if different).
- Environment Variables:
 1. Add any necessary environment variables, such as `DJANGO_SECRET_KEY`, `DATABASE_URL` (for PostgreSQL if applicable), and other settings from your `settings.py` file.
 2. Make sure `ALLOWED_HOSTS` includes `render.com` or the specific URL

Render provides for your app.

Step 4: Set up the Database (SQLite or PostgreSQL)

- SQLite Database: Since SQLite is file-based, ensure that your SQLite database file (db.sqlite3) is part of your repository.
- Migrate Database: If using PostgreSQL or other external databases:
 1. Ensure migrations are run after deployment by setting the post-deploy command to `python manage.py migrate`.

Step 5: Configure Static Files

- Django Settings for Static Files:
 1. Configure settings.py to use whitenoise for serving static files in production.
 2. Install whitenoise in your project and add 'whitenoise.middleware.WhiteNoiseMiddleware' to MIDDLEWARE in settings.py.
- Collect Static Files:
 1. Add `python manage.py collectstatic` to the Build Command section.

Step 6: Deploy the Project

- Once all settings are configured, click on Create Web Service.
- Render will build the project, install dependencies, run migrations (if configured), and start the service.
- After deployment, Render will provide a URL where your project will be live.

Hosted Link: <https://skillvoyage.onrender.com/>

Hosted QR Code:



Screenshot

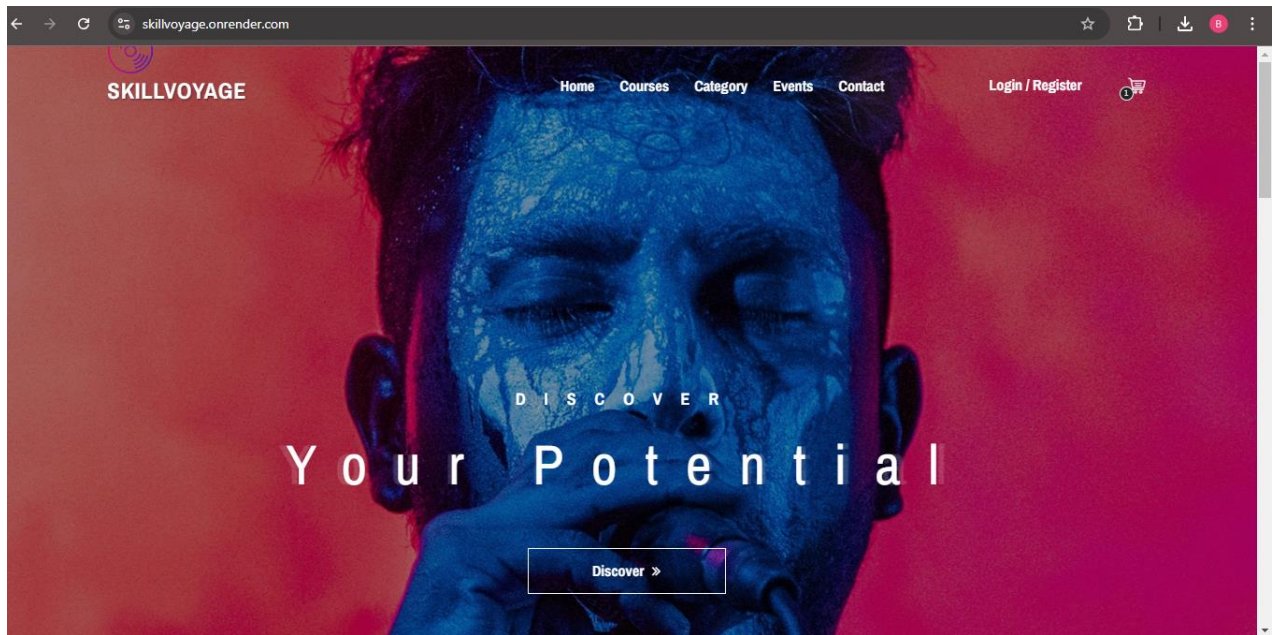


Fig 15. Hosted page

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The SkillVoyage project serves as an innovative platform that empowers learners to discover and develop various talents—ranging from dance and music to drawing and martial arts. By providing a structured yet flexible space for both learning and showcasing skills, SkillVoyage bridges the gap between aspiring learners and expert instructors. Through its user-friendly interface and streamlined course management tools, SkillVoyage enhances the educational journey, making it accessible and engaging for users of all backgrounds. With features that foster personal growth, community engagement, and skill recognition, SkillVoyage is poised to support a diverse and vibrant ecosystem where creativity and knowledge flourish.

7.2 FUTURE SCOPE

The future of the SkillVoyage platform holds exciting possibilities, with potential to integrate advanced machine learning models for personalized learning recommendations and skill-based matching between learners and instructors. Expanding multilingual support could allow SkillVoyage to reach a broader audience, embracing users from diverse regions and backgrounds. Additional digital literacy resources could empower users to make the most of the platform's tools. As online education continues to evolve, SkillVoyage could incorporate immersive technologies like virtual and augmented reality to enhance the learning experience for hands-on skills. The platform might also add gamified elements, helping to motivate and engage users in their skill development journey. With these enhancements, SkillVoyage aims to remain at the forefront of digital learning, fostering a dynamic environment for learners and instructors alike.

.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

1. Vinayak, R., & Mehta, A. (2017). *Django Design Patterns and Best Practices*. Packt Publishing.
2. Zhang, J., Li, X., & Wu, T. (2023). *Deep Learning for Art Style Classification and Description*. IEEE Transactions on Multimedia.
3. Singh, R., & Patel, A. (2024). *Machine Learning Techniques for Automated Art Description*. Journal of Artificial Intelligence Research.
4. Garcia, M., & Lee, S. (2023). *Multimodal Approaches in Art Recognition: Combining Visual and Textual Data*. International Journal of Computer Vision.
5. Martinez, F., & Chen, Z. (2024). *Explainable AI in Art Recognition: Techniques and Applications*. ACM Computing Surveys.
6. Brown, L., & Sato, K. (2022). *Exploring Reinforcement Learning for Art Curation and Personalization*. Artificial Intelligence Review.
7. Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing*. John Wiley & Sons.
8. Burnstein, I. (2003). *Practical Software Testing: A Process-Oriented Approach*. Springer.
9. Fewster, M., & Graham, D. (1999). *Software Test Automation: Effective Use of Test Execution Tools*. Addison-Wesley.
10. Molyneaux, I. (2009). *The Art of Application Performance Testing*. O'Reilly Media.
11. Raj, A., & Kumar, V. (2019). *Selenium Essentials: A Guide to Web Application Testing*. Packt Publishing.

WEBSITES:

- <https://www.djangoproject.com/>
- <https://pytorch.org/>
- <https://www.nngroup.com/articles/designing-for-digital-learning/>

CHAPTER 9

APPENDIX

9.1 Sample Code

Registration Page

```
{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="description" content="">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <title>SkillVoyage - Register</title>
    <link rel="icon" href="{% static 'img/core-img/favicon.ico' %}">
    <link rel="stylesheet" href="{% static 'style.css' %}">
    <style>
        .error-message {
            color: red;
            font-size: 0.875em;
            margin-top: 0.25em;
        }
    </style>
</head>

<body>
    <!-- PreLoader -->
    <div class="preloader d-flex align-items-center justify-content-center">
        <div class="lds-ellipsis">
            <div></div>
            <div></div>
            <div></div>
            <div></div>
        </div>
    </div>

    <!-- Header Area Start -->
    <header class="header-area">
        <!-- Navbar Area -->
        <div class="oneMusic-main-menu">
            <div class="classy-nav-container breakpoint-off">
                <div class="container">
                    <nav class="classy-navbar justify-content-between"
id="oneMusicNav">
                        <a href="{% url 'home' %}" class="nav-
brand">SkillVoyage</a>
                        <div class="classy-navbar-toggler">
                            <span
class="navbarToggler"><span></span><span></span><span></span><span></span></span></span>
```

```
</div>
<div class="classy-menu">
    <div class="classycloseIcon">
        <div class="cross-wrap"><span
class="top"></span><span class="bottom"></span></div>
        </div>
        <div class="classynav">
            <ul>
                <li><a href="{% url 'home' %}">Home</a></li>
                <li><a href="{% url 'courses'
%}">Courses</a></li>
                <li><a href="#">Categories</a>
                    <ul class="dropdown">
                        <li><a href="{% url 'courses'
%}">Music</a></li>
                        <li><a href="{% url 'courses'
%}">Dance</a></li>
                        <li><a href="{% url 'courses'
%}">Drawing</a></li>
                        <li><a href="{% url 'courses'
%}">Karate</a></li>
                    </ul>
                </li>
                <li><a href="{% url 'events'
%}">Events</a></li>
                <li><a href="{% url 'contact'
%}">Contact</a></li>
            </ul>
            <div class="login-register-cart-button d-flex
align-items-center">
                <div class="login-register-btn mr-50">
                    <a href="{% url 'login' %}"
id="loginBtn">Login / Register</a>
                </div>
                <div class="cart-btn">
                    <p><span class="icon-shopping-
cart"></span> <span class="quantity">1</span></p>
                </div>
            </div>
        </div>
    </nav>
</div>
</div>
</header>
<!-- Header Area End -->

<!-- Breadcrumb Area Start -->
```

```

    <section class="breadcumb-area bg-img bg-overlay" style="background-image:
url({% static 'img/bg-img/breadcumb3.jpg' %});">
        <div class="bradcumbContent">
            <p>Join Us</p>
            <h2>Register</h2>
        </div>
    </section>
    <!-- Breadcumb Area End -->

    <!-- Register Area Start -->
    <section class="register-area section-padding-100">
        <div class="container">
            <div class="row justify-content-center">
                <div class="col-12 col-lg-8">
                    <div class="register-content">
                        <h3>Create Your Account</h3>
                        <!-- Register Form -->
                        <div class="register-form">
                            <form id="registerForm" action="{% url 'register' %}"
method="post">
                                {% csrf_token %}
                                <div class="form-group">
                                    <label for="username">Username</label>
                                    <input type="text" class="form-control"
id="username" name="username" placeholder="Enter Username" required>
                                    <div id="usernameError" class="error-
message"></div>
                                </div>
                                <div class="form-group">
                                    <label for="email">Email address</label>
                                    <input type="email" class="form-control"
id="email" name="email" placeholder="Enter E-mail" required>
                                    <small id="emailHelp" class="form-text text-
muted"><i class="fa fa-lock mr-2"></i>We'll never share your email with anyone
else.</small>
                                    <div id="emailError" class="error-
message"></div>
                                </div>
                                <div class="form-group">
                                    <label for="password">Password</label>
                                    <input type="password" class="form-control"
id="password" name="password" placeholder="Password" required>
                                    <div id="passwordError" class="error-
message"></div>
                                </div>
                                <div class="form-group">
                                    <label for="usertype">User Type</label>
                                    <select class="form-control" id="usertype"
name="usertype" required>

```



```
<!-- ALL Javascript Script -->
<script src="{% static 'js/jquery/jquery-2.2.4.min.js' %}"></script>
<script src="{% static 'js/bootstrap/popper.min.js' %}"></script>
<script src="{% static 'js/bootstrap/bootstrap.min.js' %}"></script>
<script src="{% static 'js/plugins/plugins.js' %}"></script>
<script src="{% static 'js/active.js' %}"></script>

<!-- Custom JavaScript for form validation -->
<script>
    document.getElementById('registerForm').addEventListener('submit',
function(event) {
    let isValid = true;

    // Username validation
    const username = document.getElementById('username').value;
    const usernameError = document.getElementById('usernameError');
    if (username.length < 6 || /\d/.test(username)) {
        usernameError.textContent = 'Username must be at least 6
characters long and contain no numbers';
        isValid = false;
    } else {
        usernameError.textContent = '';
    }

    // Email validation
    const email = document.getElementById('email').value;
    const emailError = document.getElementById('emailError');
    const emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailPattern.test(email)) {
        emailError.textContent = 'Please enter a valid email address';
        isValid = false;
    } else {
        emailError.textContent = '';
    }

    // Password validation
    const password = document.getElementById('password').value;
    const passwordError = document.getElementById('passwordError');
    if (password.length < 6) {
        passwordError.textContent = 'Password must be at least 6
characters long';
        isValid = false;
    } else {
        passwordError.textContent = '';
    }

    // User type validation
    const usertype = document.getElementById('usertype').value;
```

```

        const usertypeError = document.getElementById('usertypeError');
        if (usertype === '') {
            usertypeError.textContent = 'Please select a user type';
            isValid = false;
        } else {
            usertypeError.textContent = '';
        }

        // If any validation fails, prevent form submission
        if (!isValid) {
            event.preventDefault();
        }
    });
</script>
</body>

</html>

```

Login Page

```

{% load static %}
{% load socialaccount %}
{% providers_media_js %}
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="description" content="">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <!-- The above 4 meta tags *must* come first in the head; any other head
content must come *after* these tags -->

    <!-- Title -->
    <title>SkillVoyage - Login</title>

    <!-- Favicon -->
    <link rel="icon" href="{% static 'img/core-img/favicon.ico' %}">

    <!-- Stylesheet -->
    <link rel="stylesheet" href="{% static 'style.css' %}">

    <style>
        .login-with-google-btn {
            transition: background-color .3s, box-shadow .3s;
            padding: 12px 16px 12px 42px;

```



```

        border: none;
        border-radius: 3px;
        box-shadow: 0 -1px 0 rgba(0, 0, 0, .04), 0 1px 1px rgba(0, 0, 0, .25);
        color: #757575;
        font-size: 14px;
        font-weight: 500;
        font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
Oxygen, Ubuntu, Cantarell, "Fira Sans", "Droid Sans", "Helvetica Neue", sans-
serif;

        background-color: white;
        background-image: url('{% static "img/google-logo.png" %}');
        background-repeat: no-repeat;
        background-position: 12px 11px;
        background-size: 20px;
        text-align: center;
        cursor: pointer;
    }

    .login-with-google-btn:hover {
        box-shadow: 0 -1px 0 rgba(0, 0, 0, .04), 0 2px 4px rgba(0, 0, 0, .25);
    }

    .login-with-google-btn:active {
        background-color: #eeeeee;
    }

    .login-with-google-btn:focus {
        outline: none;
        box-shadow: 0 -1px 0 rgba(0, 0, 0, .04), 0 2px 4px rgba(0, 0, 0, .25),
0 0 0 3px #c8dafc;
    }

    .login-with-google-btn:disabled {
        filter: grayscale(100%);
        background-color: #ebebeb;
        box-shadow: 0 -1px 0 rgba(0, 0, 0, .04), 0 1px 1px rgba(0, 0, 0, .25);
        cursor: not-allowed;
    }

    body {
        text-align: center;
        padding-top: 2rem;
    }
</style>

</head>

<body>

```

```

<!-- PreLoader -->
<div class="preloader d-flex align-items-center justify-content-center">
  <div class="lds-ellipsis">
    <div></div>
    <div></div>
    <div></div>
    <div></div>
  </div>
</div>

<!-- ##### Header Area Start ##### -->
<header class="header-area">
  <!-- Navbar Area -->
  <div class="oneMusic-main-menu">
    <div class="classy-nav-container breakpoint-off">
      <div class="container">
        <!-- Menu -->
        <nav class="classy-navbar justify-content-between"
id="oneMusicNav">

          <!-- Nav brand -->
          <a href="{% url 'home' %}" class="nav-
brand">SkillVoyage</a>

          <!-- Navbar Toggler -->
          <div class="classy-navbar-toggler">
            <span
class="navbarToggler"><span></span><span></span><span></span></span>
            </div>

          <!-- Menu -->
          <div class="classy-menu">

            <!-- Close Button -->
            <div class="classycloseIcon">
              <div class="cross-wrap"><span
class="top"></span><span class="bottom"></span></div>
              </div>

            <!-- Nav Start -->
            <div class="classynav">
              <ul>
                <li><a href="{% url 'home' %}">Home</a></li>
                <li><a href="{% url 'courses'
%}">Courses</a></li>

                <li><a href="#">Categories</a>
                  <ul class="dropdown">
                    <li><a href="{% url 'courses'
%}">Music</a></li>

```

```

%}">Dance</a></li>
%}">Drawing</a></li>
%}">Karate</a></li>
</ul>
</li>
<li><a href="{% url 'events'
%}">Events</a></li>
<li><a href="{% url 'contact'
%}">Contact</a></li>
</ul>

<!-- Login/Register & Cart Button -->
<div class="login-register-button d-flex
align-items-center">
    <!-- Login/Register -->
    <div class="login-register-btn mr-50">
        <a href="{% url 'login' %}"
id="loginBtn">Login / Register</a>
    </div>

    <!-- Cart Button -->
    <div class="cart-btn">
        <p><span class="icon-shopping-
cart"></span> <span class="quantity">1</span></p>
    </div>
</div>
<!-- Nav End -->

</div>
</nav>
</div>
</div>
</div>
</header>
<!-- ##### Header Area End ##### -->

<!-- ##### Breadcrumb Area Start ##### -->
<section class="breadcrumb-area bg-img bg-overlay" style="background-image:
url({% static 'img/bg-img/breadcrumb3.jpg' %});">
    <div class="bradcumbContent">
        <p>See what's new</p>
        <h2>Login</h2>
    </div>
</section>
<!-- ##### Breadcrumb Area End ##### -->

```

```

<!-- ##### Login Area Start ##### -->
<section class="login-area section-padding-100">
  <div class="container">
    <div class="row justify-content-center">
      <div class="col-12 col-lg-8">
        <div class="login-content">
          <h3>Welcome Back</h3>
          <!-- Login Form -->
          <div class="login-form">
            <form action="{% url 'login' %}" method="post">
              {% csrf_token %}
              <div class="form-group">
                <label for="exampleInputEmail1">Email
address</label>
                <input type="email" class="form-control"
id="exampleInputEmail1" name="email" aria-describedby="emailHelp"
placeholder="Enter E-mail">
                <small id="emailHelp" class="form-text text-muted"><i class="fa fa-lock mr-2"></i>We'll never share your email with anyone
else.</small>
              </div>
              <div class="form-group">
                <label
for="exampleInputPassword1">Password</label>
                <input type="password" class="form-control"
id="exampleInputPassword1" name="password" placeholder="Password">
              </div>
              <button type="submit" class="btn oneMusic-btn mt-
30">Login</button>
            </form>
          </div>
          <!-- Forgot Password Link -->
          <div class="forgot-password-link mt-3">
            <p><a href="{% url 'password_reset' %}">Forgot
Password?</a></p>
          </div>
          <!-- Google Sign-In Button -->
          <div class="form sign-in mt-3">
            <a href="{% provider_login_url 'google' %}">
              <button type="button" class="login-with-google-
btn">
                Sign in with Google
              </button>
            </a>
          </div>
          <!-- Sign Up Link -->

```

```
<div class="signup-link mt-3">
    <p>Haven't you registered? <a href="{% url 'register'
%}">Sign Up</a></p>
</div>
</div>
</div>
</div>
</div>
</section>
<!-- ##### Login Area End ##### -->

<!-- ##### Footer Area Start ##### -->
<footer class="footer-area">
    <div class="container">
        <div class="row d-flex flex-wrap align-items-center">
            <div class="col-12 col-md-6">
                <p class="copywrite-text"><a href="#"><!-- Link back to
Colorlib can't be removed. Template is licensed under CC BY 3.0. -->
<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0.
--></p>
            </div>
            <div class="col-12 col-md-6">
                <div class="footer-social-info text-right">
                    <a href="#"><i class="fa fa-pinterest" aria-
hidden="true"></i></a>
                    <a href="#"><i class="fa fa-facebook" aria-
hidden="true"></i></a>
                    <a href="#"><i class="fa fa-twitter" aria-
hidden="true"></i></a>
                    <a href="#"><i class="fa fa-dribbble" aria-
hidden="true"></i></a>
                    <a href="#"><i class="fa fa-behance" aria-
hidden="true"></i></a>
                    <a href="#"><i class="fa fa-linkedin" aria-
hidden="true"></i></a>
                </div>
            </div>
        </div>
    </div>
</footer>
<!-- ##### Footer Area End ##### -->

<!-- ##### All Javascript Files ##### -->
<!-- jQuery-2.2.4 js -->
<script src="{% static 'js/jquery/jquery-2.2.4.min.js' %}"></script>
<!-- Popper js -->
<script src="{% static 'js/bootstrap/popper.min.js' %}"></script>
<!-- Bootstrap js -->
<script src="{% static 'js/bootstrap/bootstrap.min.js' %}"></script>
```

```

    <!-- All Plugins js -->
    <script src="{% static 'js/plugins/plugins.js' %}"></script>
    <!-- Active js -->
    <script src="{% static 'js/active.js' %}"></script>

</body>

</html>

```

Admin page

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <title>Admin</title>
    <!-- plugins:css -->
    <link rel="stylesheet" href="{% static
'assets/vendors/mdi/css/materialdesignicons.min.css' %}">
    <link rel="stylesheet" href="{% static
'assets/vendors/css/vendor.bundle.base.css' %}">
    <!-- Plugin css for this page -->
    <link rel="stylesheet" href="{% static 'assets/vendors/jvectormap/jquery-
jvectormap.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/flag-icon-css/css/flag-
icon.min.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/owl-carousel-
2/owl.carousel.min.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/owl-carousel-
2/owl.theme.default.min.css' %}">
    <!-- Layout styles -->
    <link rel="stylesheet" href="{% static 'assets/css/style.css' %}">
    <!-- Custom styles for SkillVoyage -->
    <style>
        .sidebar-brand-wrapper {
            background-color: #f8f9fa;
            padding: 10px;
            border-bottom: 1px solid #ddd;
            font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;
            font-size: 24px;
            color: #333;
            font-weight: 700;
            text-transform: uppercase;
        }
    </style>

```

```

        .dashboard-card {
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0,0,0,0.1);
            padding: 20px;
            margin: 10px 0;
            text-align: center;
        }
        .dashboard-card h3 {
            margin-bottom: 10px;
        }
        .dashboard-card p {
            font-size: 20px;
            margin: 0;
        }
    }
</style>
<!-- End layout styles -->
<link rel="shortcut icon" href="{% static 'assets/images/favicon.png' %}" />
</head>
<body>
<div class="container-scroller">
    <!-- partial:partials/_sidebar.html -->
    <nav class="sidebar sidebar-offcanvas" id="sidebar">
        <div class="sidebar-brand-wrapper d-none d-lg-flex align-items-center justify-content-center fixed-top">
            <span class="brand-text">SKILLVOYAGE</span>
        </div>
        <ul class="nav">
            <li class="nav-item profile">
                <div class="profile-desc">
                    <div class="profile-pic">
                        <div class="count-indicator">
                            <!---->
                        </div>
                        <div class="profile-name">
                            <h5 class="mb-0 font-weight-normal">Admin</h5>
                            <span>Hello admin</span>
                        </div>
                    </div>
                    <a href="#" id="profile-dropdown" data-toggle="dropdown"><i class="mdi mdi-dots-vertical"></i></a>
                    <div class="dropdown-menu dropdown-menu-right sidebar-dropdown preview-list" aria-labelledby="profile-dropdown">
                        <a href="#" class="dropdown-item preview-item">
                            <div class="preview-thumbnail">
                                <div class="preview-icon bg-dark rounded-circle">
                                    <i class="mdi mdi-settings text-primary"></i>
                                </div>
                            </div>
                        </a>
                    </div>
                </li>
            </ul>
        </div>
    </body>
</html>

```

```

        <div class="preview-item-content">
            <p class="preview-subject ellipsis mb-1 text-
small">Account settings</p>
        </div>
    </a>
    <div class="dropdown-divider"></div>
    <a href="{% url 'logout' %}" class="dropdown-item preview-
item">
        <div class="preview-thumbnail">
            <div class="preview-icon bg-dark rounded-circle">
                <i class="mdi mdi-power text-info"></i>
            </div>
        </div>
        <div class="preview-item-content">
            <p class="preview-subject ellipsis mb-1 text-
small">Logout</p>
        </div>
    </a>
</div>
</div>
</li>
<li class="nav-item nav-category">
    <span class="nav-link">Navigation</span>
</li>
<li class="nav-item menu-items">
    <a class="nav-link" href="{% url 'admin_view' %}">
        <span class="menu-icon">
            <i class="mdi mdi-speedometer"></i>
        </span>
        <span class="menu-title">Dashboard</span>
    </a>
</li>
<li class="nav-item menu-items">
    <a class="nav-link" href="{% url 'add_category' %}">
        <span class="menu-icon">
            <i class="mdi mdi-view-list"></i>
        </span>
        <span class="menu-title">Categories</span>
    </a>
</li>
<li class="nav-item menu-items">
    <a class="nav-link" href="{% url 'create_subcategory' %}">
        <span class="menu-icon">
            <i class="mdi mdi-tag"></i>
        </span>
        <span class="menu-title">Subcategories</span>
    </a>
</li>
<li class="nav-item menu-items">

```



```

        <a class="nav-link" href="{% url 'teacher_requests' %}">
            <span class="menu-icon">
                <i class="mdi mdi-account-check"></i>
            </span>
            <span class="menu-title">Teacher Requests</span>
        </a>
    </li>
    <li class="nav-item menu-items">
        <a class="nav-link" href="{% url 'view_categories' %}">
            <span class="menu-icon">
                <i class="mdi mdi-eye"></i>
            </span>
            <span class="menu-title">View Categories</span>
        </a>
    </li>
    <li class="nav-item menu-items">
        <a class="nav-link" href="{% url 'view_subcategories' %}">
            <span class="menu-icon">
                <i class="mdi mdi-eye"></i>
            </span>
            <span class="menu-title">View Subcategories</span>
        </a>
    </li>
    <li class="nav-item menu-items">
        <a class="nav-link" href="{% url 'add_product_rent' %}">
            <span class="menu-icon">
                <i class="mdi mdi-plus-box"></i>
            </span>
            <span class="menu-title">Add Products for Rent</span>
        </a>
    </li>
    <li class="nav-item menu-items">
        <a class="nav-link" href="{% url 'add_product_buy' %}">
            <span class="menu-icon">
                <i class="mdi mdi-cart"></i>
            </span>
            <span class="menu-title">Add Products for Buying</span>
        </a>
    </li>
    <li class="nav-item menu-items">
        <a class="nav-link" href="{% url 'admin_contact' %}">
            <span class="menu-icon">
                <i class="mdi mdi-message"></i>
            </span>
            <span class="menu-title">Contact Us Requests</span>
        </a>
    </li>
    <li class="nav-item menu-items">
        <a class="nav-link" href="{% url 'admin_courses' %}">

```

```

        <span class="menu-icon">
            <i class="mdi mdi-play"></i>
        </span>
        <span class="menu-title">Courses</span>
    </a>
</li>
<li class="nav-item menu-items">
    <a class="nav-link" href="{% url 'admin_list_product' %}">
        <span class="menu-icon">
            <i class="mdi mdi-box"></i>
        </span>
        <span class="menu-title">Products</span>
    </a>
</li>
<li class="nav-item menu-items">
    <a class="nav-link" href="{% url 'admin_courses' %}">
        <span class="menu-icon">
            <i class="mdi mdi-cart"></i>
        </span>
        <span class="menu-title">Orders</span>
    </a>
</li>
</ul>
</nav>
<!-- partial -->
<div class="container-fluid page-body-wrapper">
    <!-- partial:partials/_navbar.html -->
    <nav class="navbar p-0 fixed-top d-flex flex-row">
        <div class="navbar-brand-wrapper d-flex d-lg-none align-items-center justify-content-center">
            <a class="navbar-brand brand-logo-mini" href="{% url 'admin_view' %}"></a>
        </div>
        <div class="navbar-menu-wrapper flex-grow d-flex align-items-stretch">
            <button class="navbar-toggler navbar-toggler align-self-center" type="button" data-toggle="minimize">
                <span class="mdi mdi-menu"></span>
            </button>
            <ul class="navbar-nav w-100">
                <li class="nav-item w-100">
                    <form class="nav-link mt-2 mt-md-0 d-none d-lg-flex search">
                        <input type="text" class="form-control" placeholder="Search " />
                    </form>
                </li>
            </ul>
            <ul class="navbar-nav navbar-nav-right">
                <li class="nav-item dropdown">

```

```

        <a class="nav-link count-indicator dropdown-toggle"
id="messageDropdown" href="#" data-toggle="dropdown" aria-expanded="false">
            <i class="mdi mdi-email"></i>
            <span class="count bg-success"></span>
        </a>
        <div class="dropdown-menu dropdown-menu-right navbar-
dropdown preview-list" aria-labelledby="messageDropdown">
            <h6 class="p-3 mb-0">Messages</h6>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                    
                </div>
                <div class="preview-item-content flex-grow">
                    <p class="preview-subject ellipsis mb-1">Mark
send you a message</p>
                    <p class="text-muted mb-0"> 1 Minutes ago </p>
                </div>
            </a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                    
                </div>
                <div class="preview-item-content flex-grow">
                    <p class="preview-subject ellipsis mb-1">Cregh
send you a message</p>
                    <p class="text-muted mb-0"> 15 Minutes ago
</p>
                </div>
            </a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                    
                </div>
                <div class="preview-item-content flex-grow">
                    <p class="preview-subject ellipsis mb-
1">Profile picture updated</p>
                    <p class="text-muted mb-0"> 18 Minutes ago
</p>
                </div>
            </a>
            <div class="dropdown-divider"></div>
            <h6 class="p-3 mb-0 text-center">4 new messages</h6>
        </div>

```

```

        </li>
        <li class="nav-item dropdown">
            <a class="nav-link count-indicator dropdown-toggle"
id="notificationDropdown" href="#" data-toggle="dropdown">
                <i class="mdi mdi-bell"></i>
                <span class="count bg-danger"></span>
            </a>
            <div class="dropdown-menu dropdown-menu-right navbar-
dropdown preview-list" aria-labelledby="notificationDropdown">
                <h6 class="p-3 mb-0">Notifications</h6>
                <div class="dropdown-divider"></div>
                <a class="dropdown-item preview-item">
                    <div class="preview-thumbnail">
                        <div class="preview-icon bg-dark rounded-
circle">
                            <i class="mdi mdi-calendar text-
success"></i>
                        </div>
                    </div>
                    <div class="preview-item-content">
                        <p class="preview-subject mb-1">Event
today</p>
                        <p class="text-muted ellipsis mb-0"> Just a
reminder that you have an event today </p>
                    </div>
                </a>
                <div class="dropdown-divider"></div>
                <a class="dropdown-item preview-item">
                    <div class="preview-thumbnail">
                        <div class="preview-icon bg-dark rounded-
circle">
                            <i class="mdi mdi-settings text-
danger"></i>
                        </div>
                    </div>
                    <div class="preview-item-content">
                        <p class="preview-subject mb-1">Settings</p>
                        <p class="text-muted ellipsis mb-0"> Update
dashboard </p>
                    </div>
                </a>
                <div class="dropdown-divider"></div>
                <a class="dropdown-item preview-item">
                    <div class="preview-thumbnail">
                        <div class="preview-icon bg-dark rounded-
circle">
                            <i class="mdi mdi-link-variant text-
warning"></i>
                        </div>
                    </div>

```

```

        </div>
        <div class="preview-item-content">
            <p class="preview-subject mb-1">Launch
Admin</p>
            <p class="text-muted ellipsis mb-0"> New admin
wow! </p>
        </div>
    </a>
    <div class="dropdown-divider"></div>
    <h6 class="p-3 mb-0 text-center">See all
notifications</h6>
    </div>
</li>
<li class="nav-item nav-settings d-none d-lg-block">
    <a class="nav-link" href="#">
        <i class="mdi mdi-format-line-spacing"></i>
    </a>
</li>
<li class="nav-item nav-logout d-none d-lg-block">
    <a class="nav-link" href="{% url 'logout' %}">
        <i class="mdi mdi-power"></i>
    </a>
</li>
<li class="nav-item nav-settings d-none d-lg-block">
    <a class="nav-link" href="#">
        <i class="mdi mdi-format-line-spacing"></i>
    </a>
</li>
</ul>
<button class="navbar-toggler navbar-toggler-right d-lg-none
align-self-center" type="button" data-toggle="offcanvas">
    <span class="mdi mdi-menu"></span>
</button>
</div>
</nav>
<!-- partial -->
<div class="main-panel">
    <div class="content-wrapper">
        <div class="row">
            <div class="col-md-12 grid-margin">
                <div class="row">
                    <div class="col-12 col-xl-8 mb-4 mb-xl-0">
                        <h3 class="font-weight-bold">Welcome Admin</h3>
                        <h6 class="font-weight-normal mb-0">All systems
are running smoothly! <span class="text-primary">3 unread alerts!</span></h6>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        <!-- Dashboard Cards Section -->
        <div class="row">
            <div class="col-md-4 grid-margin stretch-card">
                <div class="card dashboard-card">
                    <h3>Total Students Enrolled</h3>
                    <p>{{ student_count }}</p>
                </div>
            </div>
            <div class="col-md-4 grid-margin stretch-card">
                <div class="card dashboard-card">
                    <h3>Total Teachers Enrolled</h3>
                    <p>{{ teacher_count }}</p>
                </div>
            </div>
            <div class="col-md-4 grid-margin stretch-card">
                <div class="card dashboard-card">
                    <h3>Teachers To Approve</h3>
                    <p>{{ pending_teacher_count }}</p>
                </div>
            </div>
        </div>
        <!-- content-wrapper ends -->
        <!-- partial:partials/_footer.html -->
        <footer class="footer">
            <div class="d-sm-flex justify-content-center justify-content-sm-
between">
                <span class="text-muted text-center text-sm-left d-block d-sm-
inline-block">© 2023 SkillVoyage. All rights reserved.</span>
            </div>
        </footer>
        <!-- partial -->
    </div>
    <!-- main-panel ends -->
</div>
<!-- page-body-wrapper ends -->
</div>
<!-- container-scroller -->

<!-- plugins:js -->
<script src="{% static 'assets/vendors/js/vendor.bundle.base.js' %}"></script>
<!-- Plugin js for this page -->
<script src="{% static 'assets/vendors/chart.js/Chart.min.js' %}"></script>
<script src="{% static 'assets/vendors/progressbar.js/progressbar.min.js' %}"></script>
<script src="{% static 'assets/vendors/jvectormap/jquery-jvectormap.min.js' %}"></script>

```

```

<script src="{% static 'assets/vendors/jvectormap/jquery-jvectormap-world-mill-
en.js' %}"></script>
<script src="{% static 'assets/vendors/owl-carousel-2/owl.carousel.min.js'
%}"></script>
<!-- End plugin js for this page -->
<!-- inject:js -->
<script src="{% static 'assets/js/off-canvas.js' %}"></script>
<script src="{% static 'assets/js/hoverable-collapse.js' %}"></script>
<script src="{% static 'assets/js/misc.js' %}"></script>
<script src="{% static 'assets/js/settings.js' %}"></script>
<script src="{% static 'assets/js/todolist.js' %}"></script>
<!-- endinject -->
<!-- Custom js for this page -->
<script src="{% static 'assets/js/dashboard.js' %}"></script>
<!-- End custom js for this page -->
</body>
</html>

```

Teacher Dashboard page

```

{% load static %}
{% load widget_tweaks %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="description" content="">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <title>SkillVoyage - Teacher Qualification</title>
    <link rel="stylesheet" href="{% static 'style.css' %}?v-1">
</head>
<body>
    <header class="header-area bg-dark">
        <div class="oneMusic-main-menu">
            <div class="classy-nav-container breakpoint-off">
                <div class="container">
                    <nav class="classy-navbar justify-content-between"
id="oneMusicNav">
                        <a href="{% url 'home' %}" class="nav-
brand">SkillVoyage</a>
                        <div class="classy-navbar-toggler">
                            <span
class="navbarToggler"><span></span><span></span><span></span></span>
                            </div>
                        <div class="classy-menu">

```

```
<div class="classycloseIcon">
    <div class="cross-wrap"><span
class="top"></span><span class="bottom"></span></div>
    </div>
    <div class="classynav">
        <ul>

            <li><a href="#">Categories</a>
                <ul class="dropdown">
                    <li><a href="{% url 'courses'
%}">Music</a></li>

                    <li><a href="{% url 'courses'
%}">Dance</a></li>

                    <li><a href="{% url 'courses'
%}">Drawing</a></li>

                    <li><a href="{% url 'courses'
%}">Karate</a></li>

                </ul>
            </li>
            <li><a href="{% url 'events'
%}">Events</a></li>

            <li><a href="{% url 'contact'
%}">Contact</a></li>

        </ul>
        <div class="logout-btn">
            <a href="{% url 'logout' %}" class="btn
oneMusic-btn">Logout</a>
        </div>
    </div>
</div>
</nav>
</div>
</div>
</div>
</header>

<section class="login-area section-padding-100">
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-12 col-lg-10">
                <div class="login-content">
                    <div class="d-flex justify-content-between align-items-
center mb-4">

                        <h3 class="text-center">Welcome Teacher</h3>
                        <p>This is your dashboard</p>
                    </div>
                    <div class="container-fluid">
                        <div class="row d-flex" style="justify-content:
stretch;">
```



```

        {% for course in courses %}
        <div class="col-md-6 grid-margin stretch-
card">

            <div class="card">
                <div class="card-body">

                    {% if course.is_draft %}
                    <span class="badge badge-
primary">Draft</span>

                    {% elif course.is_pending %}
                    <span class="badge badge-
warning">Pending</span>

                    {% elif course.is_archived %}
                    <span class="badge badge-
danger">Archived</span>

                    {% else %}
                    <span class="badge badge-
success">Published</span>

                    {% endif %}

                    
                    <h4 class="card-title">{{
course.name }}</h4>

                    <p class="card-text">{{
course.sub_category.category }} > {{ course.sub_category }}</p>
                    <p class="card-text">
                        Lessons: {{
course.lessons.count }}

                        {% if course.lessons.count > 0
and not course.is_pending and not course.is_archived and not course.is_published
%}

                            |
                            <a href="{% url
'teacher_request_course_approval' id=course.id %}">Submit For Approval</a>
                        {% endif %}

                    </p>
                    <!-- Feedback Section -->

                    <div class="feedback-section">
                        <h5>Student Feedback:</h5>
                        <ul class="list-unstyled">
                            {% for feedback in course.feedback.all %}
                            <li>{{ feedback.student.username }}: {{ feedback.comment }} (Rating:
{{ feedback.rating }})</li>
                            {% empty %}

```

```
<li>No feedback available for this course.</li>
{% endfor %}
</ul>
</div>

        </div>
        <div class="card-footer">
            <a href="{% url
'teacher_edit_course' id=course.id %}" class="btn btn-primary">Edit</a>
            <a href="{% url
'teacher_create_course_lesson' id=course.id %}" class="btn btn-
secondary">Lessons</a>

                <!--{% if course.status !=
'archived' %}

                    <a href="{% url
'teacher_delete_course' id=course.id %}" class="btn btn-danger">Delete</a>
                    {% endif %}>-->

                {% if course.status == 'archived'
%}

                    <a href="{% url
'teacher_restore_course' id=course.id %}" class="btn btn-info">Restore</a>
                    {% endif %}

                {% if course.status == 'published'
%}

                    <a href="{% url
'teacher_course_students' id=course.id %}" class="btn btn-info">Students</a>
                    {% endif %}

            </div>
        </div>

    </div>
    {% endfor %}

</div>
</div>
</div>
</div>
</div>
</div>
</section>
<footer class="footer-area">
    <div class="container">
        <div class="row d-flex flex-wrap align-items-center">
```

```

        <div class="col-12 col-md-6">
            <p class="copywrite-text"><!-- Link back to Colorlib can't be
removed. Template is Licensed under CC BY 3.0. -->
<!-- Link back to Colorlib can't be removed. Template is Licensed under CC BY 3.0.
-->
</p>

        </div>
        <div class="col-12 col-md-6">
            <div class="footer-nav">
                <ul>
                    <li><a href="{% url 'home' %}">Home</a></li>
                    <li><a href="{% url 'courses' %}">Courses</a></li>
                    <li><a href="{% url 'events' %}">Events</a></li>
                    <li><a href="{% url 'contact' %}">Contact</a></li>
                </ul>
            </div>
        </div>
    </div>
</div>
</footer>
<script src="{% static 'js/jquery/jquery-2.2.4.min.js' %}"></script>
<script src="{% static 'js/bootstrap/popper.min.js' %}"></script>
<script src="{% static 'js/bootstrap/bootstrap.min.js' %}"></script>
<script src="{% static 'js/plugins/plugins.js' %}"></script>
<script src="{% static 'js/active.js' %}"></script>
</body>
</html>

```

Student home page

```

{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <title>Student</title>
    <!-- plugins:css -->
    <link rel="stylesheet" href="{% static
'assets/vendors/mdi/css/materialdesignicons.min.css' %}">
    <link rel="stylesheet" href="{% static
'assets/vendors/css/vendor.bundle.base.css' %}">
    <!-- Plugin css for this page -->
    <link rel="stylesheet" href="{% static 'assets/vendors/jvectormap/jquery-
jvectormap.css' %}">

```

```

    <link rel="stylesheet" href="{% static 'assets/vendors/flag-icon-css/css/flag-
icon.min.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/owl-carousel-
2/owl.carousel.min.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/owl-carousel-
2/owl.theme.default.min.css' %}">
    <!-- Layout styles -->
    <link rel="stylesheet" href="{% static 'assets/css/style.css' %}">
    <!-- Custom styles for SkillVoyage -->
    <style>
        .sidebar-brand-wrapper {
            background-color: #f8f9fa;
            padding: 10px;
            border-bottom: 1px solid #ddd;
            font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;
            font-size: 24px;
            color: #333;
            font-weight: 700;
            text-transform: uppercase;
        }

        .dashboard-card {
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            padding: 20px;
            margin: 10px 0;
            text-align: center;
        }

        .dashboard-card h3 {
            margin-bottom: 10px;
        }

        .dashboard-card p {
            font-size: 20px;
            margin: 0;
        }
    </style>
    <!-- End layout styles -->
    <link rel="shortcut icon" href="{% static 'assets/images/favicon.png' %}" />
</head>

<body>
    <div class="container-scroller">
        <!-- partial:partials/_sidebar.html -->
        <nav class="sidebar sidebar-offcanvas" id="sidebar">
            <div class="sidebar-brand-wrapper d-none d-lg-flex align-items-center
justify-content-center fixed-top">
                <span class="brand-text">SKILLVOYAGE</span>
            </div>
        </nav>
    </div>
</body>

```

```

        </div>
        <ul class="nav">
            <li class="nav-item profile">
                <div class="profile-desc">
                    <div class="profile-pic">
                        <div class="count-indicator">
                            <!---->
                        </div>
                        <div class="profile-name">
                            <h5 class="mb-0 font-weight-normal">Student</h5>
                            <span>Hello {{ user.first_name }}</span>
                        </div>
                    </div>
                    <a href="#" id="profile-dropdown" data-
toggle="dropdown"><i
                        class="mdi mdi-dots-vertical"></i></a>
                    <div class="dropdown-menu dropdown-menu-right sidebar-
dropdown preview-list"
                        aria-labelledby="profile-dropdown">
                        <a href="#" class="dropdown-item preview-item">
                            <div class="preview-thumbnail">
                                <div class="preview-icon bg-dark rounded-
circle">
                                    <i class="mdi mdi-settings text-
primary"></i>
                                </div>
                            </div>
                            <div class="preview-item-content">
                                <p class="preview-subject ellipsis mb-1 text-
small">Account settings</p>
                            </div>
                        </a>
                        <div class="dropdown-divider"></div>
                        <a href="{% url 'logout' %}" class="dropdown-item
preview-item">
                            <div class="preview-thumbnail">
                                <div class="preview-icon bg-dark rounded-
circle">
                                    <i class="mdi mdi-power text-info"></i>
                                </div>
                            </div>
                            <div class="preview-item-content">
                                <p class="preview-subject ellipsis mb-1 text-
small">Logout</p>
                            </div>
                        </a>
                    </div>
                </div>
            </li>
        </ul>
    </div>

```

```

        </li>
        <li class="nav-item nav-category">
            <span class="nav-link">Navigation</span>
        </li>
        {% include 'shared/student_sidenav.html' %}
    </ul>
</nav>
<!-- partial -->
<div class="container-fluid page-body-wrapper">
    <!-- partial:partials/_navbar.html -->
    <nav class="navbar p-0 fixed-top d-flex flex-row">
        <div class="navbar-brand-wrapper d-flex d-lg-none align-items-
center justify-content-center">
            <a class="navbar-brand brand-logo-mini" href="{% url
'admin_view' %}"></a>
        </div>
        <div class="navbar-menu-wrapper flex-grow d-flex align-items-
stretch">
            <button class="navbar-toggler navbar-toggler align-self-
center" type="button"
                data-toggle="minimize">
                <span class="mdi mdi-menu"></span>
            </button>
            <ul class="navbar-nav w-100">
                <li class="nav-item w-100">
                    <form class="nav-link mt-2 mt-md-0 d-none d-lg-flex
search">
                        <input type="text" class="form-control"
placeholder="Search ">
                    </form>
                </li>
            </ul>
            <ul class="navbar-nav navbar-nav-right">
                <li class="nav-item dropdown">
                    <a class="nav-link count-indicator dropdown-toggle"
id="messageDropdown" href="#"
                        data-toggle="dropdown" aria-expanded="false">
                        <i class="mdi mdi-email"></i>
                        <span class="count bg-success"></span>
                    </a>
                    <div class="dropdown-menu dropdown-menu-right navbar-
dropdown preview-list"
                        aria-labelledby="messageDropdown">
                        <h6 class="p-3 mb-0">Messages</h6>
                        <div class="dropdown-divider"></div>
                        <a class="dropdown-item preview-item">
                            <div class="preview-thumbnail">

```

```

                                
                                </div>
                                <div class="preview-item-content flex-grow">
                                <p class="preview-subject ellipsis mb-
1">Mark send you a message</p>
                                <p class="text-muted mb-0"> 1 Minutes ago
</p>
                                </div>
                                </a>

                                <li class="nav-item nav-logout d-none d-lg-block">
                                <a class="nav-link" href="{% url 'logout' %}">
                                <i class="mdi mdi-power"></i>
                                </a>
                                </li>
                                <li class="nav-item nav-settings d-none d-lg-block">
                                <a class="nav-link" href="#">
                                <i class="mdi mdi-format-line-spacing"></i>
                                </a>
                                </li>
                                </ul>
                                <button class="navbar-toggler navbar-toggler-right d-lg-none
align-self-center" type="button"
                                data-toggle="offcanvas">
                                <span class="mdi mdi-menu"></span>
                                </button>
                                </div>
                                </nav>
                                <!-- partial -->
                                <div class="main-panel">
                                <div class="content-wrapper">
                                <div class="row">
                                <div class="col-md-12 grid-margin">
                                <div class="row">
                                <div class="col-12 col-xl-8 mb-4 mb-xl-0">
                                <h3 class="font-weight-bold">Welcome {{
user.email }}</h3>
                                <p>Your Courses</p>
                                </div>
                                </div>
                                </div>

```

```

        </div>
    </div>

    <div class="row">

        {% if not courses %}
        <div class="alert alert-primary" role="alert">
            <strong>No courses enrolled.</strong>
            Click <a href="{% url 'student_more_courses' %}"
class="alert-link">here</a>
            to enroll in a course.
        </div>

        {% endif %}

        {% for course in courses %}
        <div class="col-sm-4">
            <div class="card">
                <div class="card-body">
                    
                    <h5 class="card-title">{{ course.course.name
}}</h5>

                    <p class="card-text">Category: {{
course.course.sub_category }}</p>
                    <p class="card-text">Price: {{
course.course.price }}</p>
                    <p class="card-text">{{
course.course.description }}</p>
                    <a href="{% url 'student_view_lesson'
id=course.course.lessons.first.id %}" class="btn btn-primary">Continue
Learning</a>
                </div>
            </div>
        </div>
        {% endfor %}

    </div>
</div>
<!-- content-wrapper ends -->
<!-- partial:partials/_footer.html -->
<footer class="footer">
    <div class="d-sm-flex justify-content-center justify-content-
sm-between">
        <span class="text-muted text-center text-sm-left d-block
d-sm-inline-block">© 2023 SkillVoyage.
            All rights reserved.</span>
    </div>

```



```

        </footer>
        <!-- partial -->
    </div>
    <!-- main-panel ends -->
</div>
<!-- page-body-wrapper ends -->
</div>
<!-- container-scroller -->

<!-- plugins:js -->
<script src="{% static 'assets/vendors/js/vendor.bundle.base.js' %}"></script>
<!-- Plugin js for this page -->
<script src="{% static 'assets/vendors/chart.js/Chart.min.js' %}"></script>
<script src="{% static 'assets/vendors/progressbar.js/progressbar.min.js' %}"></script>
<script src="{% static 'assets/vendors/jvectormap/jquery-jvectormap.min.js' %}"></script>
<script src="{% static 'assets/vendors/jvectormap/jquery-jvectormap-world-mill-en.js' %}"></script>
<script src="{% static 'assets/vendors/owl-carousel-2/owl.carousel.min.js' %}"></script>
<!-- End plugin js for this page -->
<!-- inject:js -->
<script src="{% static 'assets/js/off-canvas.js' %}"></script>
<script src="{% static 'assets/js/hoverable-collapse.js' %}"></script>
<script src="{% static 'assets/js/misc.js' %}"></script>
<script src="{% static 'assets/js/settings.js' %}"></script>
<script src="{% static 'assets/js/todolist.js' %}"></script>
<!-- endinject -->
<!-- Custom js for this page -->
<script src="{% static 'assets/js/dashboard.js' %}"></script>
<!-- End custom js for this page -->
</body>
</html>

```

Student enrollment page

```

{% load static %}
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Student</title>
    <!-- plugins:css -->

```

```

    <link rel="stylesheet" href="{% static
'assets/vendors/mdi/css/materialdesignicons.min.css' %}">
    <link rel="stylesheet" href="{% static
'assets/vendors/css/vendor.bundle.base.css' %}">
    <!-- Plugin css for this page -->
    <link rel="stylesheet" href="{% static 'assets/vendors/jvectormap/jquery-
jvectormap.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/flag-icon-css/css/flag-
icon.min.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/owl-carousel-
2/owl.carousel.min.css' %}">
    <link rel="stylesheet" href="{% static 'assets/vendors/owl-carousel-
2/owl.theme.default.min.css' %}">
    <!-- Layout styles -->
    <link rel="stylesheet" href="{% static 'assets/css/style.css' %}">
    <!-- Custom styles for SkillVoyage -->
    <style>
        .sidebar-brand-wrapper {
            background-color: #f8f9fa;
            padding: 10px;
            border-bottom: 1px solid #ddd;
            font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;
            font-size: 24px;
            color: #333;
            font-weight: 700;
            text-transform: uppercase;
        }

        .dashboard-card {
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            padding: 20px;
            margin: 10px 0;
            text-align: center;
        }

        .dashboard-card h3 {
            margin-bottom: 10px;
        }

        .dashboard-card p {
            font-size: 20px;
            margin: 0;
        }
    </style>
    <!-- End layout styles -->
    <link rel="shortcut icon" href="{% static 'assets/images/favicon.png' %}" />
</head>

```

```

<body>
  <div class="container-scroller">
    <!-- partial:partials/_sidebar.html -->
    <nav class="sidebar sidebar-offcanvas" id="sidebar">
      <div class="sidebar-brand-wrapper d-none d-lg-flex align-items-center justify-content-center fixed-top">
        <span class="brand-text">SKILLVOYAGE</span>
      </div>
      <ul class="nav">
        <li class="nav-item profile">
          <div class="profile-desc">
            <div class="profile-pic">
              <div class="count-indicator">
                <!---->
              </div>
              <div class="profile-name">
                <h5 class="mb-0 font-weight-normal">Student</h5>
                <span>Hello {{ user.first_name }}</span>
              </div>
            </div>
            <a href="#" id="profile-dropdown" data-toggle="dropdown"><i class="mdi mdi-dots-vertical"></i></a>
            <div class="dropdown-menu dropdown-menu-right sidebar-dropdown preview-list" aria-labelledby="profile-dropdown">
              <a href="#" class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                  <div class="preview-icon bg-dark rounded-circle">
                    <i class="mdi mdi-settings text-primary"></i>
                  </div>
                </div>
                <div class="preview-item-content">
                  <p class="preview-subject ellipsis mb-1 text-small">Account settings</p>
                </div>
              </a>
              <div class="dropdown-divider"></div>
              <a href="{% url 'logout' %}" class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                  <div class="preview-icon bg-dark rounded-circle">
                    <i class="mdi mdi-power text-info"></i>
                  </div>
                </div>
              </a>
            </div>
          </li>
        </ul>
      </div>
    </div>
  </div>

```

```

        <div class="preview-item-content">
            <p class="preview-subject ellipsis mb-1 text-
small">Logout</p>
        </div>
    </a>
</div>
</div>
</li>
<li class="nav-item nav-category">
    <span class="nav-link">Navigation</span>
</li>
{% include 'shared/student_sidenav.html' %}
</ul>
</nav>
<!-- partial -->
<div class="container-fluid page-body-wrapper">
    <!-- partial:partials/_navbar.html -->
    <nav class="navbar p-0 fixed-top d-flex flex-row">
        <div class="navbar-brand-wrapper d-flex d-lg-none align-items-
center justify-content-center">
            <a class="navbar-brand brand-logo-mini" href="{% url
'admin_view' %}"></a>
        </div>
        <div class="navbar-menu-wrapper flex-grow d-flex align-items-
stretch">
            <button class="navbar-toggler navbar-toggler align-self-
center" type="button"
                data-toggle="minimize">
                <span class="mdi mdi-menu"></span>
            </button>
            <ul class="navbar-nav w-100">
                <li class="nav-item w-100">
                    <form class="nav-link mt-2 mt-md-0 d-none d-lg-flex
search">
                        <input type="text" class="form-control"
placeholder="Search ">
                    </form>
                </li>
            </ul>
            <ul class="navbar-nav navbar-nav-right">
                <li class="nav-item dropdown">
                    <a class="nav-link count-indicator dropdown-toggle"
id="messageDropdown" href="#"
                        data-toggle="dropdown" aria-expanded="false">
                        <i class="mdi mdi-email"></i>
                        <span class="count bg-success"></span>
                    </a>

```

```

<div class="dropdown-menu dropdown-menu-right navbar-
dropdown preview-list"
    aria-labelledby="messageDropdown">
    <h6 class="p-3 mb-0">Messages</h6>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item preview-item">
        <div class="preview-thumbnail">
            
            </div>
            <div class="preview-item-content flex-grow">
                <p class="preview-subject ellipsis mb-
1">Mark send you a message</p>
                <p class="text-muted mb-0"> 1 Minutes ago
</p>
            </div>
        </a>
        <div class="dropdown-divider"></div>
        <a class="dropdown-item preview-item">
            <div class="preview-thumbnail">
                
                </div>
                <div class="preview-item-content flex-grow">
                    <p class="preview-subject ellipsis mb-
1">Cregh send you a message</p>
                    <p class="text-muted mb-0"> 15 Minutes ago
</p>
                </div>
            </a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                    
                    </div>
                    <div class="preview-item-content flex-grow">
                        <p class="preview-subject ellipsis mb-
1">Profile picture updated</p>
                        <p class="text-muted mb-0"> 18 Minutes ago
</p>
                    </div>
                </a>
                <div class="dropdown-divider"></div>
                <h6 class="p-3 mb-0 text-center">4 new
messages</h6>

```

```

        </div>
    </li>
    <li class="nav-item dropdown">
        <a class="nav-link count-indicator dropdown-toggle"
id="notificationDropdown" href="#"
        data-toggle="dropdown">
            <i class="mdi mdi-bell"></i>
            <span class="count bg-danger"></span>
        </a>
        <div class="dropdown-menu dropdown-menu-right navbar-
dropdown preview-list"
            aria-labelledby="notificationDropdown">
            <h6 class="p-3 mb-0">Notifications</h6>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                    <div class="preview-icon bg-dark rounded-
circle">
                        <i class="mdi mdi-calendar text-
success"></i>
                    </div>
                </div>
                <div class="preview-item-content">
                    <p class="preview-subject mb-1">Event
today</p>
                    <p class="text-muted ellipsis mb-0"> Just
a reminder that you have an event
                        today </p>
                </div>
            </a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item preview-item">
                <div class="preview-thumbnail">
                    <div class="preview-icon bg-dark rounded-
circle">
                        <i class="mdi mdi-settings text-
danger"></i>
                    </div>
                </div>
                <div class="preview-item-content">
                    <p class="preview-subject mb-
1">Settings</p>
                    <p class="text-muted ellipsis mb-0">
Update dashboard </p>
                </div>
            </a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item preview-item">
                <div class="preview-thumbnail">

```

```

        <div class="preview-icon bg-dark rounded-
circle">
            <i class="mdi mdi-link-variant text-
warning"></i>
        </div>
    </div>
    <div class="preview-item-content">
        <p class="preview-subject mb-1">Launch
Admin</p>
        <p class="text-muted ellipsis mb-0"> New
admin wow! </p>
    </div>
</a>
<div class="dropdown-divider"></div>
<h6 class="p-3 mb-0 text-center">See all
notifications</h6>
    </div>
</li>
<li class="nav-item nav-settings d-none d-lg-block">
    <a class="nav-link" href="#">
        <i class="mdi mdi-format-line-spacing"></i>
    </a>
</li>
<li class="nav-item nav-logout d-none d-lg-block">
    <a class="nav-link" href="{% url 'logout' %}">
        <i class="mdi mdi-power"></i>
    </a>
</li>
<li class="nav-item nav-settings d-none d-lg-block">
    <a class="nav-link" href="#">
        <i class="mdi mdi-format-line-spacing"></i>
    </a>
</li>
</ul>
<button class="navbar-toggler navbar-toggler-right d-lg-none
align-self-center" type="button"
    data-toggle="offcanvas">
    <span class="mdi mdi-menu"></span>
</button>
</div>
</nav>
<!-- partial -->
<div class="main-panel">
    <div class="content-wrapper">
        <div class="row">
            <div class="col-md-12 grid-margin">
                <div class="row">
                    <div class="col-12 col-xl-8 mb-4 mb-xl-0">

```

```

        <h3 class="font-weight-bold">Welcome {{
user.email }}</h3>

        <p>Your Courses</p>
    </div>
</div>
</div>
</div>

<div class="container">
    <script src="https://js.stripe.com/v3/"></script>
    <h2>Complete Your Payment</h2>
    <br/>
    <button id="checkout-button" class="btn btn-primary">Pay
Using Stripe</button>

    <script type="text/javascript">
        var stripe = Stripe('{{ stripe_public_key }}'); //
Use public key from Django

        var checkoutButton =
document.getElementById('checkout-button');

        checkoutButton.addEventListener('click', function () {
            fetch('/student/create-course-payment/{{ course.id
}}/', {

                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                    'X-CSRFToken': '{{ csrf_token }}' //
Include CSRF token

                }
            })
                .then(function (response) {
                    return response.json();
                })
                .then(function (sessionId) {
                    return stripe.redirectToCheckout({
sessionId: sessionId.id });

                })
                .then(function (result) {
                    if (result.error) {
                        alert(result.error.message);
                    }
                });
        });
    </script>
</div>
</div>
<!-- content-wrapper ends -->

```



```
        <!-- partial:partials/_footer.html -->
        <footer class="footer">
            <div class="d-sm-flex justify-content-center justify-content-
sm-between">
                <span class="text-muted text-center text-sm-left d-block
d-sm-inline-block">© 2023 SkillVoyage.
                    All rights reserved.</span>
            </div>
        </footer>
        <!-- partial -->
    </div>
    <!-- main-panel ends -->
</div>
<!-- page-body-wrapper ends -->
</div>
<!-- container-scroller -->

<!-- plugins:js -->
<script src="{% static 'assets/vendors/js/vendor.bundle.base.js' %}"></script>
<!-- Plugin js for this page -->
<script src="{% static 'assets/vendors/chart.js/Chart.min.js' %}"></script>
<script src="{% static 'assets/vendors/progressbar.js/progressbar.min.js'
%}"></script>
<script src="{% static 'assets/vendors/jvectormap/jquery-jvectormap.min.js'
%}"></script>
<script src="{% static 'assets/vendors/jvectormap/jquery-jvectormap-world-
mill-en.js' %}"></script>
<script src="{% static 'assets/vendors/owl-carousel-2/owl.carousel.min.js'
%}"></script>
<!-- End plugin js for this page -->
<!-- inject:js -->
<script src="{% static 'assets/js/off-canvas.js' %}"></script>
<script src="{% static 'assets/js/hoverable-collapse.js' %}"></script>
<script src="{% static 'assets/js/misc.js' %}"></script>
<script src="{% static 'assets/js/settings.js' %}"></script>
<script src="{% static 'assets/js/todolist.js' %}"></script>
<!-- endinject -->
<!-- Custom js for this page -->
<script src="{% static 'assets/js/dashboard.js' %}"></script>
<!-- End custom js for this page -->
</body>
</html>
```

9.1 Screen Shots

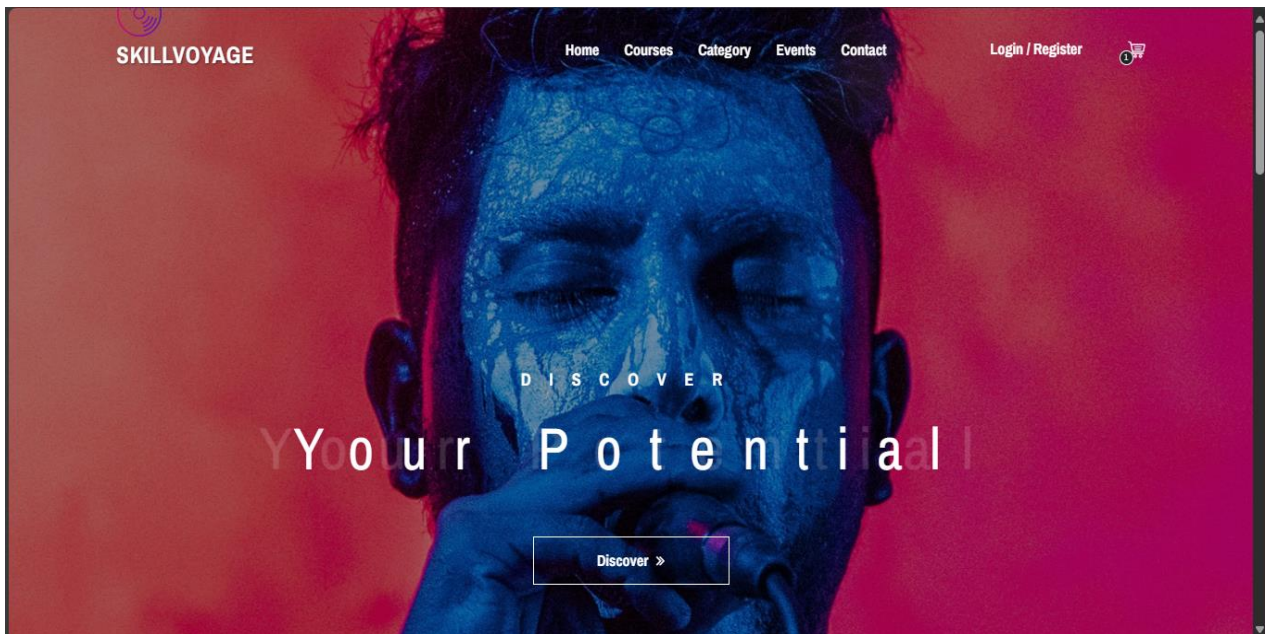


Fig 16. Home

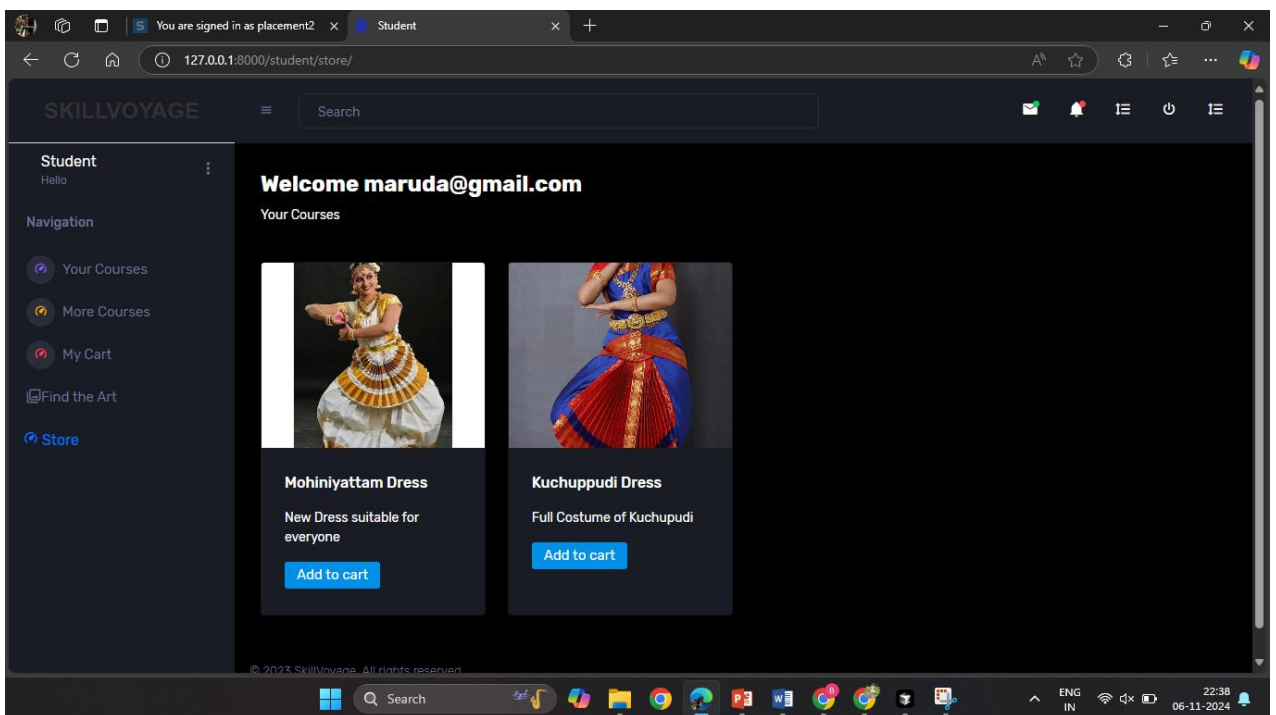


Fig 17. User store page

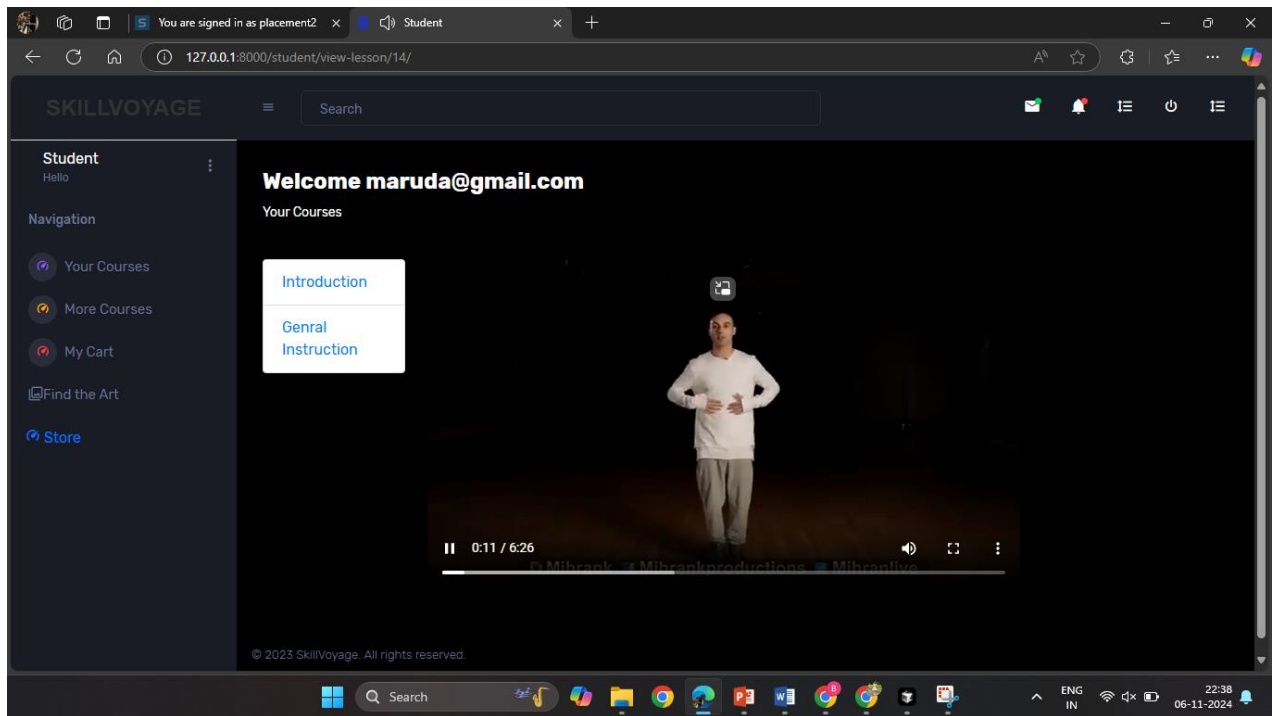


Fig 18.Course watching page

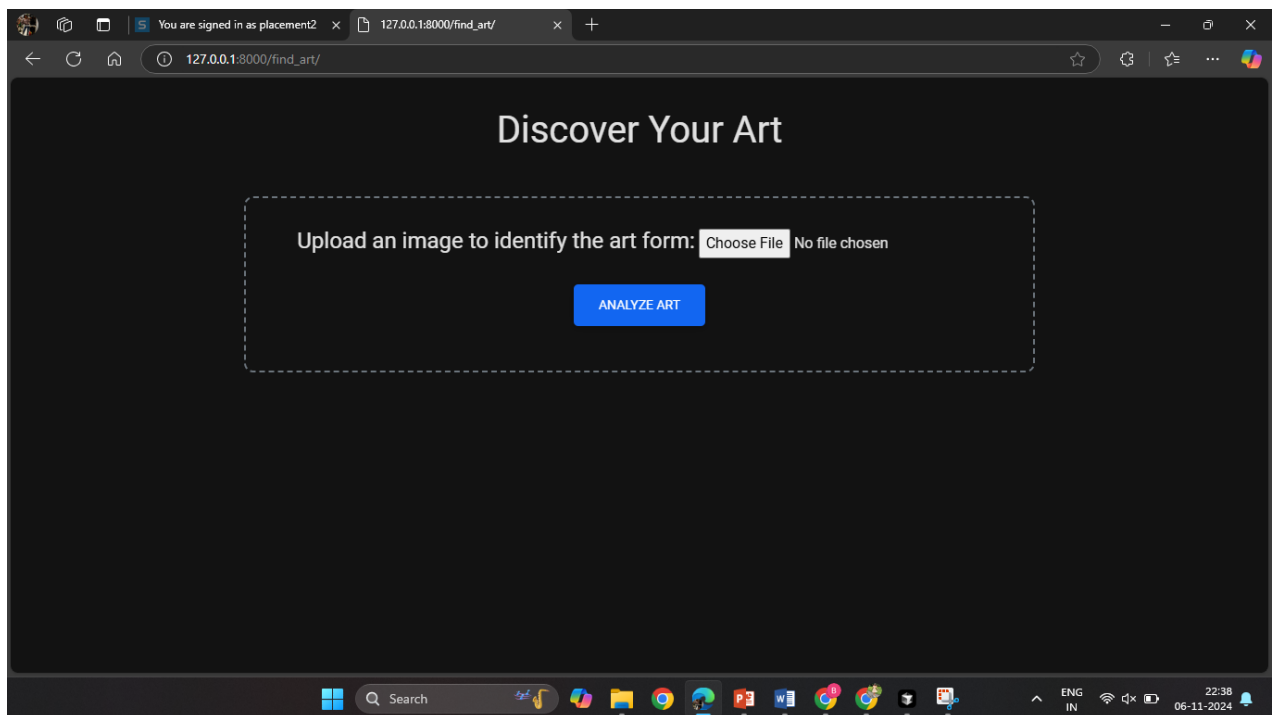


Fig 19. Art recognition

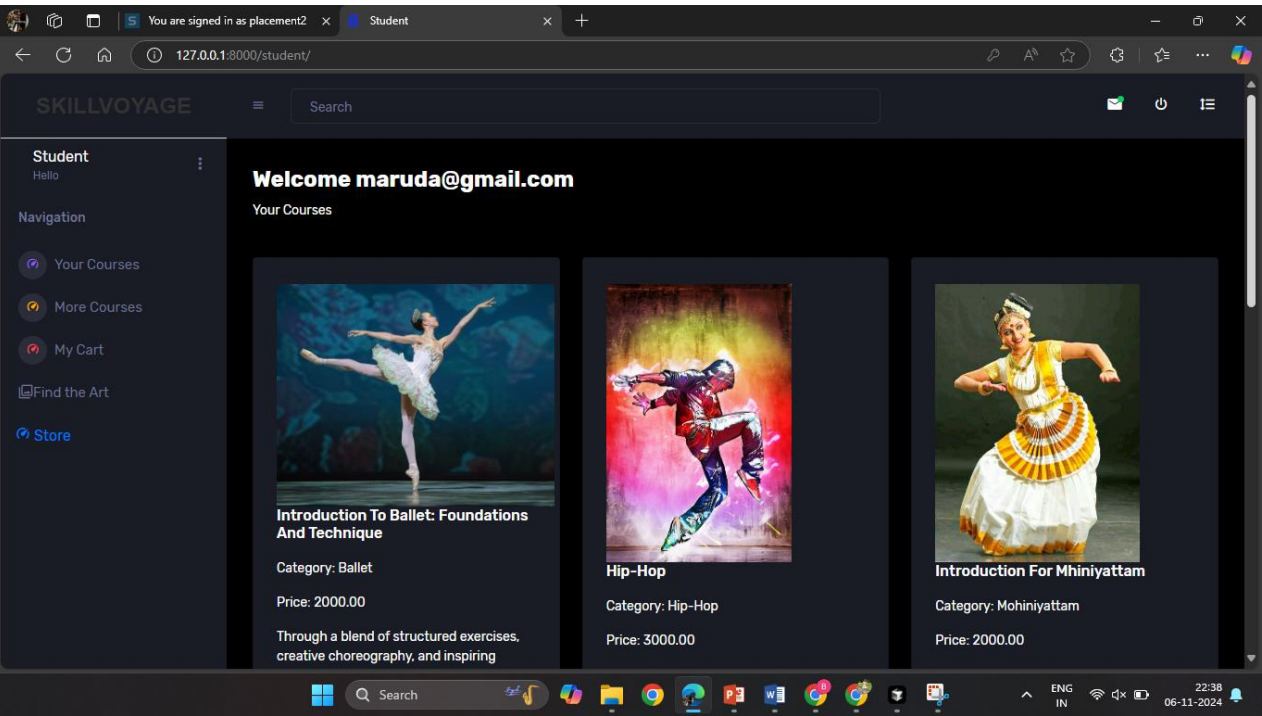


Fig 20. Student home page

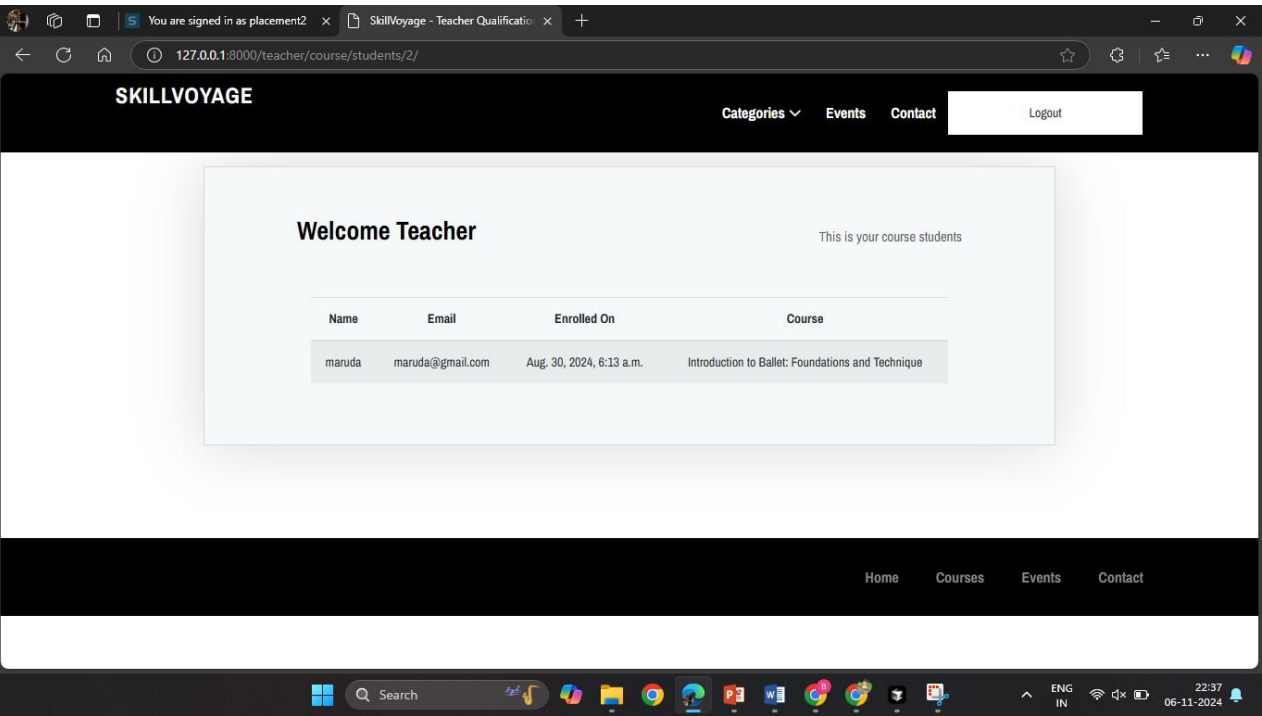


Fig 21. Teacher course enrolled students

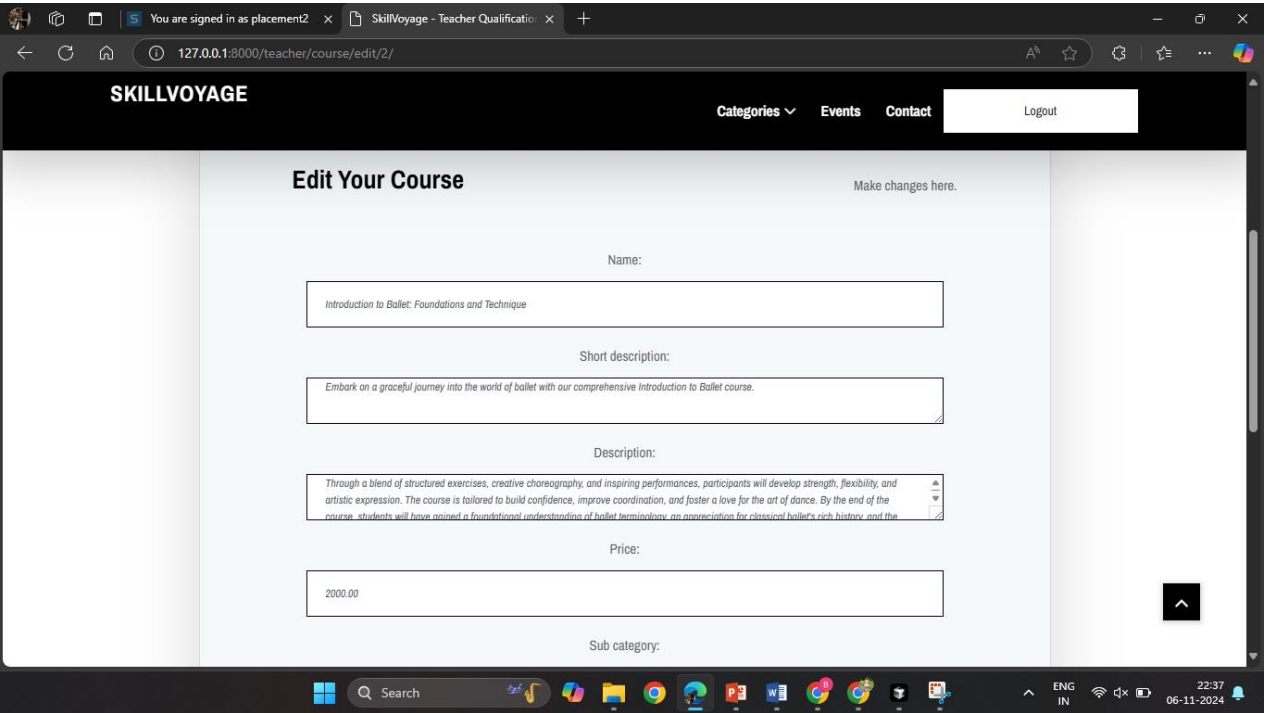


Fig 22. Course edit

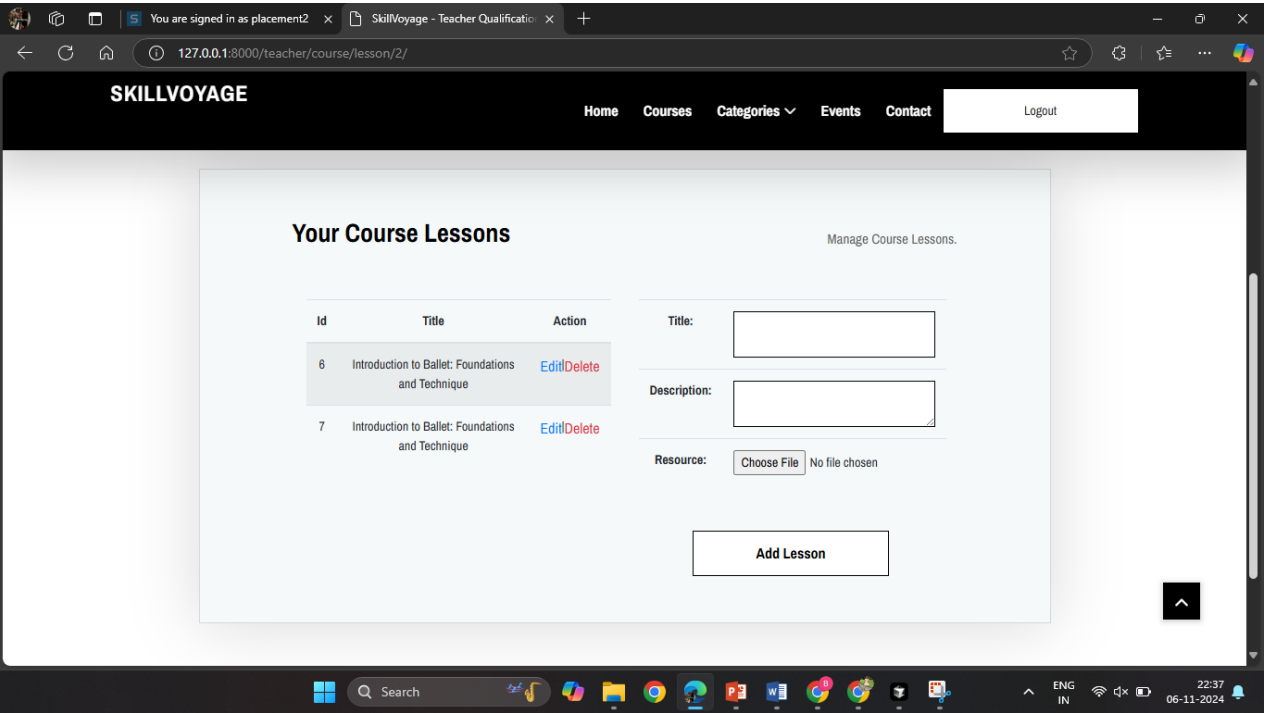


Fig 23. Lesson management page

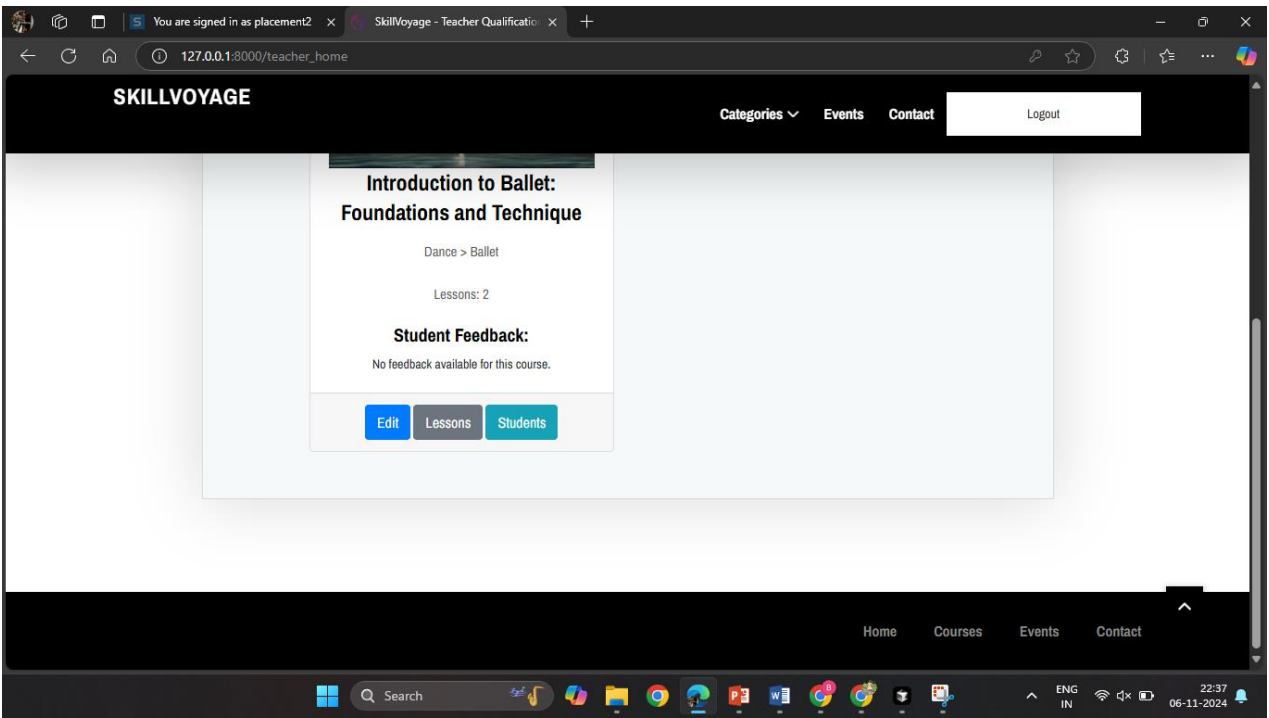


Fig 24. Teacher dashboard

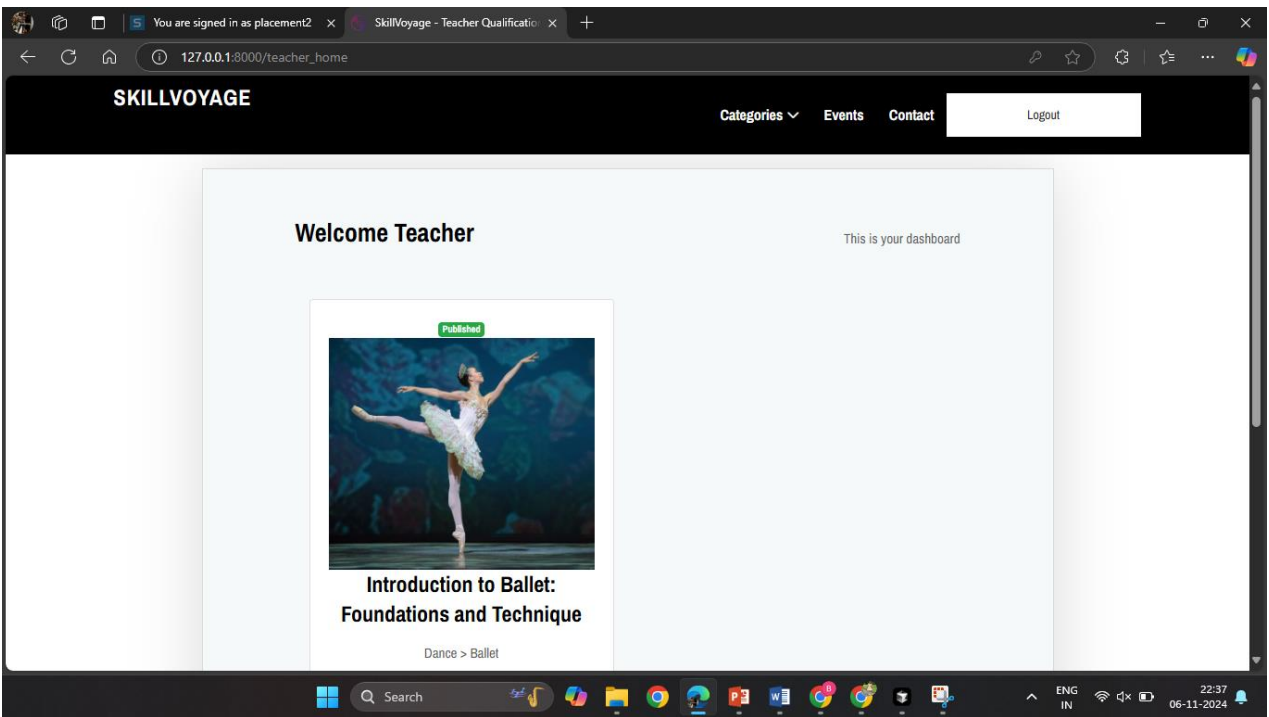


Fig 25. Teacher Dashboard

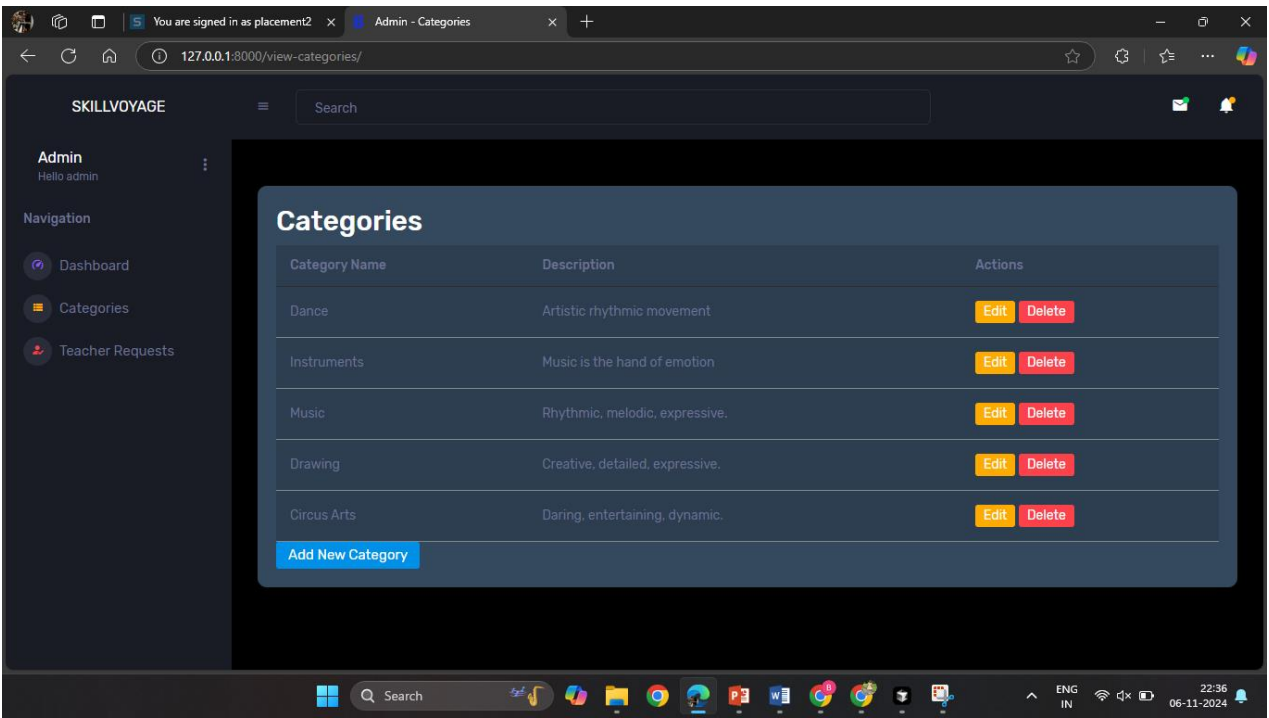


Fig 26. Category view

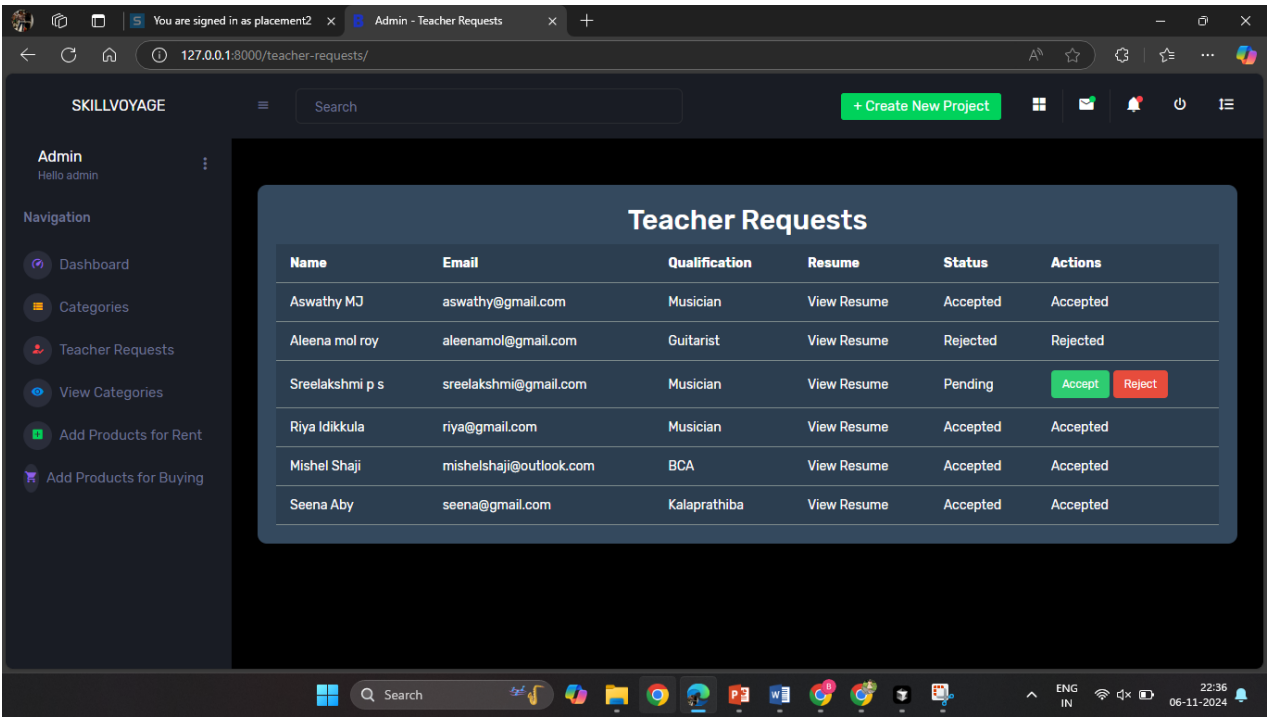


Fig 27. Teacher request page

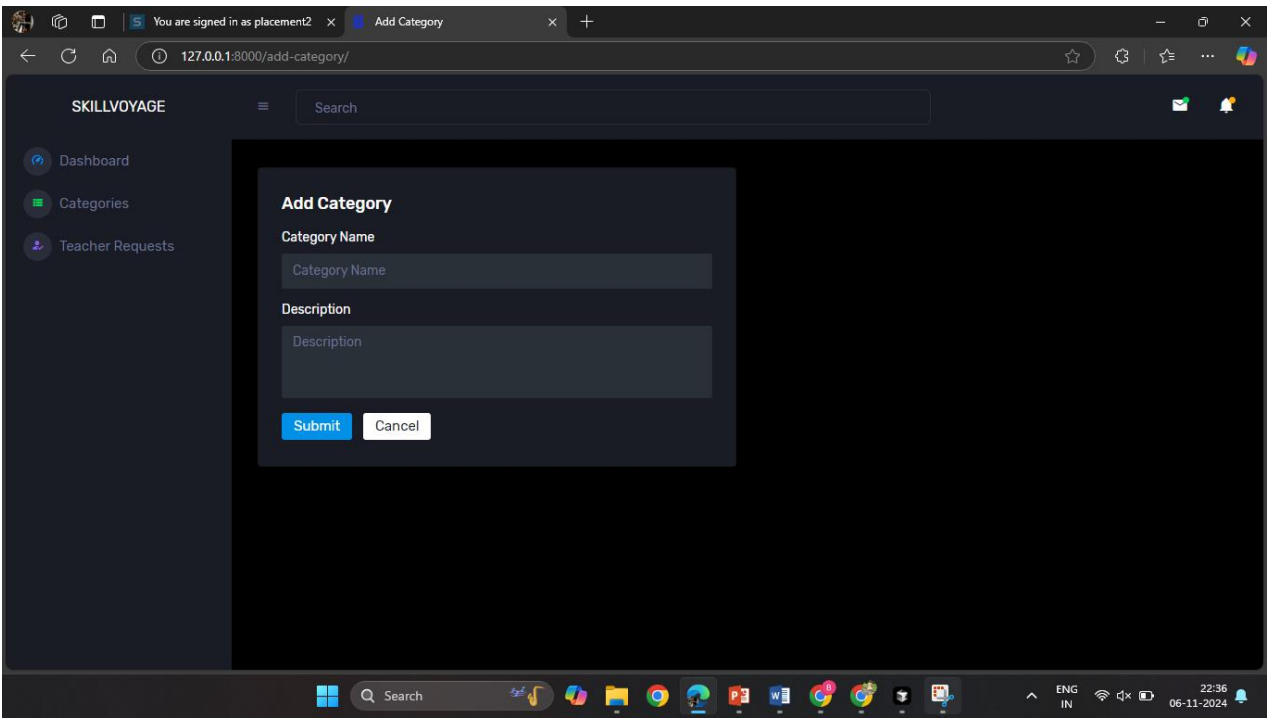


Fig 28. Admin category add page

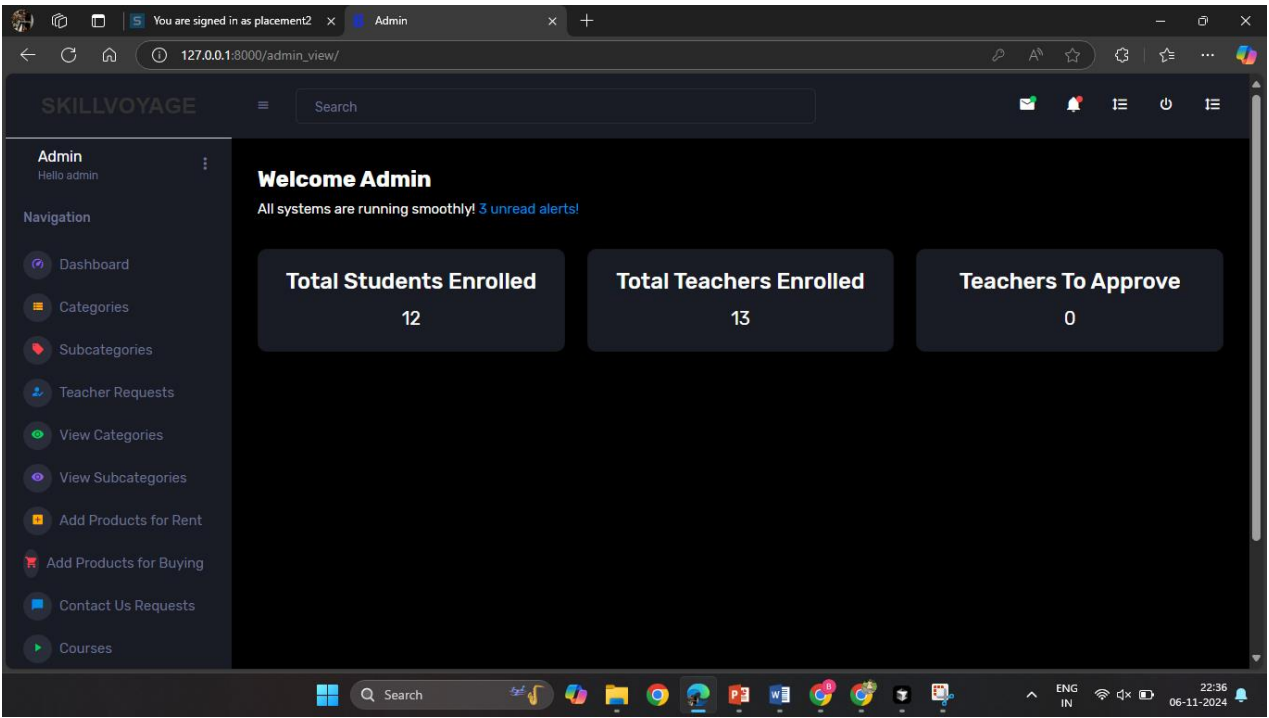


Fig 29. Admin dashboard

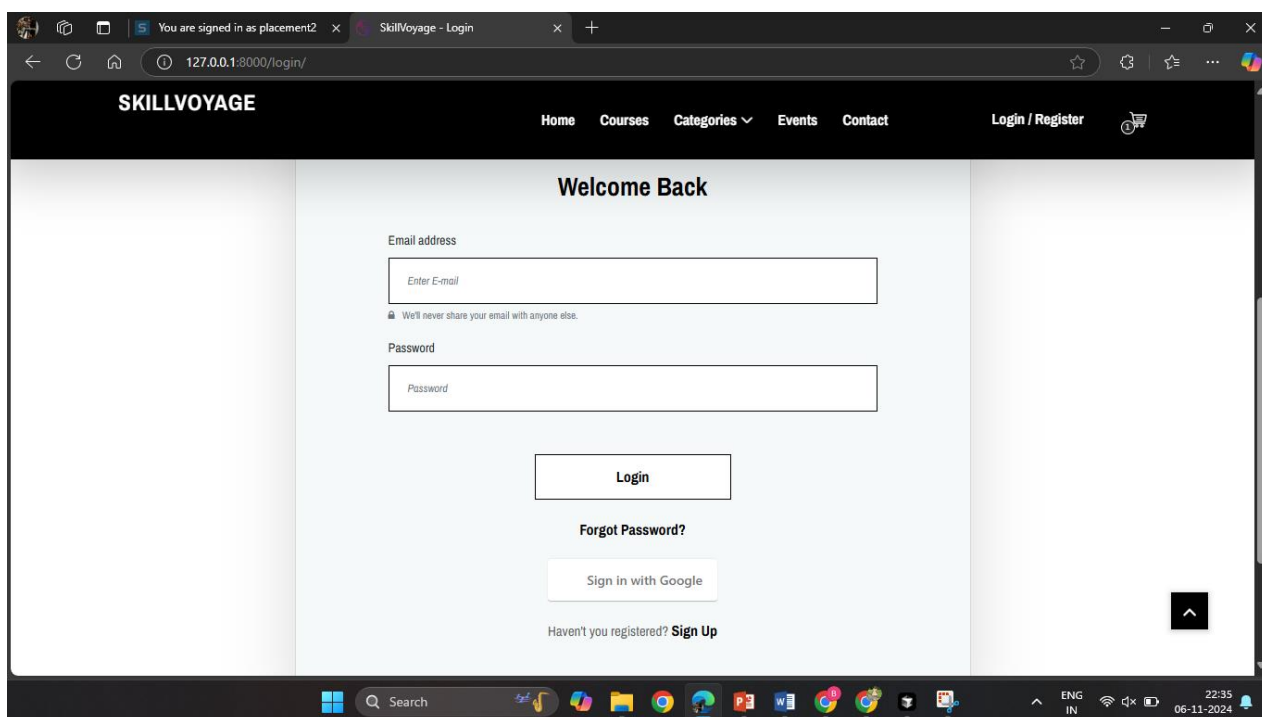


Fig 30. Login page

9.3 GIT LOG

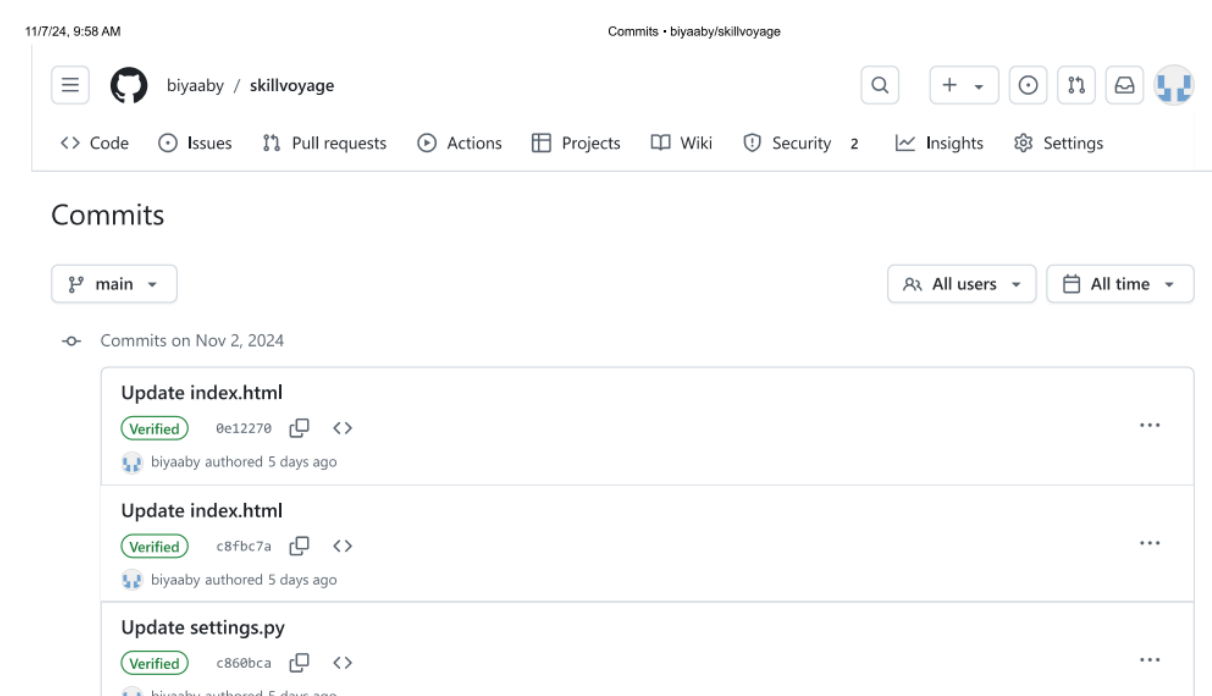


Fig 31. Git commit (1)

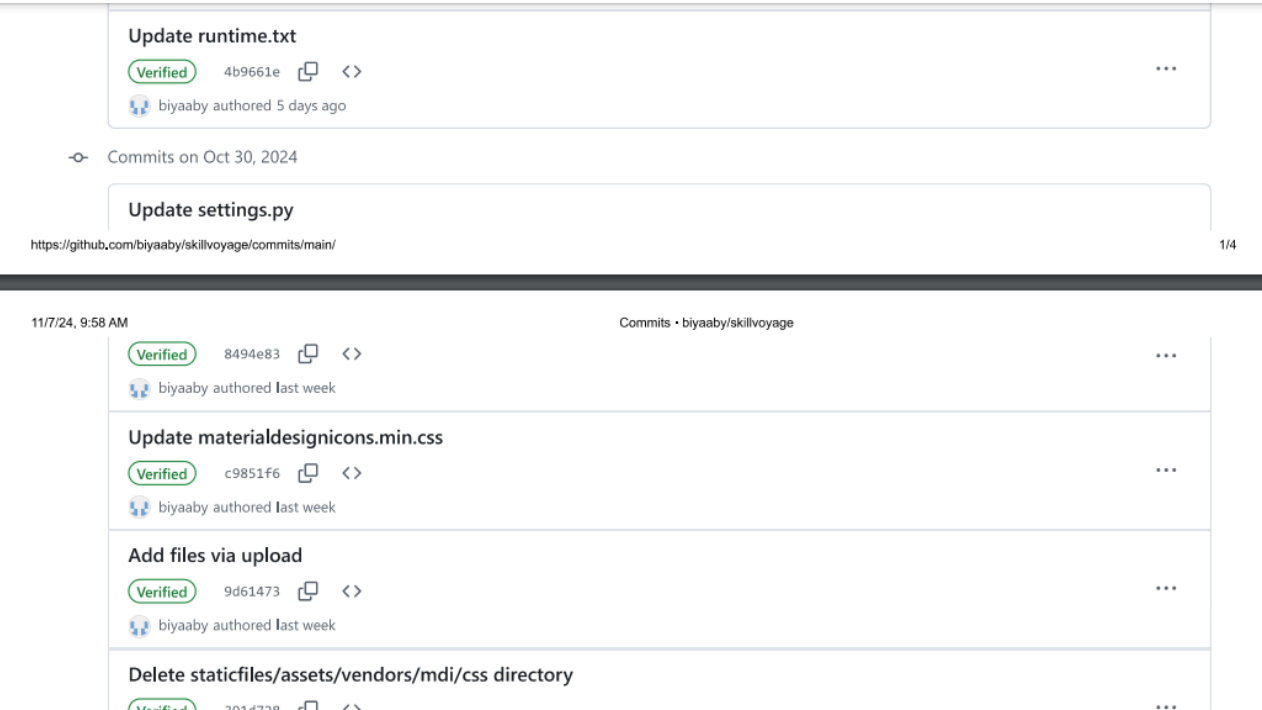


Fig 32. Git commit (2)

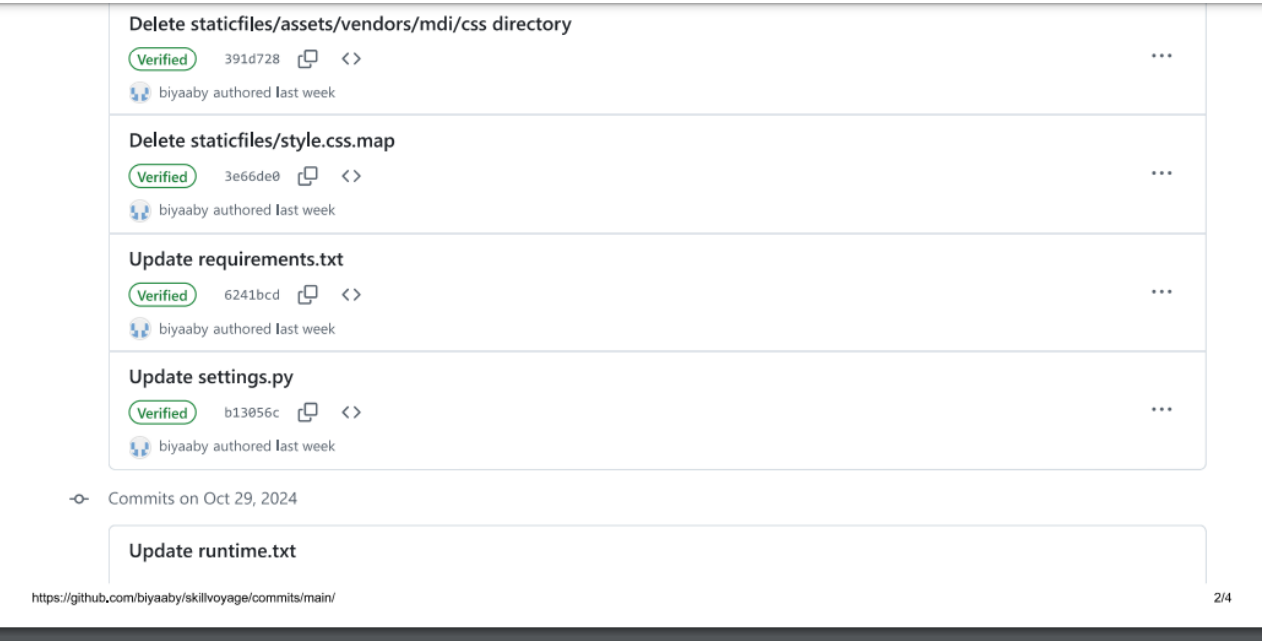


Fig 33. Git commit (3)

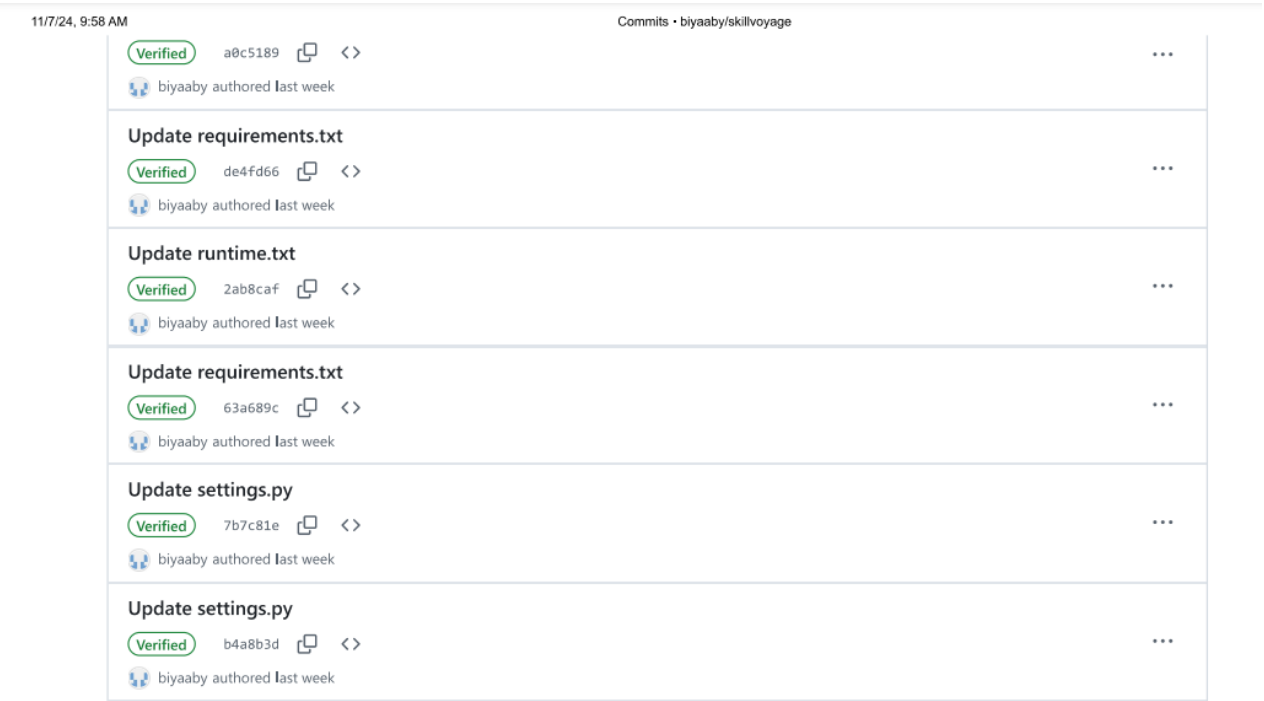


Fig 34. Git commit (4)

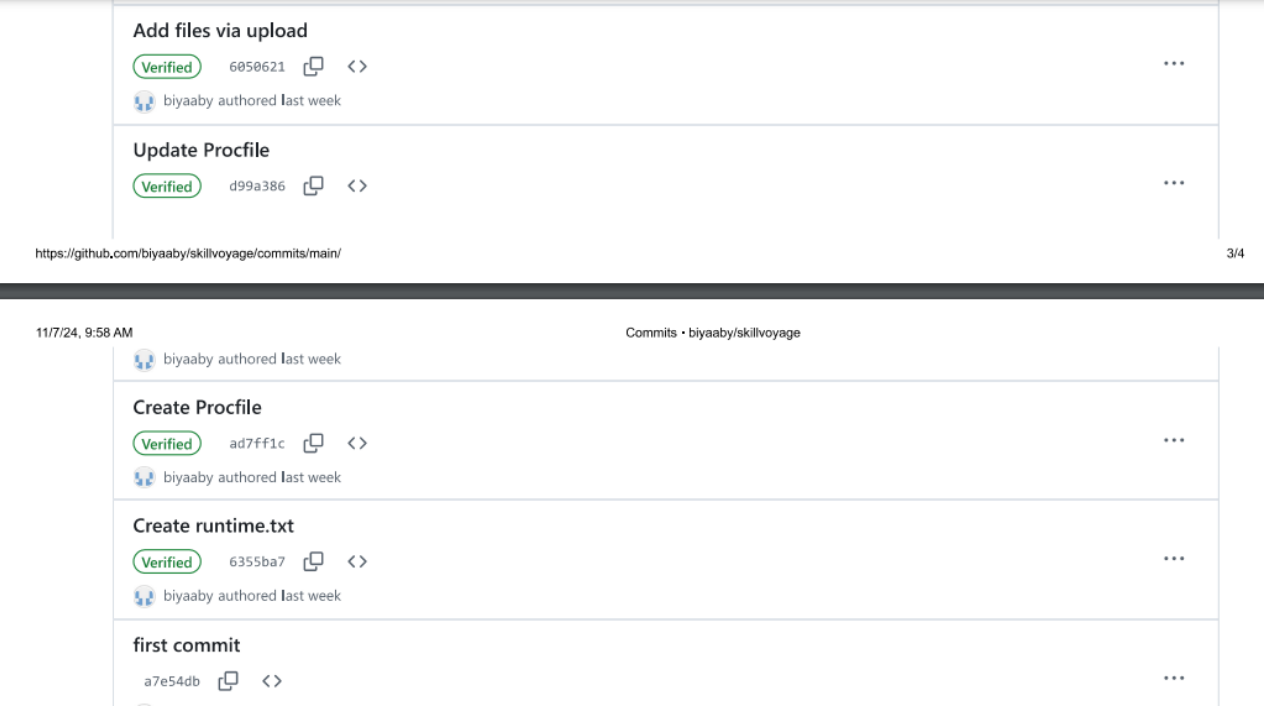


Fig 35. Git commit (5)

