SPORTSHUB
[Details]

Mini Project Report

Submitted by

Vidhu Krishnan Vinod

Reg. No.: AJC19MCA-000

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

KANJIRAPPALLY

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

2023-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**SPORTSHUB"** is the bona fide work of **VIDHU KRISHNAN VINOD (Regno: AJC00MCA-0000)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Guide Name	**Coordinator Name**
Internal Guide	**Coordinator**

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **"SPORTSHUB"** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date:	**STUDENT NAME**
KANJIRAPPALLY	**Reg: AJC00MCA-0000**

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. Dr. Mathew Paikatt** 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 **Coordinator Name** 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 **Guide Name** 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.

VIDHU KRISHNAN VINOD

ABSTRACT

.

The proposed project titled -Sportshub is a comprehensive platform designed for sports enthusiasts and fitness-conscious individuals. Sports Hub provides professional facilities that include the booking space and equipment for sports items such as football, basketball, cricket, tennis, and volleyball for a registered user based on payment. The second category of users consists of fitness enthusiasts who register and use the fitness related resources to achieve their fitness goals. The platform features registered certified trainers who provide personalized training and monitor daily activities as per the schedule and scheme of training type opted by the members with customs provided and specific payment plans. The interface incorporates AI-integrated technology to enable improved fitness monitoring on a regular basis. This innovation is introduced through the cutting-edge virtual assisted gym trainer, powered by Pose Detection ML technology. Users have the option to participate in virtual training sessions, where the AI-powered trainer analyses their workout posture and precision during custom activities in real-time. The system provides alarm feedback on posture violations and offers guidance to optimize performance. Detailed reports are generated to track progress. This advanced feature brings the advantage and convenience of a professional gym trainer, even after the completion of the training program, and can be used directly for home workouts. The proposed system consists of four modules: Admin, Registered Sports User, Registered Gym User, and Registered Trainers. The envisioned project covers the overall system, with a partial functionality of the Admin, Registered Gym User, and Registered Trainer modules being completed in the mini project.

CONTENT

List of Abbreviation

IDE	-	Integrated Development Environment.
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
UML	-	Unified Modeling Language
RDBMS	-	Relational Database Management System

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

SportsHub is a revolutionary platform designed to innovate the sports and fitness experience for enthusiasts and those dedicated to their health. This comprehensive system seamlessly integrates professional sports facility bookings, equipment reservations, personalized fitness training and advanced AI-enabled fitness tracking. Of particular importance is the implementation of MovNet technology, an advanced advancement in precise posture detection that ensures real-time feedback during fitness activities. SportsHub provides users with easy access to booking sports facilities and equipment, including options for football, basketball, cricket, tennis and volleyball, all with unsurpassed convenience. . At the same time, health-conscious people can create personalized workout programs and workout routines, under the supervision of certified trainers. MovNet technology fundamentally transforms fitness monitoring, allowing users to partake in virtual training sessions where their workout posture is meticulously analyzed, offering instant feedback, posture violation alerts, and detailed progress reports. By merging the power of MovNet with tailored user experiences, SportsHub aims to redefine the standard for sports and fitness platforms, setting a new benchmark for accessibility, efficiency, and personalization.

1.2 PROJECT SPECIFICATION

Technology stack:
SportsHub uses an advanced technology stack, including Django as the core framework, HTML for site building, CSS for layout and style and SQLite as a relational database. for effective data management. The integration of MovNet technology enhances posture detection, ensuring accurate analysis when monitoring fitness.
Module details:
Administration module:
The administration module serves as the central platform for system administration. Administrators have authority to manage sports facilities and equipment, update product catalogs, and oversee user management and system configuration.
Registered Sports Users Module:
Suitable for sports enthusiasts, this module simplifies the booking of sports facilities and equipment.
's secure payment options enable a smooth booking process, enhancing the sports experience.

Registered Gym User Module:

Designed for those interested in fitness, this module provides access to a variety of fitness resources.

users can choose from personalized workout programs and secure payment methods for their fitness journey.

Registered Instructors Module:

Certified

Trainers use this module to interact with users, provide personalized fitness plans, and closely monitor progress user program.

Real-time feedback and guidance during training sessions optimizes user performance. Project Goals and Future Developments:

Project

Goals:

SportsHub's main mission is to democratize sports and fitness activities, ensuring accessibility, efficiency and personalization. Leveraging MovNet technology, it aims to provide exceptional user experience and expert advice through real-time fitness tracking, thereby setting new industry standards.

Future expansion and growth:

The initial implementation of SportsHub is the basis for future expansion. Future phases will focus on improving the capabilities of all modules, increasing user engagement, and introducing additional features including improved MovNet integration for detection more comprehensive posture.

SportsHub, with its advanced MovNet posture detection technology, is poised to redefine the sports and fitness landscape, delivering unparalleled user experience, unmatched convenience and expert guidance. This project continues to serve as a solid foundation for future expansion, enhancing the sports and fitness journey of all users.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

Transforming sports and fitness with SportsHub
In an era where technology and personal health are intertwined, SportsHub is emerging as a pioneering platform poised to redefine the sports and fitness experience. Fitness. This groundbreaking project seamlessly integrates professional sports facility bookings, equipment bookings, personal fitness training and advanced AI-integrated fitness tracking, powered by cutting-edge MovNet technology to automatically detect correct postures. SportsHub's goal is to democratize sports and fitness, ensuring accessibility, effectiveness and personalization, setting new industry standards. Through an innovative combination of technology and personalized user experience, SportsHub is not only growing but revolutionizing the sports and fitness landscape, delivering convenience, personalization and advice Great expertise.

2.2 EXISTING SYSTEM

2.2.1 NATURAL SYSTEM STUDIED

The existing system for sports and fitness activities typically relies on traditional methods, often involving manual booking processes for sports facilities and equipment. Users may have limited options for personalized fitness training, and progress monitoring primarily occurs through personal records and sporadic trainer interactions. Advanced AI technology, such as MovNet, is not integrated into the existing system, limiting real-time fitness monitoring capabilities. Access to sports and fitness resources might not be user-friendly, and there may be a lack of centralized, comprehensive platforms that cater to both sports and fitness needs.

2.2.2 DESIGNED SYSTEM STUDIED

The designed system, SportsHub, is meticulously crafted to overcome the limitations of the existing system by introducing automated online booking, personalized fitness programs, real-time AI-integrated monitoring, enhanced safety through precise posture detection, the consolidation of sports and fitness resources, user-friendly digital platforms, and improved accessibility. SportsHub revolutionizes the user experience, making it more efficient and accessible, while also providing comprehensive solutions for sports enthusiasts and fitness-conscious individuals.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Manual booking processes for sports facilities and equipment.
- Limited personalization options for fitness training.
- Lack of real-time fitness monitoring capabilities.
- Safety concerns due to a lack of immediate feedback.
- Fragmented resources across different platforms or locations.
- Lack of user-friendly digital platforms for booking.
- Limited accessibility for a wide range of users.

2.4 PROPOSED SYSTEM

SportsHub is a forward-thinking solution that addresses the limitations of the existing system. It streamlines the process of booking sports facilities and equipment, offering a wide range of options for sports enthusiasts. Additionally, it provides personalized fitness training, enabling users to tailor fitness programs to their specific needs. The integration of MovNet technology allows for precise and real-time pose detection during fitness activities, enhancing safety and performance optimization. The proposed system creates an innovative, user-friendly ecosystem that combines sports and fitness resources, making them accessible, efficient, and tailored to individual needs. It aims to set new industry standards for the sports and fitness experience by offering convenience, personalization, and expert guidance.

In summary, SportsHub represents a significant leap forward from the limitations of the existing system, utilizing technology, personalization, and advanced fitness monitoring to create a comprehensive and transformative platform for sports and fitness enthusiasts.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Efficient online booking.
- Personalized fitness training.
- Real-time fitness monitoring.
- Enhanced safety features.
- Comprehensive resource access.
- User-friendly digital interface.
- Inclusivity for a wider user base.
- Scalability and future expansion.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility refers to the practicality of successfully executing a project. To evaluate the feasibility of a project, a feasibility study is conducted, which assesses the practicality and workability of the
proposed solution to meet the requirements of the software. The feasibility study takes into account
various factors such as resource availability, cost estimation for software development, the benefits
of the software to the organization after its development, and the cost of maintaining the software.
The findings of the feasibility study should be presented in a report that recommends whether or not to proceed with the requirements engineering and system development process. The primary objective of the feasibility study is to determine the reasons for developing software that is acceptable to users, adaptable to change, and compliant with established standards.
Other objectives of the feasibility study are listed below:
• To analyse whether the software will meet organizational requirements
• To determine whether the software can be implemented using the current technology and within
the specified budget and schedule
• To determine whether the software can be integrated with other existing software.

3.1.1 Economical Feasibility

Economic feasibility for the SportsHub project is a fundamental assessment of its financial viability, encompassing a comprehensive cost-benefit analysis, revenue projections, break-even analysis, and return on investment calculation. This analysis aims to determine whether the project's expected financial benefits, including potential revenue streams from user subscriptions, equipment bookings, and trainer fees, outweigh the initial development and ongoing operational costs. By evaluating potential cost reduction benefits, conducting risk assessments, and performing sensitivity analyses, the project's economic feasibility is meticulously examined to inform stakeholders about its financial prospects and to guide sound decision-making regarding its implementation.

3.1.2 Technical Feasibility

Technical feasibility analysis for the SportsHub project is a critical examination of the system's technical viability, encompassing an evaluation of the chosen technology stack, the integration of MovNet technology for precise pose detection, the availability of development expertise, data security measures, scalability considerations, and potential technical risks. This assessment aims to ensure that the project can be efficiently developed, operated, and maintained, utilizing the selected technologies and technical resources. By addressing compatibility, security, scalability, and potential challenges, the technical feasibility analysis provides stakeholders with a clear understanding of the project's readiness for implementation and any necessary technical adaptations to ensure its success.

3.1.3 Behavioral Feasibility

The SportsHub Project's behavioral feasibility analysis is a critical assessment of the acceptability and adoption of the system within the user community, including considerations of users' willingness to adopt the system. new platform adoption, behavior change management strategy, user experience, feedback integration, cultural alignment, training . and stakeholder engagement. This analysis ensures that the project aligns with user needs and preferences, facilitates a smooth transition, and provides a positive user experience. By addressing both the human and organizational aspects, the Behavioral Feasibility Assessment informs stakeholders about user and organizational readiness to successfully adopt and integrate the SportsHub system into their exercise and sports routines.

3.1.4 Feasibility Study Questionnaire

1. Do you have access to outdoor fields, courts, or tracks for practicing sports, or are they limited?

Ans: Yes, good quality is also rare

2. How easy or difficult is it to book or reserve sports facilities for personal training or practice sessions?

Ans: booking not hard, but others are difficult.

3. Are there any local clubs or teams you can join to participate in organized sports competitions or
leagues?

Ans: yes, but there also they can't afford to access professional facilities are rare.

4. Have you ever faced challenges in accessing sports or fitness facilities due to factors like distance, cost, or availability?

Yes, mostly distance is prominent for good quality facility.

5. Are there any initiatives or programs in your community that promote sports and fitness for teenagers?

None

6. How do you cope with the frustration of not having access to specific gym equipment during your workout sessions?

Yes, overcrowding is there

4

7. Do you believe that the fitness center should take measures to address equipment shortages, and if so, what suggestions do you have for improvement?

yes

8. Have you ever discussed the issue of equipment shortage with other fitness enthusiasts at the gym, and what are their thoughts on the matter?

Proper management can overcome that problem.

9. If there were a system for reserving gym equipment during busy hours, would you find it helpful, or do you prefer a first-come, first-serve approach?

Yes would be helpful.

10. Do you think the shortage of gym equipment affects the overall quality of your fitness center experience? How could this be improved?

Yes it can be improved

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	intel i5
RAM	-	8 g b r a m
Hard disk	-	1 T b

3.2.2 Software Specification

Front End	-	HTML, CSS, JavaScript
Back End	-	Python (Django framework)
Database	-	SQLite
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, CSS,MoveNet

3.3 SOFTWARE DESCRIPTION

3.3.1 HTML

HTML, short for Hypertext Markup Language, serves as the basic language for creating web pages. It provides a set of essential elements that structure and give meaning to the document's content. As part of the SportsHub project, HTML5 is used to design websites. HTML not only helps organize content into logical sections, but also allows for the inclusion of multimedia elements, such as images and videos. Additionally, HTML validation is used to maintain code quality, ensuring proper compliance with web standards and guidelines.

3.3.2 CSS

CSS, or cascading style sheets, play a central role in defining the layout and visual style of website content. While HTML defines the structure and semantic meaning of content, CSS manages its appearance. It allows customization of various aspects, including fonts, colors, sizes, spacing, and visual effects. In the SportsHub project, CSS3 is integrated to style web pages and create animations that enhance the visual appeal of the website. CSS is an essential technology that ensures a consistent and visually appealing user experience.

3.3.3 Python (Django Framework)

Python, a flexible and powerful programming language, is used as the back-end language for the SportsHub project. Specifically, the Django web framework is used to support the development of robust, scalable, and secure web applications. Django simplifies various aspects of web development, including user authentication, database management, and URL routing. It follows the Model-View-Controller (MVC) architectural pattern, which prioritizes clean and maintainable code. Python, combined with Django, forms the backbone of the system, allowing for efficient data processing and seamless execution of business logic.

3.3.4 SQLite

Although not used in this project, it is worth mentioning that SQLite is a lightweight and efficient "RDBMS" relational database management system; Widely used for small and medium applications. It excels in situations that require compact, self-contained databases. However, in the SportsHub project, the more powerful MySQL is used as the RDBMS. MySQL offers scalability, speed, and reliability, making it suitable for managing system data effectively.

3.3.5 MovNet

The integration of MovNet technology represents an advanced addition to the SportsHub project. MovNet, an advanced solution, specializes in precise posture detection. It enables real-time analysis of users' exercise postures and provides valuable feedback when tracking physical activity. This technology improves user experience by providing instant information and guidance, thereby promoting consistent and effective training techniques. The system leverages MovNet to ensure accurate progress tracking and physical activity tracking, which distinguishes it as a modern and innovative platform.

CHAPTER 4
SYSTEM DESIGN

4.1INTRODUCTION

Any engineered system or product's development process begins with design. A creative process is design. The secret to an efficient system is a decent design. The process of using different methodologies and concepts to specify a process or a system in enough detail to allow for its physical realization is referred to as "design." One way to describe it is as the process of using different methodologies and concepts to specify a device, a process, or a system in enough detail to allow for its physical realization. Regardless of the development paradigm that is employed, software design forms the technical core of the software engineering process. The architectural details required to build a system or product are developed during the system design process. This program has also gone through the best possible design phase, fine-tuning all levels of efficiency, performance,

and accuracy, just like any system engineering. A user-facing document is converted into a document for programmers or database staff during the design phase. The two stages of system design development are logical design and physical design.

4.2 UML DIAGRAM

The standardised language known as UML, which is used for the purposes of designing, developing, visualising, and describing software system products, was created by the Object Management Group (OMG). The UML 1.0 standard's initial draught was published in January 1997, and since then it has been widely utilised as a visual language for describing and developing software systems. In contrast to programming languages like Java, C++, and COBOL, UML is used to depict the flow and interactions of a system's components through diagrams rather than writing code. Although UML is most frequently used for software system design, it can also be applied to non-software systems, such as the process flow in a manufacturing facility. With the use of accessible tools, UML diagrams may be translated into code for a number of computer languages. UML is now recognized as a standard by OMG due to its successful standardisation and tight relationship to the analysis and design of object-oriented systems.

The nine different UML diagrams are:

- 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 graphical representation that demonstrates the relationships between the functions of various system components. It is a method for determining, outlining, and organising the system's requirements, such as the necessity for a website for ordering goods and services for mailing. Use case diagrams are one of the tools used by the popular modelling language UML to describe the construction and functioning of systems and objects seen in the real world. The tasks that make up a system's objectives could range from defining the general needs to approving hardware designs, testing and troubleshooting currently under development software products, creating online help resources, and carrying out customer service-related tasks. For instance, in the context of a scenario involving the selling of goods, use cases can include responding to customer service requests, processing product orders, maintaining product catalogues, and controlling payment processing. Typically, a use case diagram has four essential components.

- The wall separating the studied system from its surroundings.
- The actors are often participants in the system who are identified by their roles.
- The actors within and outside of the system carry out the use cases, which are specialist roles.
- The conversations and interactions between the participants and the use cases.

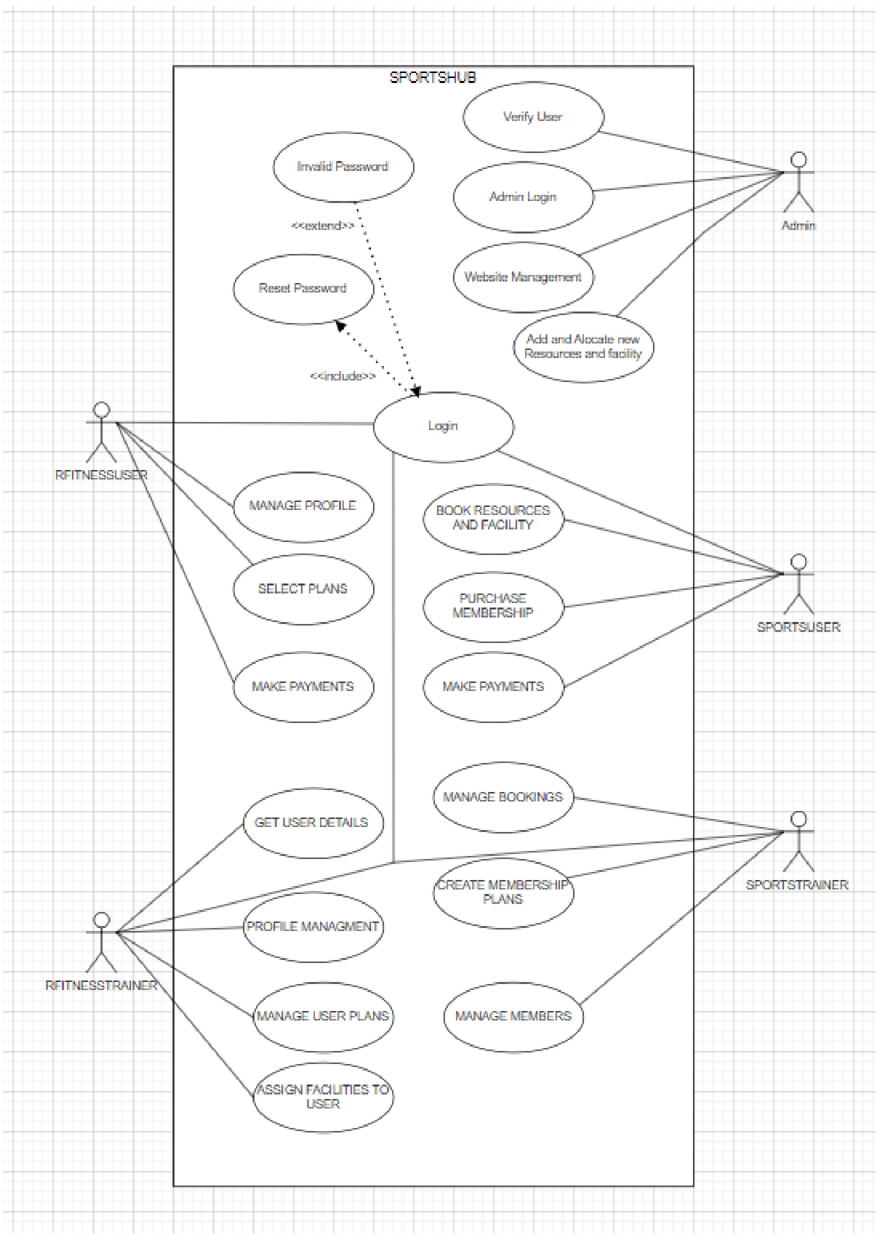


Figure 1: Use Case Diagram

4.2.2 SEQUENCE DIAGRAM

A sequence diagram is a visual representation of the interaction between objects in a specific order. It can also be referred to as an event diagram or event scenario. The purpose of a sequence diagram is to illustrate the order in which objects in a system perform their functions. These diagrams are frequently utilized by software developers and business professionals to document and comprehend the requirements for both new and existing systems.

Sequence Diagram Notations –

I. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

II. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

III. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

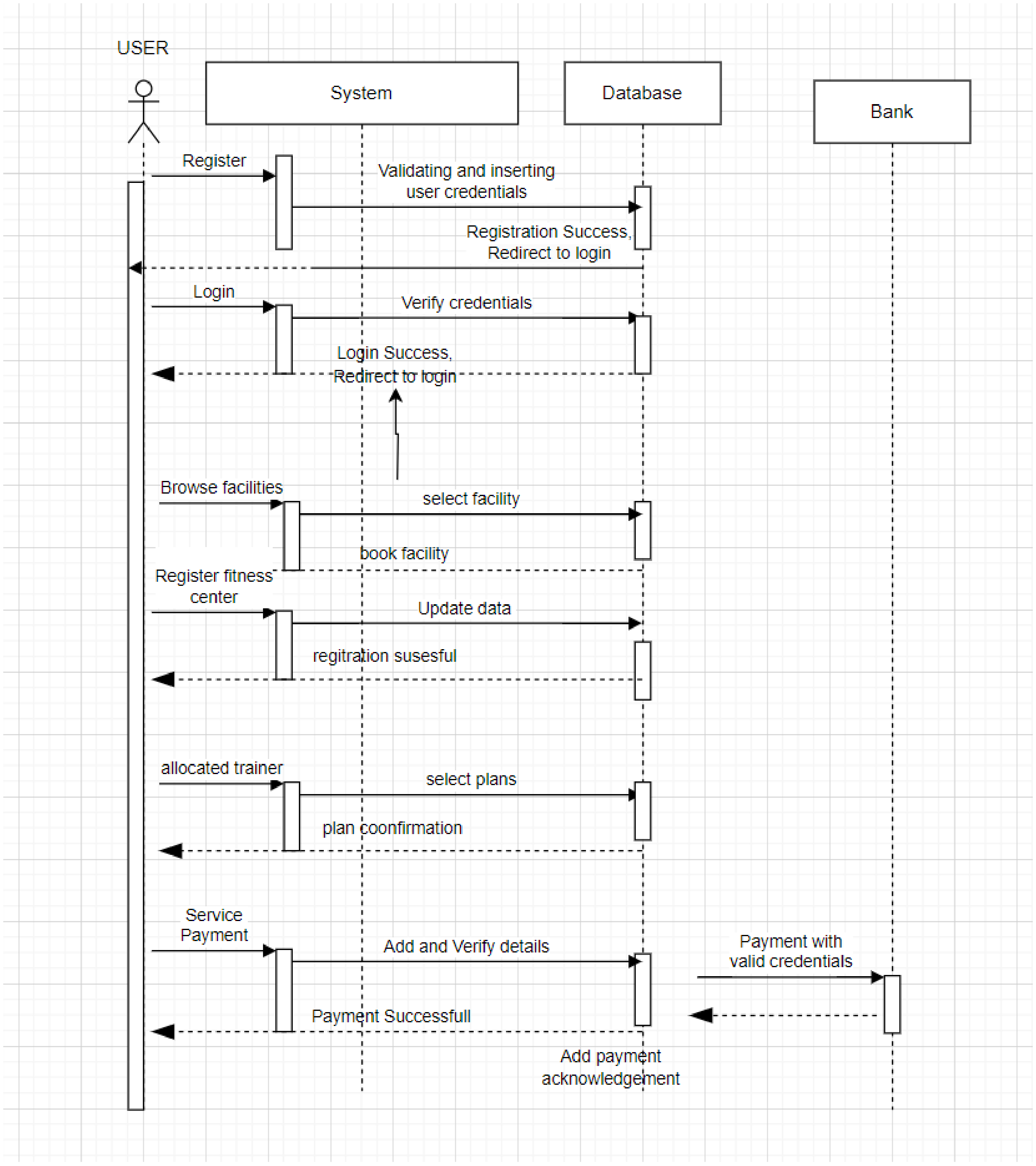
Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

IV. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.



4.2.3 State Chart Diagram

The state machine diagram is also called the State chart or State Transition diagram, which shows the order of states underwent by an object within the system. It captures the software system's behavior. It models the behavior of a class, a subsystem, a package, and a complete system. It tends out to be an efficient way of modeling the interactions and collaborations in the external entities and the system. It models event-based systems to handle the state of an object. It also defines several distinct states of a component within the system. Each object/component has a specific state.

Notation of a State Machine Diagram

Following are the notations of a state machine diagram enlisted below:

a) **Initial state:** It defines the initial state (beginning) of a system, and it is represented by a black filled circle.

b) **Final state:** It represents the final state (end) of a system. It is denoted by a filled circle present within a circle.

c) **Decision box:** It is of diamond shape that represents the decisions to be made on the basis of an evaluated guard.

d) **Transition:** A change of control from one state to another due to the occurrence of some event is termed as a transition. It is represented by an arrow labeled with an event due to which the change has ensued.

e) **State box:** It depicts the conditions or circumstances of a particular object of a class at a specific point of time. A rectangle with round corners is used to represent the state box.

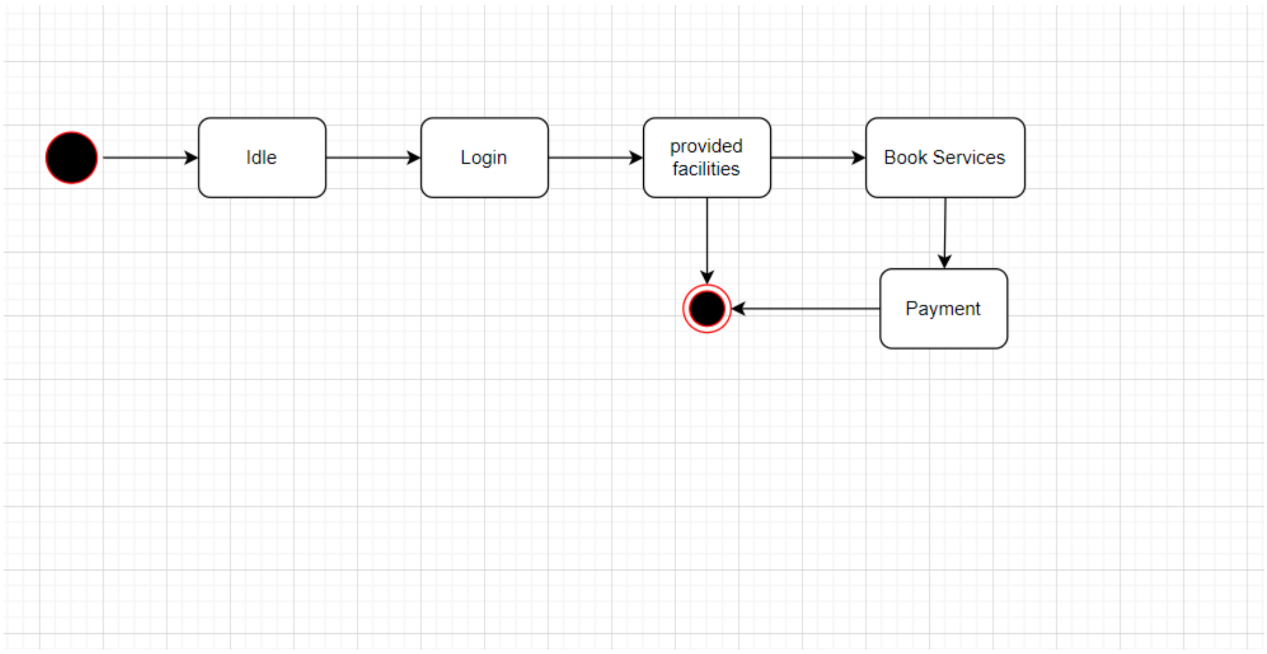


Figure 3: State Chart Diagram

4.2.4 Activity Diagram

An activity diagram in UML is used to illustrate the flow of control within a system, focusing on the workflow between activities rather than the implementation. It can depict both concurrent and sequential activities, and emphasizes the order and conditions of the flow. The diagram includes features such as fork and join to handle different types of flow. It is often referred to as an object- oriented flowchart and is composed of a set of actions or operations to model the behavioral diagram.

Components of an Activity Diagram

a) **Initial Node:** This node marks the starting point of the activity diagram.

b) **Activity States:** These states represent the various activities involved in a process. They are depicted by rounded rectangles.

c) **Decision Nodes:** Decision nodes represent a point in the activity diagram where the flow can take different paths depending on some condition.

d) **Fork Nodes:** Fork nodes indicate parallel processing of activities.

e) **Join Nodes:** Join nodes are used to merge parallel processing flows back into a single flow.

f) **Final Nodes:** Final nodes indicate the end of an activity diagram.

g) **Control Flows:** Control flows represent the order in which the activities are performed and are represented by arrows connecting the various nodes.

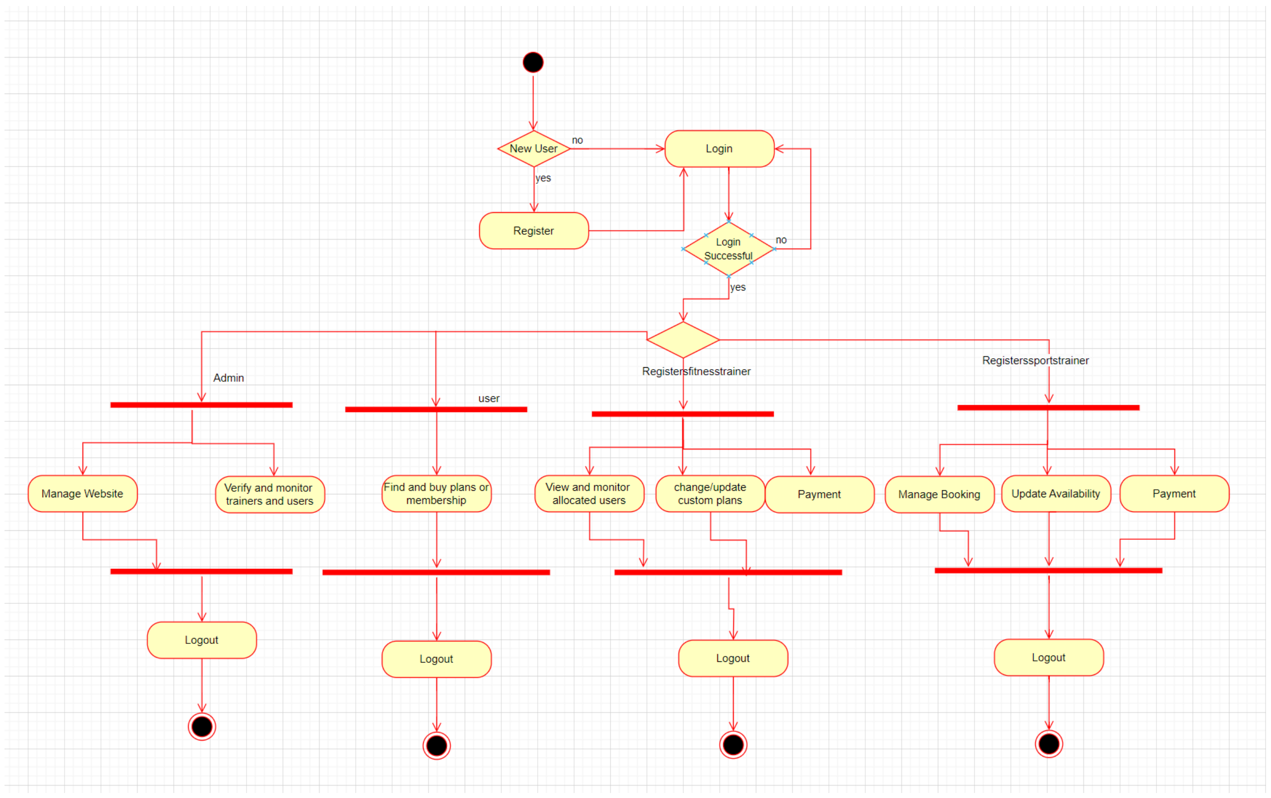
h) **Action States:** Action states represent the actions or operations performed during an activity and are represented by rectangles with rounded edges.

i) **Swimlanes:** Swimlanes are used to group activities according to their responsible parties or departments, making it easier to understand the flow of work.

Notation of an Activity diagram

Activity diagram constitutes following notations:

1. **Initial State:** It depicts the initial stage or beginning of the set of actions.
2. **Final State:** It is the stage where all the control flows and object flows end.
3. **Decision Box:** It makes sure that the control flow or object flow will follow only one path.
4. **Action Box:** It represents the set of actions that are to be performed.



4.2.5 Class Diagram

A static view of a programme is shown in the class diagram. The many item types that the system contains, as well as any links between them, are displayed. Despite the fact that a class can also derive from other classes, a class is only as good as its objects. Class diagrams are used to depict, describe, and document a variety of system components. They are also employed to create executable software code. A structural diagram gives an overall picture of the architecture of a software system by displaying the classes, relationships, properties, and functionalities of the system. In order to support the creation of software applications in a given domain, it helps to collect information about class names, properties, and capabilities.

Components of a Class Diagram

The class diagram is made up of three sections:

Upper Section: The upper section encompasses the name of the class. A class is representation of similar objects that shares the same relationships, attributes, operations, and semantics. Some of the following rules that should be taken into account while representing a class are given below:

a) Capitalize the initial letter of the class name.

b) Place the class name in the center of the upper section.

c) A class name must be written in bold format.

d) The name of the abstract class should be written in italics format.

Middle Section: The middle section constitutes the attributes, which describe the quality of class. •
The attributes have the following characteristics:

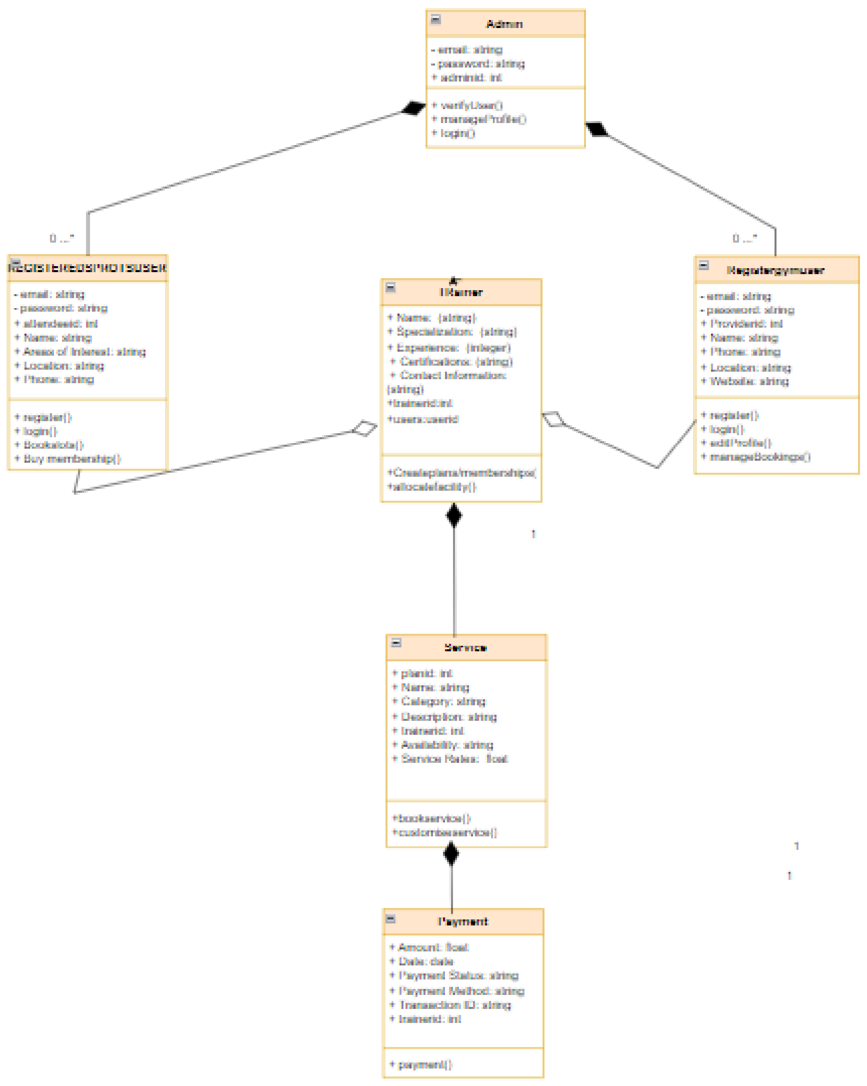
- The attributes are written along with its visibility factors, which are public (+), private (-), protected (#), and package (~).
- The accessibility of an attribute class is illustrated by the visibility factors.
- A meaningful name should be assigned to the attribute, which will explain its usage inside the class.

Lower Section: The lower section contain methods or operations. The methods are represented in the form of a list, where each method is written in a single line. It demonstrates how a class interacts with data.

Relationships

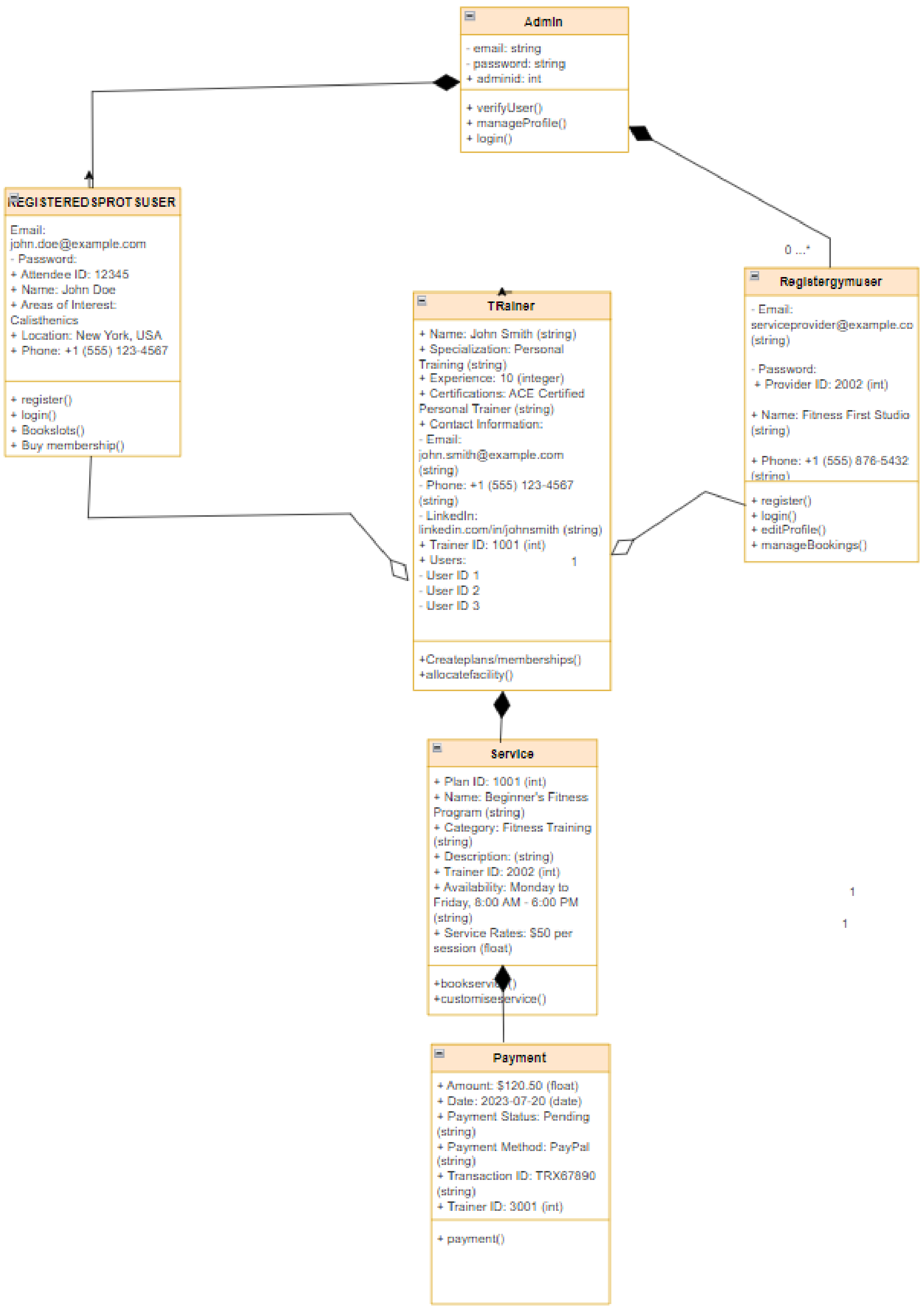
In UML, relationships are of three types:

- **Dependency:** A dependency is a semantic relationship between two or more classes where a change in one class cause changes in another class. It forms a weaker relationship.
- **Generalization:** A generalization is a relationship between a parent class (superclass) and a child class (subclass). In this, the child class is inherited from the parent class.
- **Association:** It describes a static or physical connection between two or more objects. It depicts how many objects are there in the relationship.



4.2.6 Object Diagram

Object diagrams rely on the class diagram as they are based on the class diagram. They illustrate an occurrence of a class diagram by representing an object. Objects provide a static representation of an object-oriented system at a particular moment. The object and class diagrams are similar in some ways, but the class diagram presents a conceptual view of a system, giving a visual representation of a specific function of a system.



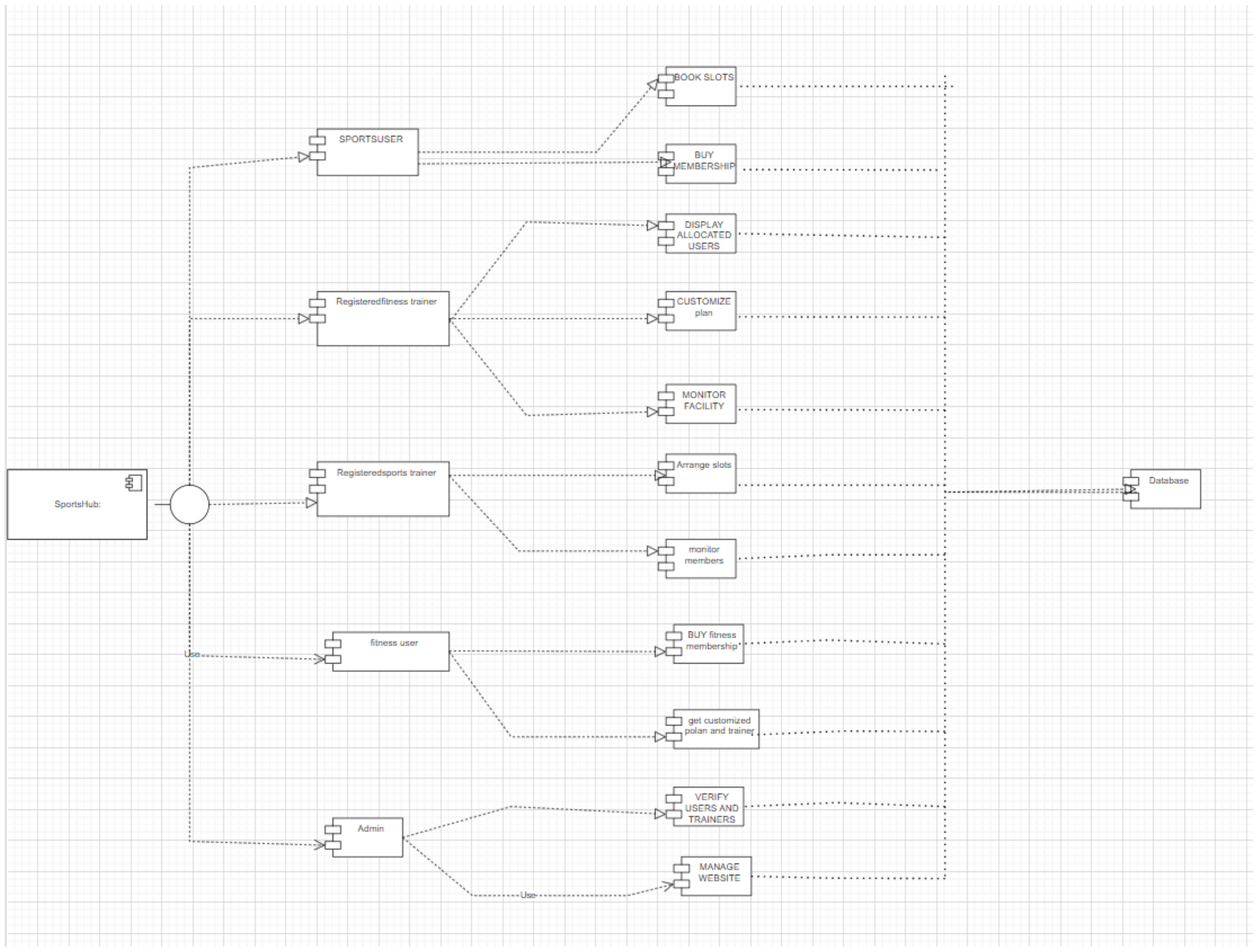
4.2.7 Component Diagram

To make a complex object-oriented system more understandable, the smaller components are separated out using a component diagram. It simulates the physical view of a system, including its internal node's executables, files, libraries, etc. It depicts the connections and hierarchies that exist between the system's components. It aids in creating an operational system. An individual, replaceable, and executable system unit is referred to as a component. A component's implementation details are concealed, therefore an interface is required to carry out a function. It functions like a "black box," with the provided and necessary interfaces explaining its behaviour.

Notation of a Component Diagram

a) A component

b) A node



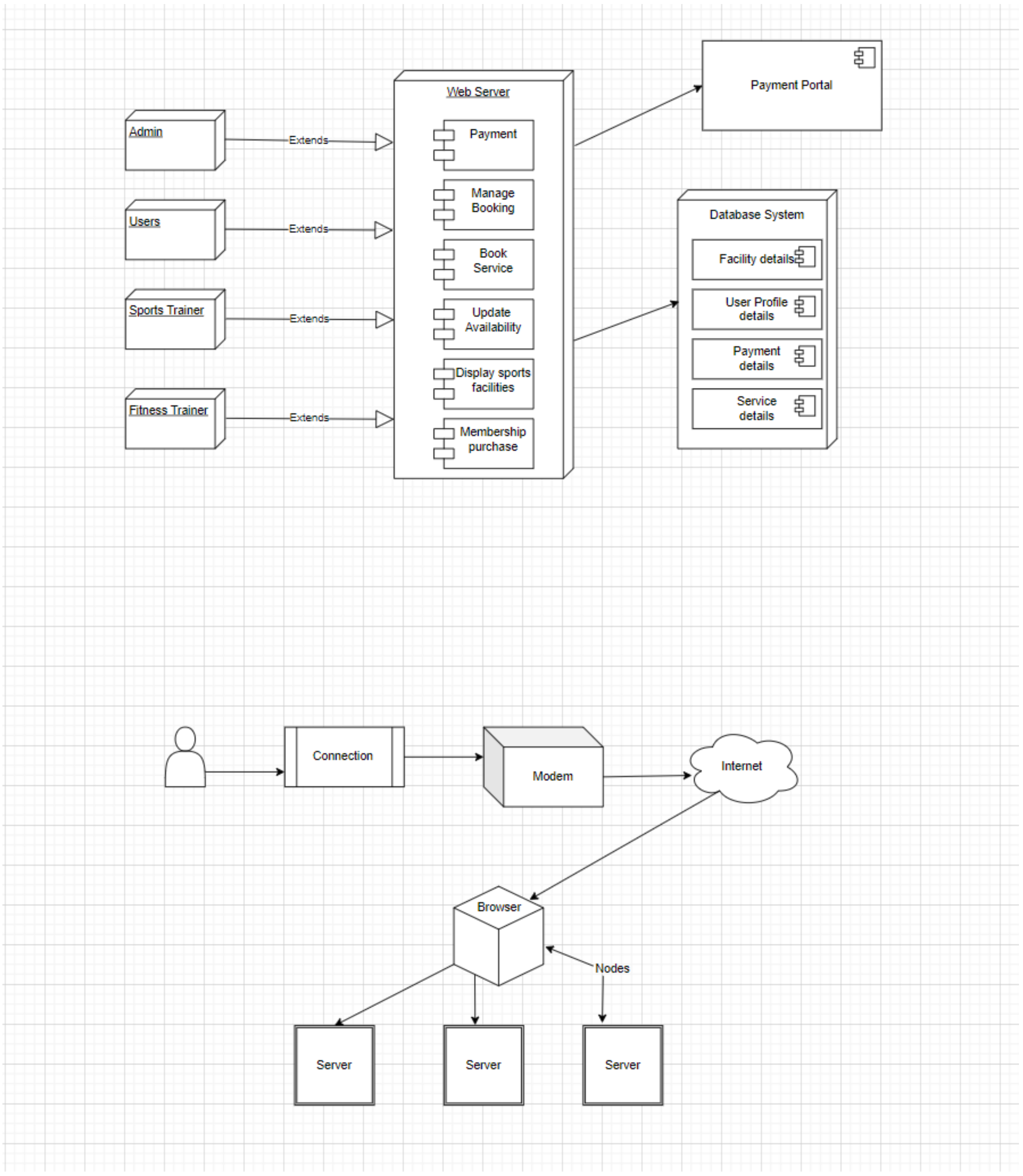
4.2.8 Deployment Diagram

A deployment diagram is a type of UML diagram that shows the physical hardware or system architecture on which software applications will be deployed. It provides a static view of the system deployment and includes nodes and their relationships. The diagram illustrates how the software components are distributed and deployed on the physical hardware or nodes. The diagram is useful for mapping the software architecture designed to the physical system architecture, where the software will be executed. Communication channels are used to show the relationships between the nodes involved.

Notation of Deployment diagram

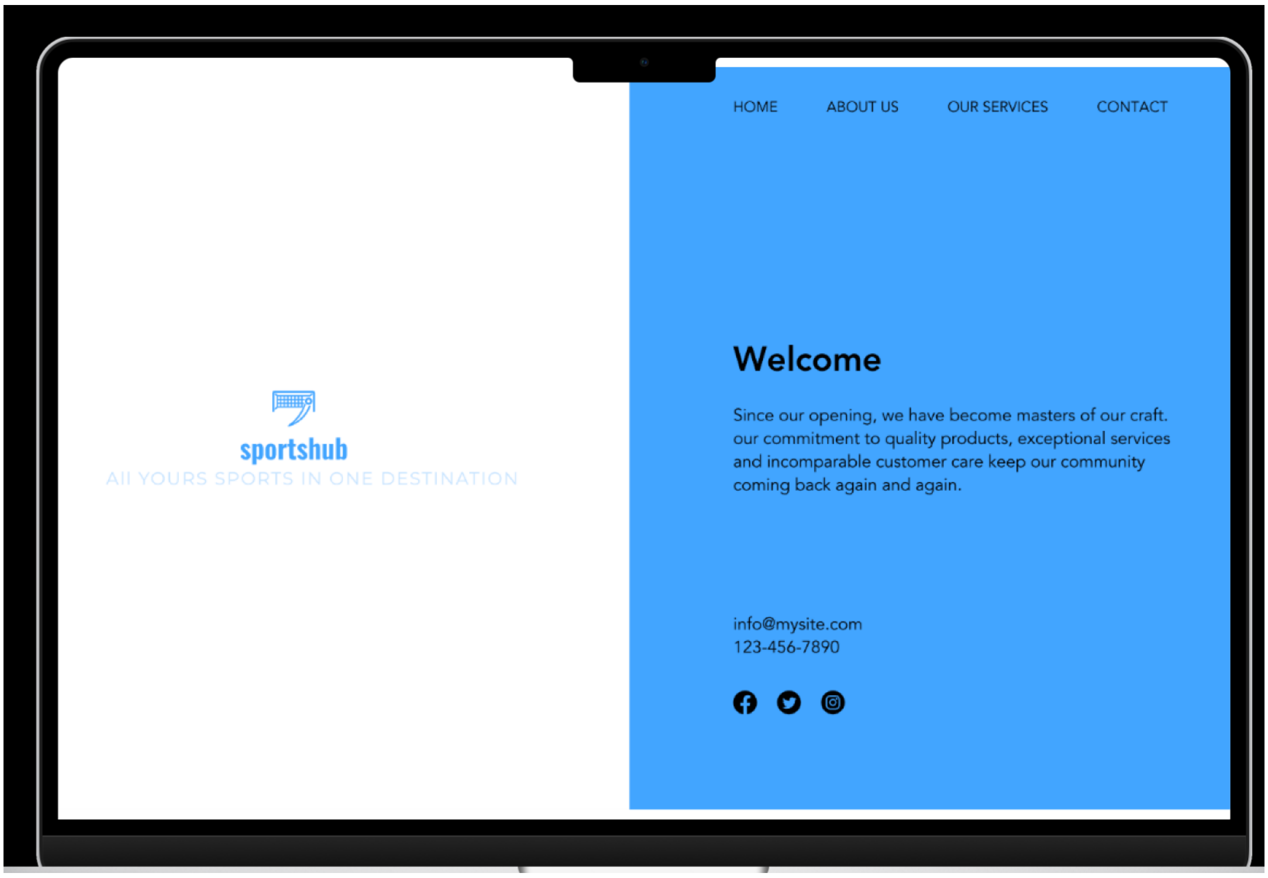
The deployment diagram consists of the following notations:

1. A component
2. An artifact
3. An interface
4. A node



4.3 USER INTERFACE DESIGN USING FIGMA

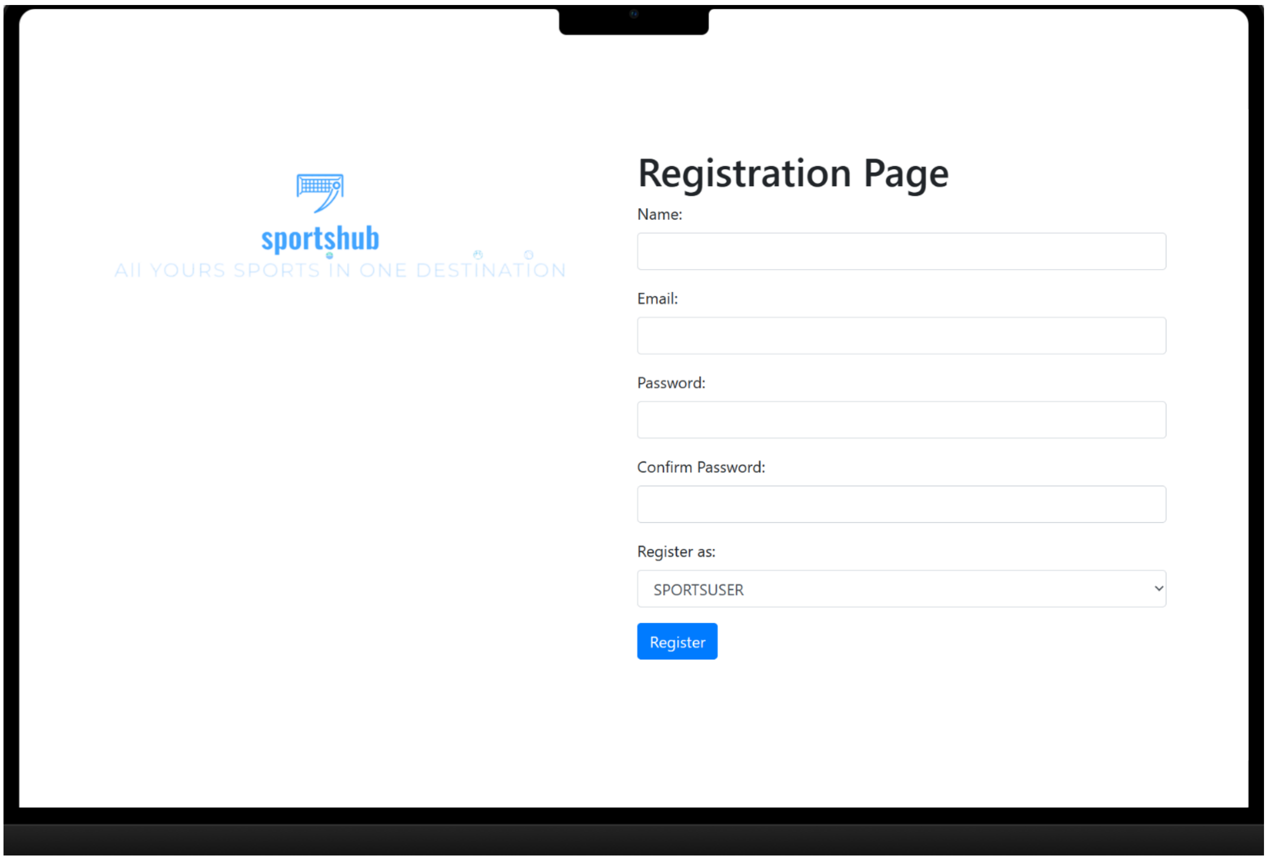
Form Name: index



Welcome

Since our opening, we have become masters of our craft. our commitment to quality products, exceptional services and incomparable customer care keep our community coming back again and again.

info@mysite.com
123-456-7890



sportshub

All YOURS SPORTS IN ONE DESTINATION

Registration Page

Name:

Email:

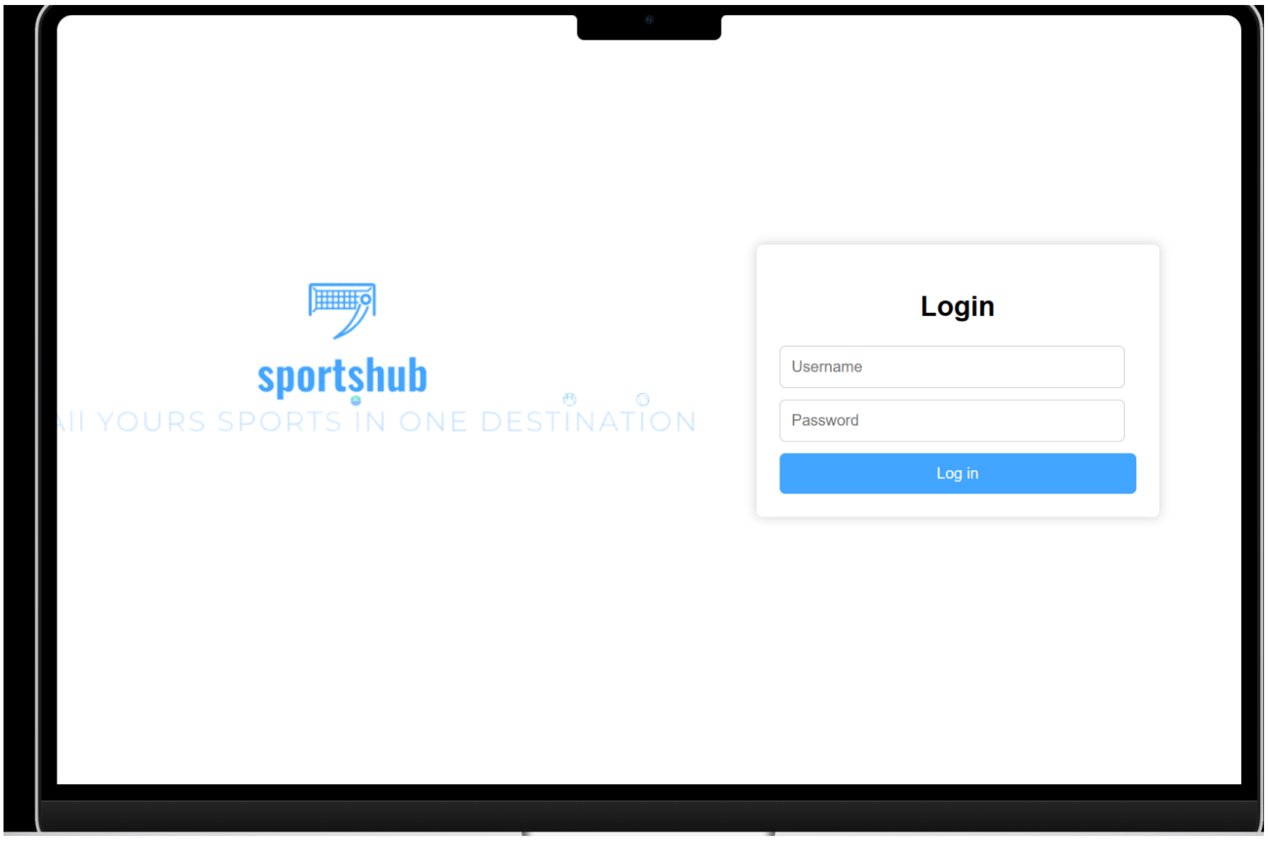
Password:

Confirm Password:

Register as:

SPORTSUSER

Form Name: Registration

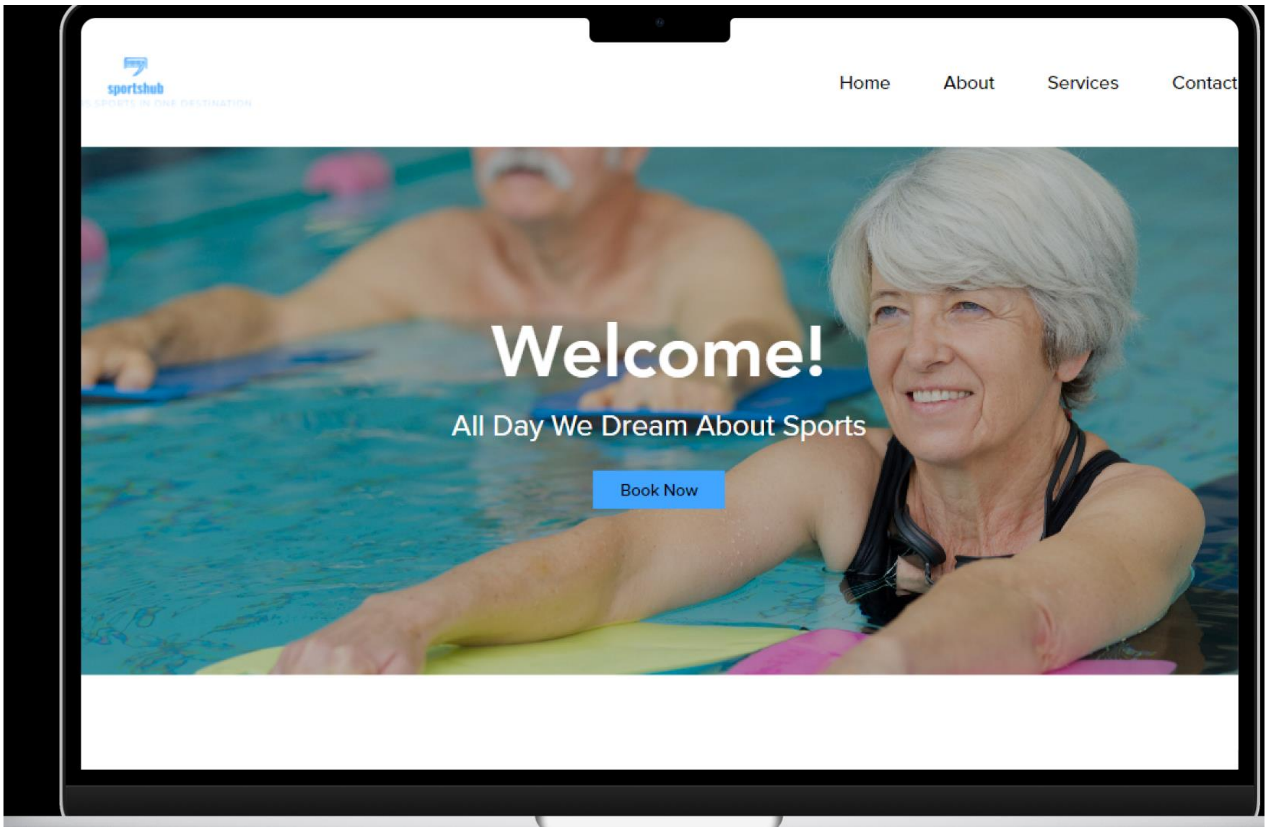


sportshub

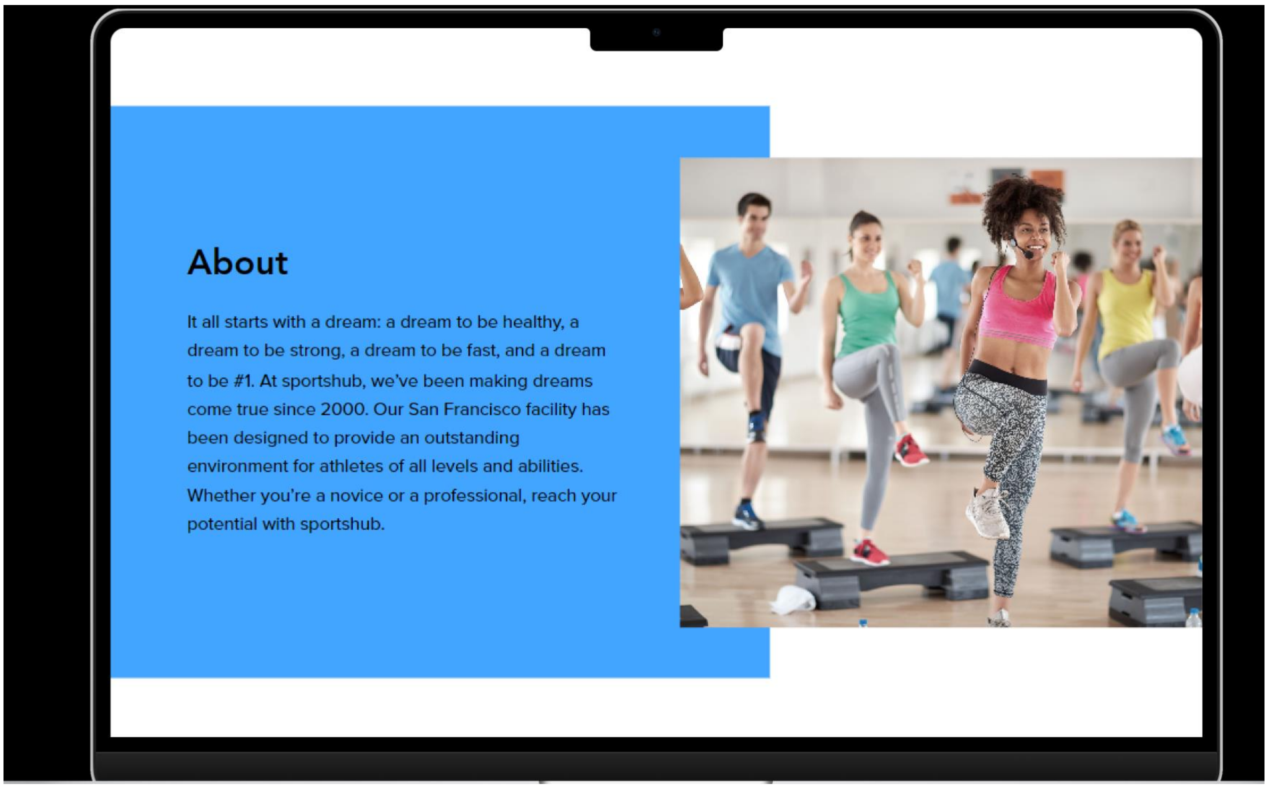
All YOURS SPORTS IN ONE DESTINATION

Login

Form Name: Login

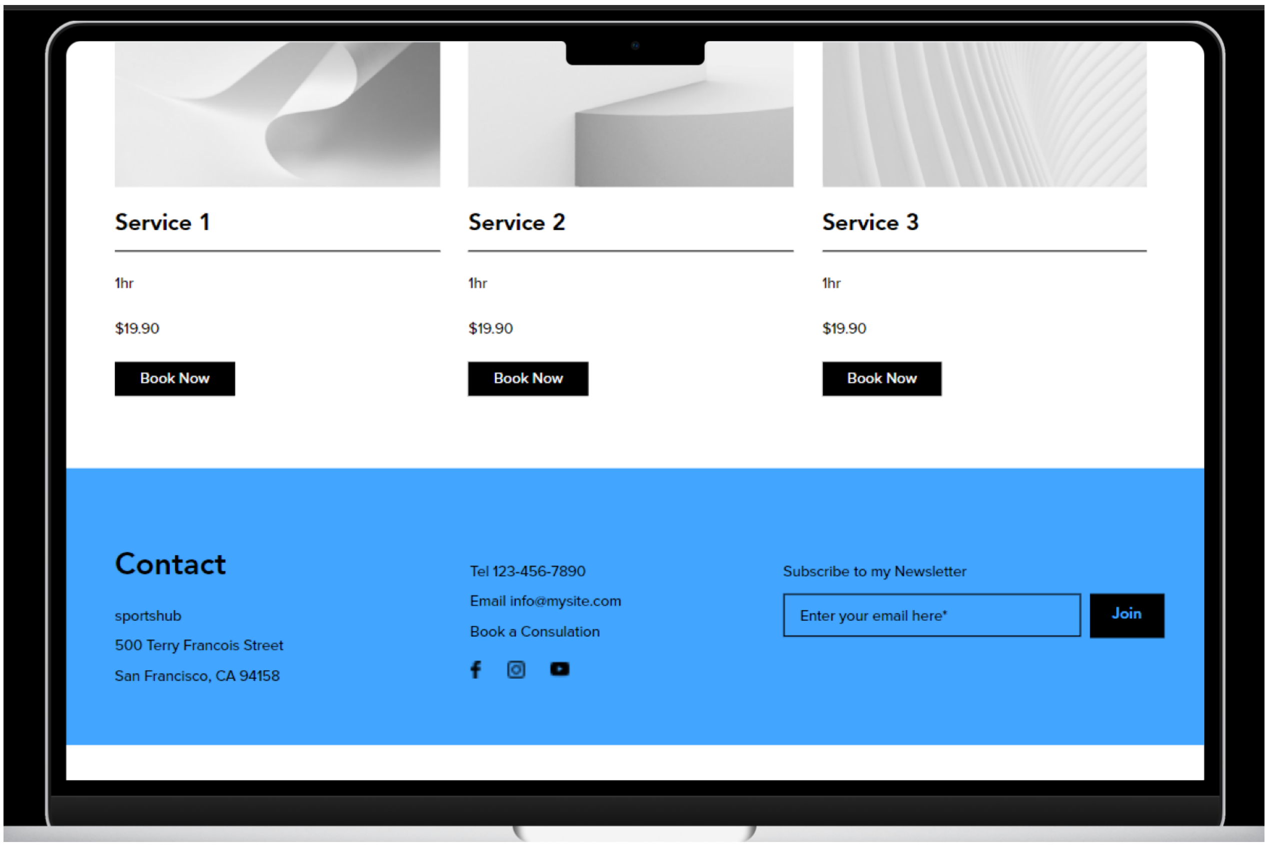


Form Name: Home



About

It all starts with a dream: a dream to be healthy, a dream to be strong, a dream to be fast, and a dream to be #1. At sportshub, we've been making dreams come true since 2000. Our San Francisco facility has been designed to provide an outstanding environment for athletes of all levels and abilities. Whether you're a novice or a professional, reach your potential with sportshub.



Service 1

1hr

$19.90

Book Now

Service 2

1hr

$19.90

Book Now

Service 3

1hr

$19.90

Book Now

Contact

sportshub
500 Terry Francois Street
San Francisco, CA 94158

Tel 123-456-7890
Email info@mysite.com
Book a Consulation

Subscribe to my Newsletter

Enter your email here*

Join

4.4 DATABASE DESIGN

A organized system called a database allows users to retrieve information quickly and easily and has the capacity to retain information. The main objective of any database is its data, which need protection. The database design process is divided into two steps. In order to create a database that as clearly as possible satisfies these objectives, the user requests must first be identified. This stage, called as Information Level Design, is carried out independently from any specific DBMS.
UML is a popular language utilized for modeling systems and objects found in the real world, and one of the tools it employs is use case diagrams to represent their creation and operation. The system architecture is paralleled by a database design.
The database's data layout seeks to achieve the two objectives listed below

- Data independence
- Data Integrity

4.4.1 Relational Database Management System (RDBMS)

RDBMS stands for Relational Database Management System. All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access are based on RDBMS. It is called Relational Database Management System (RDBMS) because it is based on the relational model introduced by E.F. Codd. Data is represented in terms of tuples (rows) in RDBMS.A relational database is the most commonly used database. It contains several tables, and each table has its primary key. Due to a collection of an organized set of tables, data can be accessed easily in RDBMS.

Relations, Domains & Attributes

A table is a data structure representing a relation between entities, where the rows in a table, commonly referred to as tuples, are composed of an ordered set of n elements. These elements are organized into columns, also known as attributes, which define the characteristics of the entities. Relationships between tables in a database are established to ensure both referential and entity relationship integrity. To define the domain of a data attribute, a set of atomic values is specified, typically drawn from a particular data type. Assigning a name to the domain aids in interpreting its values and is a common practice.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.

- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.4.2 Normalization

During the initial phase, data is grouped together in a basic manner to minimize the impact of future changes on data structures. The process of normalizing data structures formally aims to enhance data integrity and reduce duplication by organizing it into logical and efficient groupings. Using the normalization technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. In a table, In a table, each row is distinguished by a unique key, which may either be a primary key or a foreign key. A primary key is a single or multiple elements that act as an exclusive identifier for the particular row in the table, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized. It means placing things in their natural form, as the name suggests. By using normalization, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user. By removing recurring groups from the data, normalization prevents data redundancy, which puts a heavy strain on the computer's resources.

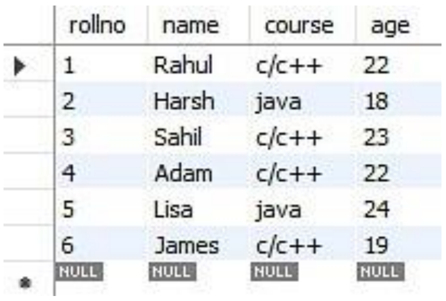
These consist of

- Select appropriate table and column names.
- Normalize the data.
- Choose the proper name for the data.

First Normal Form(1NF)

- A table is considered to be in the First Normal Form if it satisfies the requirement of atomicity, which means that the table's values cannot be divided into smaller, more granular parts, and each attribute or column contains only a single value. In other words, each column should hold only one piece of information, and there should be no repeating groups or arrays of values within a single row.
- The concept of atomicity in this context denotes that a cell within a database cannot contain more than one value, and must exclusively store a single-valued attribute.
- The first normal form in database normalization places limitations on attributes that have multiple values, are composed of multiple sub-attributes, or contain a combination of both. Therefore, in order to satisfy the conditions of the first normal form., these attributes need to be modified or removed, a relation must not have multi-valued or composite attributes, and any such attributes must be split into individual attributes to form atomic values.

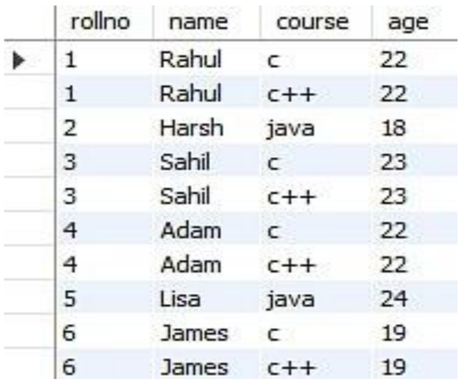
E.g. The table contains information pertaining to students, including their roll number, name, course of study, and age.



rollno	name	course	age
1	Rahul	c/c++	22
2	Harsh	java	18
3	Sahil	c/c++	23
4	Adam	c/c++	22
5	Lisa	java	24
6	James	c/c++	19
NULL	NULL	NULL	NULL

Figure 12 : Table not in 1NF

The table containing the records of students displays a violation of the First Normal Form due to the presence of two distinct values in the course column. To ensure compliance with the First Normal Form, the table has been modified resulting in the formation of a new table.



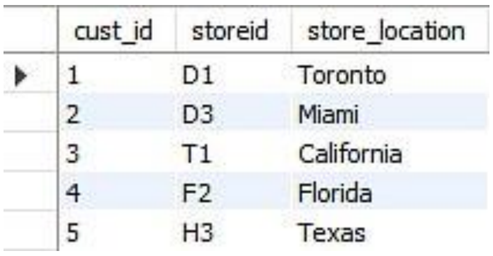
rollno	name	course	age
1	Rahul	c	22
1	Rahul	c++	22
2	Harsh	java	18
3	Sahil	c	23
3	Sahil	c++	23
4	Adam	c	22
4	Adam	c++	22
5	Lisa	java	24
6	James	c	19
6	James	c++	19

Figure 13 : Table in 1NF

The First Normal Form is applied to achieve atomicity in the system, which ensures that each column has unique values. By following this normalization process, the data is organized into individual atomic values, eliminating any redundancies or repeating groups. As a result, data integrity is maintained, and the system can efficiently handle complex data manipulations and updates.

Second Normal Form(2NF)

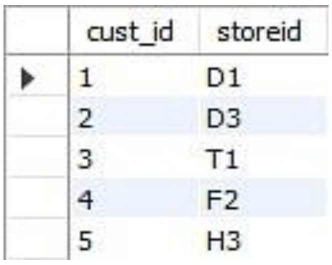
To meet requirements of Second Normal Form, a table must satisfy the criteria of First Normal Form as a prerequisite. Furthermore, the table must not exhibit partial dependency, which refers to a scenario where a non-prime attribute is dependent on a proper subset of the candidate key. In other words, the table should have no attributes that are determined by only a portion of the primary key.
Now understand the Second Normal Form with the help of an example.
Consider the table Location:



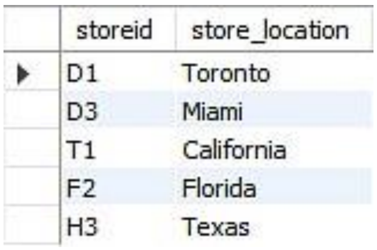
	cust_id	storeid	store_location
▶	1	D1	Toronto
	2	D3	Miami
	3	T1	California
	4	F2	Florida
	5	H3	Texas

Figure 14 : Table not in 2NF

The Location table currently has a composite primary key consisting of cust id and storied, and its non-key attribute is store location. However, the store location attribute is directly determined by the storied attribute, which is part of the primary key. As a result, this table does not meet the requirements of second normal form. To address this issue and ensure second normal form is met, it is necessary to separate the Location table into two separate tables. This will result in the creation of two distinct tables that accurately represent the relevant data and relationships: one for customer IDs and store IDs, and another for store IDs and their respective locations.:



	cust_id	storeid
▶	1	D1
	2	D3
	3	T1
	4	F2
	5	H3



	storeid	store_location
▶	D1	Toronto
	D3	Miami
	T1	California
	F2	Florida
	H3	Texas

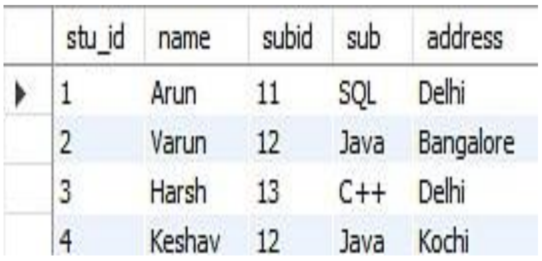
Figure 15 : Table in 2NF

After eliminating the partial functional dependency in the location table, it can be observed that the column storing the location is now entirely dependent on the primary key of the same table. In other words, the location column is fully determined by the primary key, thereby ensuring greater data consistency and integrity in the table.

Third Normal Form

A table must meet the requirements of Second Normal Form in order to be considered in Third Normal Form, and also fulfill two additional conditions. The second condition states that non- prime attributes should not rely on non-prime characteristics that are not a part of the candidate key within the same table, thus avoiding transitive dependencies. A transitive dependency arises when A → C indirectly, due to A → B and B → C, where B is not functionally dependent on A. The main objective of achieving Third Normal Form is to reduce data redundancy and guarantee data integrity.

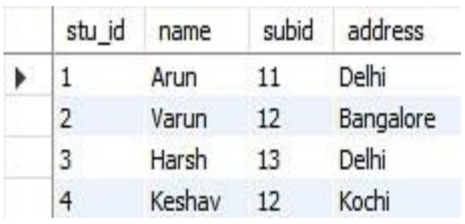
For instance, let's consider a student table that includes columns like student ID, student name, subject ID, subject name, and address of the student. To comply with the requirements for Third Normal Form, this table must first meet the standards of Second Normal Form and then ensure that there are no transitive dependencies between non-prime attributes.



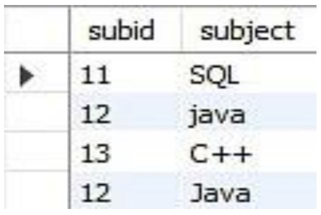
stu_id	name	subid	sub	address
1	Arun	11	SQL	Delhi
2	Varun	12	Java	Bangalore
3	Harsh	13	C++	Delhi
4	Keshav	12	Java	Kochi

Figure 16 : Table not in 3NF

Now to change the table to the third normal form, you need to divide the table as shown below: Based on the given student table, it can be observed that the stu_id attribute determines the sub_id attribute, and the sub_id attribute determines the subject (sub). This implies that there is a transitive functional dependency between stu_id and sub. As a result, the table does not satisfy the criteria for the third normal form. To adhere to the third normal form, the table must be divided into separate tables where each table represents a unique entity or relationship. In this case, the table can be divided into two tables: one for the student-subject relationship (stu_id and sub_id), and another for the subject information (sub_id and sub):



stu_id	name	subid	address
1	Arun	11	Delhi
2	Varun	12	Bangalore
3	Harsh	13	Delhi
4	Keshav	12	Kochi



subid	subject
11	SQL
12	java
13	C++
12	Java

Figure 17 : Table in 3NF

The two tables illustrate that the non-key attributes are completely determined by and reliant on the primary key. In the first table, the columns for name, sub_id, and addresses are all exclusively linked to the stu_id. Likewise, the second table demonstrates that the sub column is entirely dependent on the sub_id.

4.4.3 Sanitization

Data Sanitization involves the secure and permanent erasure of sensitive data from datasets and media to guarantee that no residual data can be recovered even through extensive forensic analysis. Data sanitization has a wide range of applications but is mainly used for clearing out end-of-life electronic devices or for the sharing and use of large datasets that contain sensitive information. The main strategies for erasing personal data from devices are physical destruction, cryptographic erasure, and data erasure. While the term data sanitization may lead some to believe that it only includes data on electronic media, the term also broadly covers physical media, such as paper copies. These data types are termed soft for electronic files and hard for physical media paper copies. Data sanitization methods are also applied for the cleaning of sensitive data, such as through heuristic-based methods, machine-learning based methods, and k-source anonymity.

4.4.4 Indexing

Indexing is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. The index is a type of data structure. It is used to locate and access the data in a database table quickly.

- Primary Index − Primary index is defined on an ordered data file. The data file is ordered on a key field. The key field is generally the primary key of the relation.
- Secondary Index − Secondary index may be generated from a field which is a candidate key and has a unique value in every record, or a non-key with duplicate values.

Clustering Index − Clustering index is defined on an ordered data file. The data file is ordered on a non-key field.

4.5 TABLE DESIGN

1.tbl_login

Primary key: **login_id**

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	username	varchar(100)	Unique	Username
3	email	varchar(254)	Unique	Email of the user
4	first_name	varchar(100)	Not Null	First name
5	last_name	varchar(100)	Not Null	Last name
6	role	varchar(20)	Not Null	User role (e.g., "FitnessUser", "FitnessTrainer", "SportsTrainer")
7	description	text	Blank	Description (optional)

8	profile_picture	varchar(100)	Blank	Profile picture filename (optional)
9	address	text	Blank	Address (optional)
10	contact_number	varchar(15)	Blank	Contact number (optional)
11	second_contact_number	varchar(15)	Blank	Second contact number (optional)
12	is_active	boolean	Default=True	User is active or not (default is active)
13	is_staff	boolean	Default=False	User is a staff member or not (default is not staff)
14	last_login	datetime	Blank	Last login timestamp
15	date_joined	datetime	Not Null	Date joined (registration date)

2. tbl_role

Primary Key: id

No:	**Fieldname**	**Datatype**	**Key Constraints**	**Description**
1	id	int(11)	Primary Key	Primary Key
2	name	varchar(20)	Unique	Role name

3. tbl_commonchoice

Primary Key: id

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	name	varchar(255)		Display name of the choice
3	choice_type	varchar(25)		Type of choice (e.g., "Fitness Goal", "Trainer Specialization")

4. tbl_roleapplication

Primary Key: id
Foreign Key: user_id references table tbl_customuser
Foreign Key: role_id references table tbl_role
Foreign Key: fitness_goal_id references table tbl_commonchoice

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	role_id	int(11)	Foreign Key	References table tbl_role
4	specialization_details	text	Blank	Specialization details (optional)
5	is_approved	boolean	Default=False	Application approval status
6	fitness_goal_id	int(11)	Foreign Key	References table tbl_commonchoice
7	height	float	Blank	Height (optional)
8	weight	float	Blank	Weight (optional)
9	experience	int(11)	Blank	Experience (optional)
10	certification	varchar(255)		Certification
11	certification_link	varchar(255)	Blank	Certification link (optional)

5. tbl_sportscenter

Primary Key: id

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	name	varchar(100)	Not Null	Sports Center name
3	location	varchar(200)	Not Null	Location of the sports center
4	description	text	Blank	Description of the sports center (optional)
5	image	varchar(100)	Blank, Null	Image filename of the sports center (optional)
6	price_per_slot	decimal(10, 2)	Blank, Null	Price per slot (optional)

6. tbl_sportscenterslot

Primary Key: id

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	start_time	time	Not Null	Start time
3	end_time	time	Not Null	End time

7. tbl_reservation

Primary Key: id
Foreign Key: reserver_id references table tbl_customuser
Foreign Key: sport_id references table tbl_sportscenter
Foreign Key: slot_id references table tbl_sportscenterslot

No:	Fieldname	Datatype	Key Constraints	Description

1	id	int(11)	Primary Key	Primary Key
2	reserver_id	int(11)	Foreign Key	References table tbl_customuser
3	sport_id	int(11)	Foreign Key	References table tbl_sportscenter
4	slot_id	int(11)	Foreign Key	References table tbl_sportscenterslot
5	reservation_date	date	Not Null	Reservation date
6	reservation_time	datetime	Auto-Generated	Reservation timestamp
7	status	boolean	Default=False	Reservation status

8. tbl_payment

Primary Key: id
Foreign Key: user_id references table tbl_customuser
Foreign Key: reservation_id references table tbl_reservation
Foreign Key: trainingplanassignment_id references table tbl_trainingplanassignment

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	razorpay_order_id	varchar(255)	Not Null	Razorpay order ID
4	payment_id	varchar(255)	Not Null	Razorpay payment ID
5	amount	decimal(8, 2)	Not Null	Amount paid
6	currency	varchar(3)	Not Null	Currency code (e.g., "INR")
7	timestamp	datetime	Auto-Generated	Timestamp of the payment
8	payment_status	varchar(20)	Default=Pending	Payment status
9	reservation_id	int(11)	Foreign Key	References table tbl_reservation (optional)
10	trainingplanassignment_id	int(11)	Foreign Key	References table tbl_trainingplanassignment (optional)

9. tbl_roleapplication

Primary Key: id
Foreign Key: user_id references table tbl_customuser
Foreign Key: role_id references table tbl_role
Foreign Key: fitness_goal_id references table tbl_commonchoice

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	role_id	int(11)	Foreign Key	References table tbl_role
4	specialization_details	text	Blank	Specialization details (optional)
5	is_approved	boolean	Default=False	Application approval status
6	fitness_goal_id	int(11)	Foreign Key	References table tbl_commonchoice
7	height	float	Blank	Height (optional)
8	weight	float	Blank	Weight (optional)
9	experience	int(11)	Blank	Experience (optional)
10	certification	varchar(255)		Certification
11	certification_link	varchar(255)	Blank	Certification link (optional)

10. tbl_usertrainerconnection

Primary Key: id
Foreign Key: user_id references table tbl_customuser
Foreign Key: trainer_id references table tbl_fitnesstrainer

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	trainer_id	int(11)	Foreign Key	References table tbl_fitnesstrainer
4	start_date	date	Not Null	Start date of the connection

5	end_date	date	Not Null	End date of the connection
6	(Other fields)	(Datatype)	(Constraints)	(Description)

11. tbl_trainingplan

Primary Key: id
Foreign Key: connection_id references table tbl_usertrainerconnection

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	connection_id	int(11)	Foreign Key	References table tbl_usertrainerconnection
3	name	varchar(255)	Not Null	Training plan name
4	description	text		Description of the plan
5	hours_per_day	int(11)		Hours per day for training
6	days_per_week	int(11)		Days per week for training
7	pace	varchar(255)		Pace of training plan

12. tbl_fitnessuser

Primary Key: id
Foreign Key: user_id references table tbl_customuser

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	fitness_goal	varchar(255)	Not Null	Fitness goal of the user
4	height	float	Not Null	Height of the user

5	weight	float	Not Null	Weight of the user

13. tbl_fitnessuser

Primary Key: id
Foreign Key: user_id references table tbl_customuser

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	fitness_goal	varchar(255)	Not Null	Fitness goal of the user
4	height	float	Not Null	Height of the user
5	weight	float	Not Null	Weight of the user

14. tbl_fitnessuser

Primary Key: id
Foreign Key: user_id references table tbl_customuser

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	fitness_goal	varchar(255)	Not Null	Fitness goal of the user
4	height	float	Not Null	Height of the user
5	weight	float	Not Null	Weight of the user

15. tbl_sportstrainer

Primary Key: id
Foreign Key: user_id references table tbl_customuser

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	specialization	varchar(255)	Not Null	Specialization of the trainer
4	height	float	Blank, Null	Height of the trainer (optional)
5	weight	float	Blank, Null	Weight of the trainer (optional)
6	(Other fields)	(Datatype)	(Constraints)	(Description)

16. tbl_fitnesstrainer

Primary Key: id
Foreign Key: user_id references table tbl_customuser

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	experience	int(11)	Not Null	Experience of the trainer
4	certification	varchar(255)		Trainer's certification
5	training_goal	varchar(255)	Not Null	Training goal of the trainer
6	certification_link	varchar(255)	Blank, Null	Certification link (optional)
7	height	float	Blank, Null	Height of the trainer (optional)
8	weight	float	Blank, Null	Weight of the trainer (optional)

17. tbl_traineruserconnection

Primary Key: id
Foreign Key: fitness_trainer_id references table tbl_fitnesstrainer
Foreign Key: fitness_user_id references table tbl_fitnessuser

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	fitness_trainer_id	int(11)	Foreign Key	References table tbl_fitnesstrainer
3	fitness_user_id	int(11)	Foreign Key	References table tbl_fitnessuser
4	status	varchar(10)	Default=pending	Connection status (pending, approved, rejected)
5	certification_link	varchar(255)	Blank, Null	Certification link (optional)

18. tbl_fitnesstrainer

Primary Key: id
Foreign Key: user_id references table tbl_customuser

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	user_id	int(11)	Foreign Key	References table tbl_customuser
3	experience	int(11)	Not Null	Experience of the trainer
4	certification	varchar(255)		Trainer's certification
5	training_goal	varchar(255)	Not Null	Training goal of the trainer
6	certification_link	varchar(255)	Blank, Null	Certification link (optional)
7	height	float	Blank, Null	Height of the trainer (optional)
8	weight	float	Blank, Null	Weight of the trainer (optional)

19 . tbl_trainingplan

Primary Key: id
Foreign Key: created_by_trainer_id references table tbl_fitnesstrainer

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	plan_name	varchar(255)	Not Null	Training plan name
3	description	text		Description of the plan
4	duration	int(11)		Duration of the plan (in days)
5	created_by_trainer_id	int(11)	Foreign Key	References table tbl_fitnesstrainer
6	amount	decimal(10, 2)	Blank, Null	Amount (optional)

20. tbl_equipment

Primary Key: id

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	name	varchar(255)	Not Null	Equipment name
3	description	text		Equipment description

21. tbl_timeslot

Primary Key: id

No:	Fieldname	Datatype	Key Constraints	Description
1	id	int(11)	Primary Key	Primary Key
2	slot_number	int(11)	Unique	Slot number (unique)
3	start_time	time	Not Null	Start time of the slot
4	end_time	time	Not Null	End time of the slot

22. tbl_equipmentreservation

Primary Key: id
Foreign Key: trainer_id references table tbl_fitnesstrainer
Foreign Key: fitness_user_id references table tbl_fitnessuser
Foreign Key: equipment_id references table tbl_equipment
Foreign Key: timeslot_id references table tbl_timeslot

No:	**Fieldname**	**Datatype**	**Key Constraints**	**Description**
1	id	int(11)	Primary Key	Primary Key
2	trainer_id	int(11)	Foreign Key	References table tbl_fitnesstrainer
3	fitness_user_id	int(11)	Foreign Key	References table tbl_fitnessuser
4	equipment_id	int(11)	Foreign Key	References table tbl_equipment
5	timeslot_id	int(11)	Foreign Key	References table tbl_timeslot
6	date	date	Default=Today	Reservation date

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is a thorough method of analysing a piece of software's performance to see if it performs as planned. This procedure, which is also known as verification and validation, entails evaluating a product to make sure it complies with relevant standards and meets user needs. Software testing is accomplished using a variety of techniques, including reviews, analyses, inspections, and walkthroughs, as well as associated processes like static analysis. Static analysis, which evaluates source code without running it, and dynamic analysis, which keeps track of programme behaviour while it is in use and produces information like execution traces, timing profiles, and test coverage statistics. The various tasks that make up testing can be pre-organized and completed in a methodical manner. The testing process for computer-based systems begins with separate modules and moves forward through system integration. There are many laws that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful.

5.2 TEST PLAN

A test plan defines the steps necessary to carry out different testing approaches, as well as the tasks that must be accomplished. The task of generating computer programmes, documentation, and data structures falls under the purview of software developers. Individual testing is required to make sure that every part of the programme operates as planned. An independent test group (ITG) is ready to give developers feedback and spot any potential problems. The test strategy must include a quantification of the testing objectives. These goals may include measurements like the mean time to failure, the cost of fixing errors, the density of remaining errors, the frequency of recurrence, and the number of hours needed for regression testing. The testing levels could include:

- Unit testing
- Integration Testing
- Validation Testing or System Testing
- Output Testing or User Acceptance Testing
- Automation Testing
- Selenium Testing

5.2.1 Unit Testing

Unit testing is a crucial stage of software verification that concentrates on testing individual components or modules, which are the fundamental building blocks of software design. This testing process entails reviewing the component-level design specifications to detect any errors within the module's boundaries by analyzing key control paths. The complexity of the test and the untested areas are determined during the process of unit testing, which is designed to be white-box focused, and multiple components can be tested simultaneously. To ensure proper functionality of the program unit being tested, the modular interface undergoes checks to ensure correct data flow in and out. Additionally, the integrity of temporarily stored data in the local data structure is examined throughout the algorithm's execution. The evaluation of boundary conditions confirms that every statement within the module has been executed at least once. Finally, inspection of each error management path is conducted to ensure proper error handling.

Performing data flow tests across module interfaces prior to any other testing is crucial for ensuring the effectiveness of the testing process. This is because if data cannot flow in and out of the system correctly, all other tests will be useless. Error handling channels must be established, and fault scenarios must be anticipated during the design stage to ensure that the system can redirect or stop working when an error occurs. The final stage of unit testing is boundary testing, which involves testing the software at its boundaries. This is because software often fails at its boundaries.

5.2.2 Integration Testing

Integration testing is a rigorous procedure that entails creating the structure of a program and executing tests to identify any problems related to the connections between different components. The objective is to establish a program structure based on design, utilizing components that have undergone unit testing. The overall program is then tested as a whole. However, correcting any issues can be challenging since the program's size makes it difficult to isolate the root causes. Once these errors are corrected, new ones may emerge, and the process may seem to repeat itself endlessly. After completing unit testing on all modules, they are integrated into the system to verify that there are no interface inconsistencies. This integration process also leads to the development of a distinctive program structure, as any discrepancies in the program structures are eliminated.

5.2.3 Validation Testing or System Testing

After conducting the testing phase, the entire system, including its code, modules, and class modules, was thoroughly examined using a process known as system tests or black box testing. Black box testing specifically aims to ensure that the software's functional requirements are met by creating input conditions that replicate all possible scenarios. The objective of this testing procedure is to detect and address a range of issues spanning from inadequate or inaccurate functionalities, interface discrepancies, errors in data arrangement or external data retrieval, performance drawbacks, initialization malfunctions, to termination glitches.

5.2.4 Output Testing or User Acceptance Testing

The system under consideration is tested for user acceptance; in this case, it must satisfy the business' requirements. The programme should consult the user and the perspective system while it is being developed in order to make any necessary adjustments. This was accomplished in regards to the following areas : • Input Screen Designs

• Output Screen Designs

A variety of test data are used to conduct the aforementioned tests. The process of system testing requires the preparation of test data. After the preparation of sample data, the system being analyzed is put to the test using that data. Test data errors are discovered once more and resolved using the testing techniques mentioned above when the system is tested. Additionally, the fixes are noted for future use.

5.2.5 Automation Testing

Automation testing is a software testing process that verifies whether a software product meets specific requirements by automatically executing tests. The main purpose of automation testing is to detect bugs, defects, and other issues in software products and ensure that they perform as designed.

Benefits of using automation testing for software development:

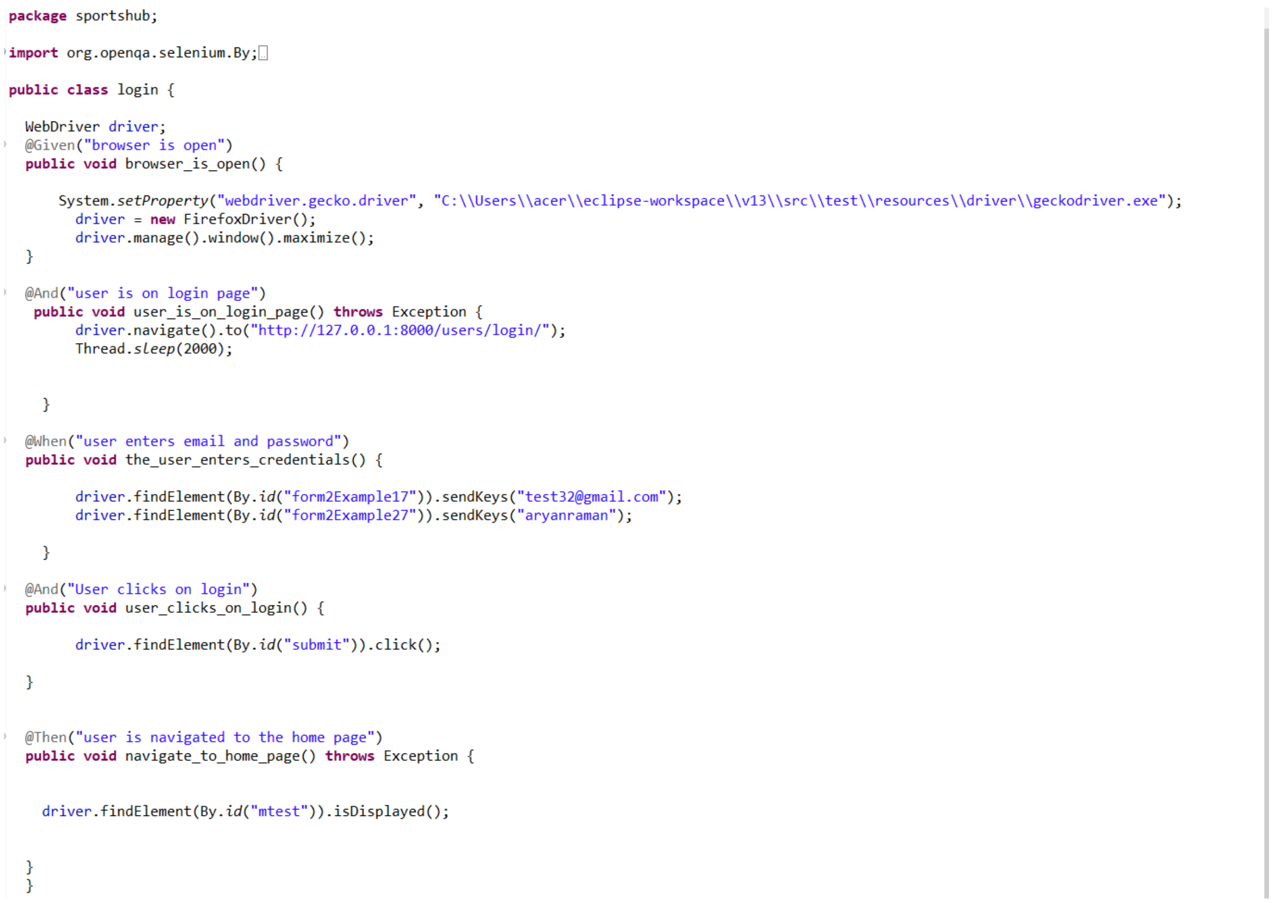
- Detailed reporting capabilities
- Improved bug detection
- Simplifies testing
- Speeds up the testing process
- Reduces human intervention

5.2.6 Selenium Testing

Selenium is a widely used open-source automation testing suite for web UI. Originally developed by Jason Huggins in 2004, Selenium supports automation across different browsers, platforms, and programming languages. It can be deployed on Windows, Linux, Solaris, Macintosh, and even supports mobile OS such as iOS, Windows Mobile, and Android. Selenium supports various programming languages through drivers specific to each language including C#, Java, Perl, PHP, Python, and Ruby. Selenium Web Driver is the most popular with Java and C#. Test scripts can be coded in any supported language and run directly in modern web browsers such as Internet Explorer, Mozilla Firefox, Google Chrome, and Safari. Selenium is not only used for functional tests but can also be integrated with automation test tools like Maven, Jenkins, and Docker for continuous testing. Furthermore, it can be integrated with TestNG and JUnit for managing test cases and generating reports.

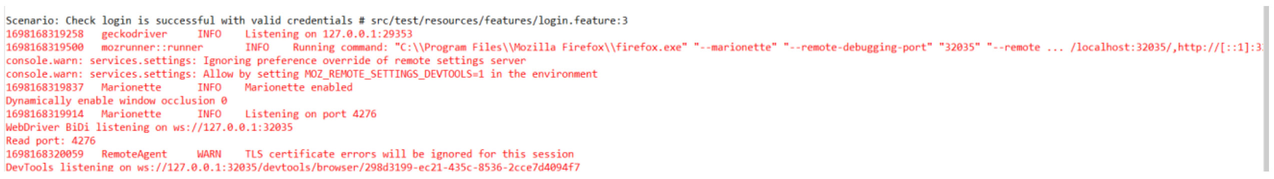
Test Case 1

Code



```java
package sportshub;

import org.openqa.selenium.By;

public class login {

  WebDriver driver;
  @Given("browser is open")
  public void browser_is_open() {

      System.setProperty("webdriver.gecko.driver", "C:\\Users\\acer\\eclipse-workspace\\v13\\src\\test\\resources\\driver\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
  }

  @And("user is on login page")
   public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://127.0.0.1:8000/users/login/");
        Thread.sleep(2000);


    }

  @When("user enters email and password")
  public void the_user_enters_credentials() {

        driver.findElement(By.id("form2Example17")).sendKeys("test32@gmail.com");
        driver.findElement(By.id("form2Example27")).sendKeys("aryanraman");

    }

  @And("User clicks on login")
  public void user_clicks_on_login() {

        driver.findElement(By.id("submit")).click();

  }


  @Then("user is navigated to the home page")
  public void navigate_to_home_page() throws Exception {


    driver.findElement(By.id("mtest")).isDisplayed();


  }
  }
```

Eg.Screenshot

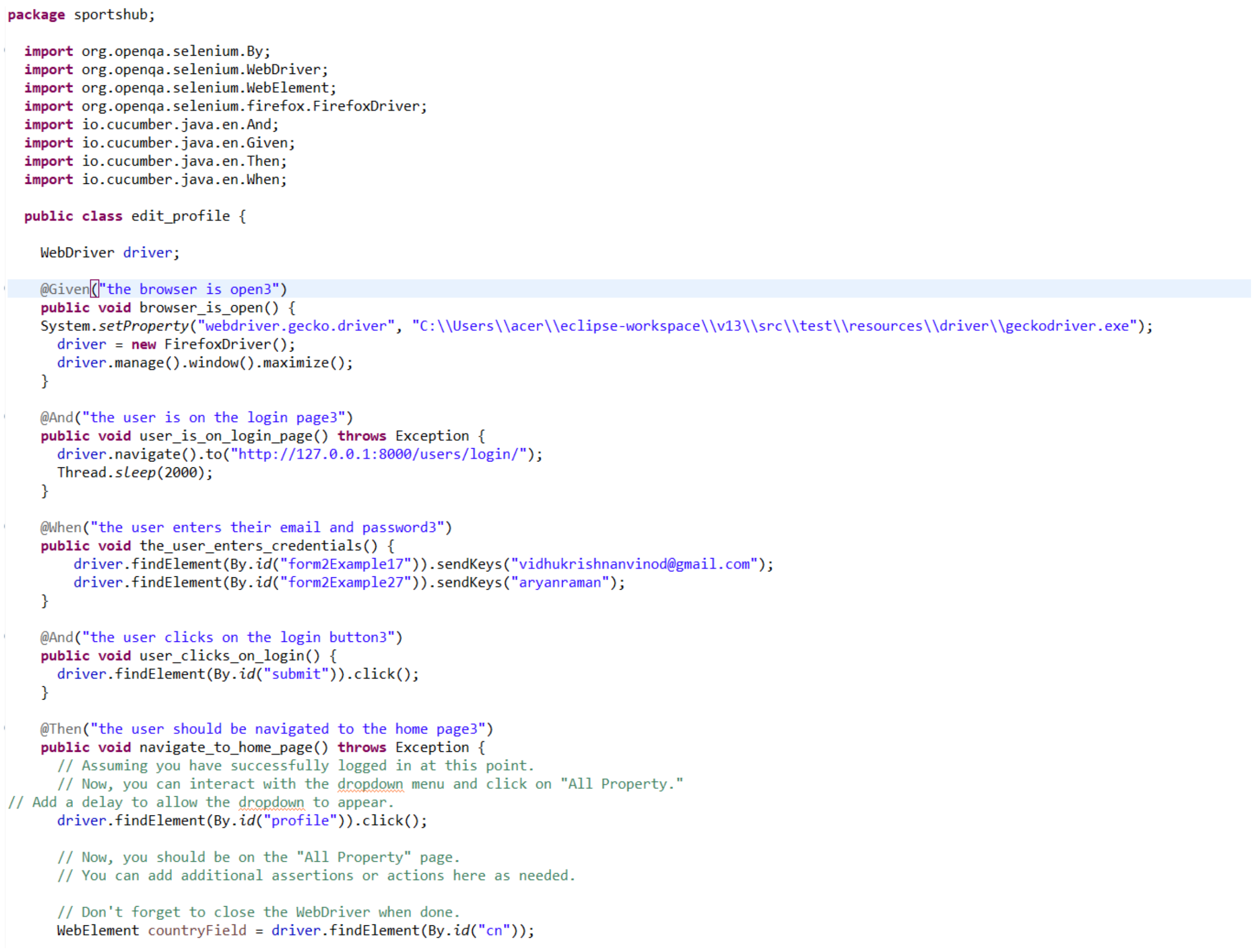


```
Scenario: Check login is successful with valid credentials # src/test/resources/features/login.feature:3
1698168319258   geckodriver     INFO    Listening on 127.0.0.1:29353
1698168319500   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "--marionette" "--remote-debugging-port" "32035" "--remote ... /localhost:32035/,http://[::1]:3
console.warn: services.settings: Ignoring preference override of remote settings server
console.warn: services.settings: Allow by setting MOZ_REMOTE_SETTINGS_DEVTOOLS=1 in the environment
1698168319837   Marionette      INFO    Marionette enabled
Dynamically enable window occlusion 0
1698168319914   Marionette      INFO    Listening on port 4276
WebDriver BiDi listening on ws://127.0.0.1:32035
Read port: 4276
1698168320059   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:32035/devtools/browser/298d3199-ec21-435c-8536-2cce7d4094f7
```

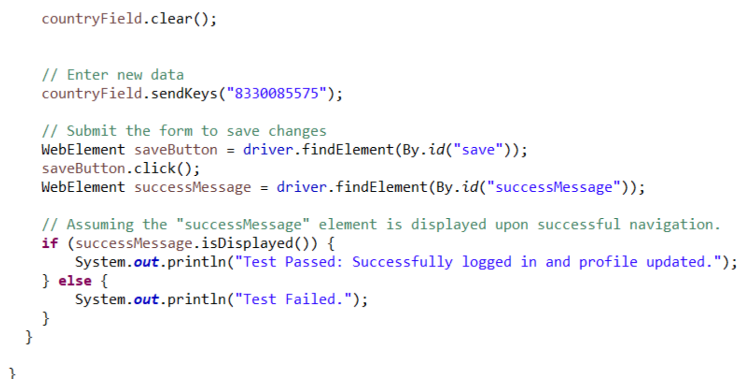
Test Report

Project Name : Sportshub					
Login Test Case					
Test Case ID: 1			**Test Designed By: Vidhu krishnan vinod**		
Test Priority (Low/Medium/High): High			**Test Designed Date: 23**-10-2023		
Module Name: Login Page			**Test Executed By: Dr.Paulin Paul**		
Test Title: Verify login with valid email and password			**Test Execution Date: 23**-10-2023		
Description: Test the Login Page					
Pre-Condition: User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid email	User Name: test32@gmail.com"	User should be able to Login	User Logged in and navigated to the dashboard with records	Pass
3	Provide Valid Password	Password: aryanraman			
4	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 2:

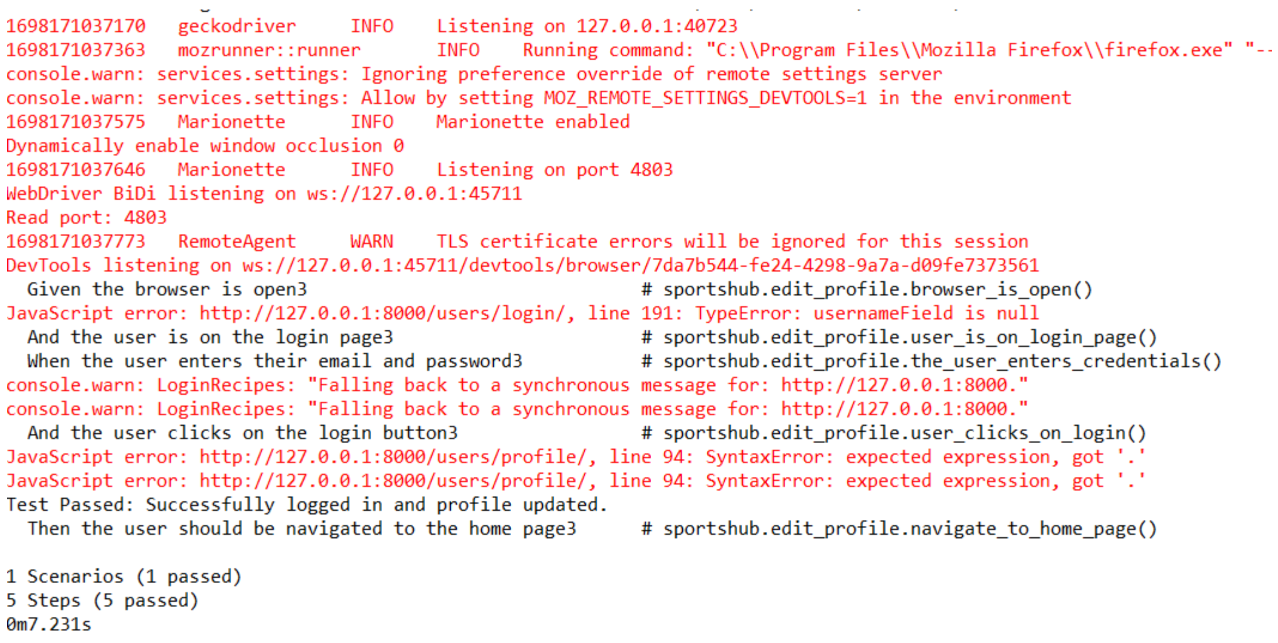


```
package sportshub;

 import org.openqa.selenium.By;
 import org.openqa.selenium.WebDriver;
 import org.openqa.selenium.WebElement;
 import org.openqa.selenium.firefox.FirefoxDriver;
 import io.cucumber.java.en.And;
 import io.cucumber.java.en.Given;
 import io.cucumber.java.en.Then;
 import io.cucumber.java.en.When;

 public class edit_profile {

   WebDriver driver;

   @Given("the browser is open3")
   public void browser_is_open() {
   System.setProperty("webdriver.gecko.driver", "C:\\Users\\acer\\eclipse-workspace\\v13\\src\\test\\resources\\driver\\geckodriver.exe");
     driver = new FirefoxDriver();
     driver.manage().window().maximize();
   }

   @And("the user is on the login page3")
   public void user_is_on_login_page() throws Exception {
     driver.navigate().to("http://127.0.0.1:8000/users/login/");
     Thread.sleep(2000);
   }

   @When("the user enters their email and password3")
   public void the_user_enters_credentials() {
       driver.findElement(By.id("form2Example17")).sendKeys("vidhukrishnanvinod@gmail.com");
       driver.findElement(By.id("form2Example27")).sendKeys("aryanraman");
   }

   @And("the user clicks on the login button3")
   public void user_clicks_on_login() {
     driver.findElement(By.id("submit")).click();
   }

   @Then("the user should be navigated to the home page3")
   public void navigate_to_home_page() throws Exception {
     // Assuming you have successfully logged in at this point.
     // Now, you can interact with the dropdown menu and click on "All Property."
// Add a delay to allow the dropdown to appear.
     driver.findElement(By.id("profile")).click();

     // Now, you should be on the "All Property" page.
     // You can add additional assertions or actions here as needed.

     // Don't forget to close the WebDriver when done.
     WebElement countryField = driver.findElement(By.id("cn"));
```



```
     countryField.clear();


     // Enter new data
     countryField.sendKeys("8330085575");

     // Submit the form to save changes
     WebElement saveButton = driver.findElement(By.id("save"));
     saveButton.click();
     WebElement successMessage = driver.findElement(By.id("successMessage"));

     // Assuming the "successMessage" element is displayed upon successful navigation.
     if (successMessage.isDisplayed()) {
         System.out.println("Test Passed: Successfully logged in and profile updated.");
     } else {
         System.out.println("Test Failed.");
     }
   }

 }
```

Screenshot



```
1698171037170   geckodriver     INFO    Listening on 127.0.0.1:40723
1698171037363   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "--
console.warn: services.settings: Ignoring preference override of remote settings server
console.warn: services.settings: Allow by setting MOZ_REMOTE_SETTINGS_DEVTOOLS=1 in the environment
1698171037575   Marionette      INFO    Marionette enabled
Dynamically enable window occlusion 0
1698171037646   Marionette      INFO    Listening on port 4803
WebDriver BiDi listening on ws://127.0.0.1:45711
Read port: 4803
1698171037773   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:45711/devtools/browser/7da7b544-fe24-4298-9a7a-d09fe7373561
  Given the browser is open3                           # sportshub.edit_profile.browser_is_open()
JavaScript error: http://127.0.0.1:8000/users/login/, line 191: TypeError: usernameField is null
  And the user is on the login page3                   # sportshub.edit_profile.user_is_on_login_page()
  When the user enters their email and password3       # sportshub.edit_profile.the_user_enters_credentials()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
  And the user clicks on the login button3             # sportshub.edit_profile.user_clicks_on_login()
JavaScript error: http://127.0.0.1:8000/users/profile/, line 94: SyntaxError: expected expression, got '.'
JavaScript error: http://127.0.0.1:8000/users/profile/, line 94: SyntaxError: expected expression, got '.'
Test Passed: Successfully logged in and profile updated.
  Then the user should be navigated to the home page3      # sportshub.edit_profile.navigate_to_home_page()

1 Scenarios (1 passed)
5 Steps (5 passed)
0m7.231s
```

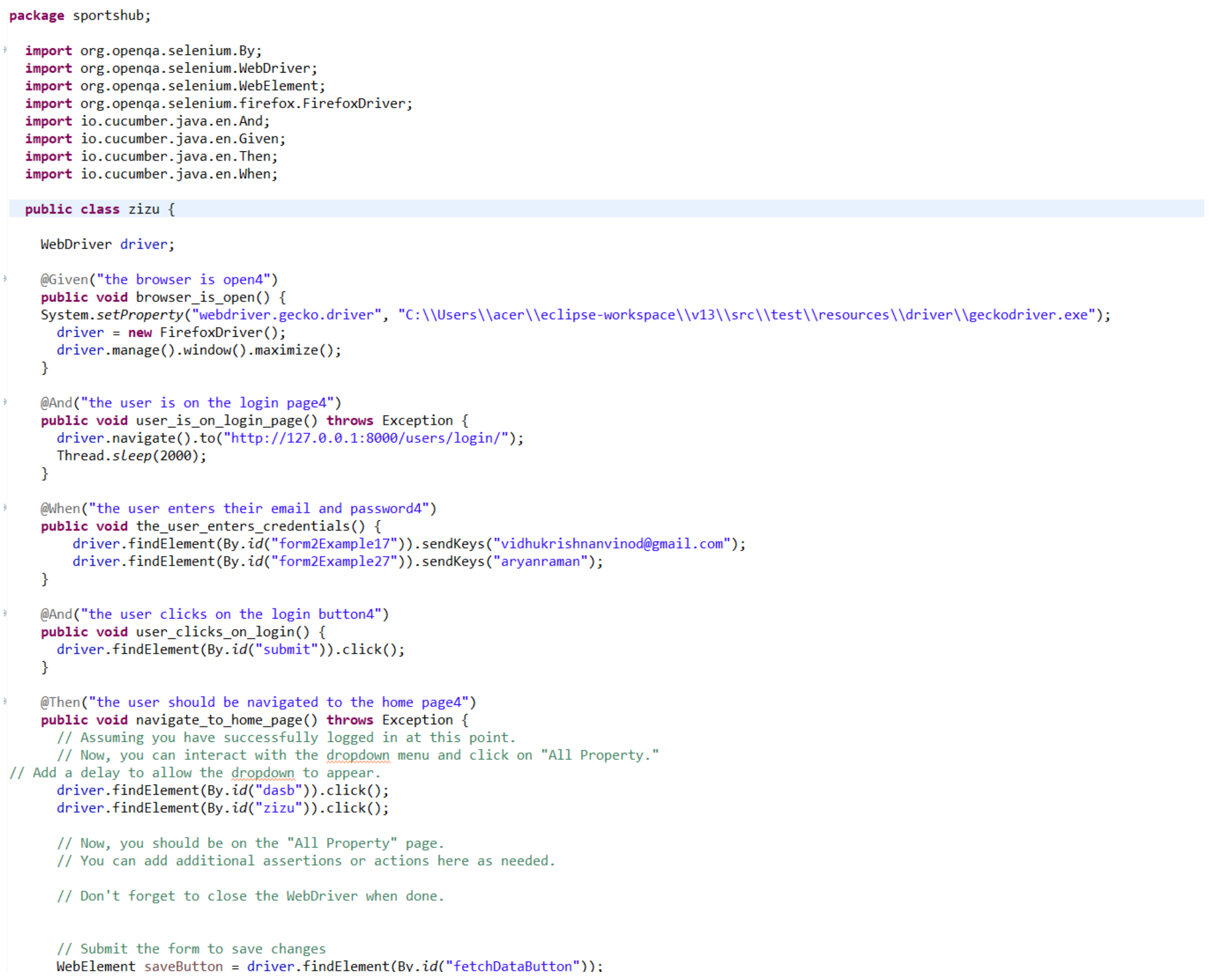
Test report

Project Name : SoulCure					
Login Test Case					
Test Case ID: 2			**Test Designed By: Vidhu krishnan vinod**		
Test Priority (Low/Medium/High): High			**Test Designed Date: 23**-10-2023		
Module Name: Login Page			**Test Executed By: Dr.Paulin Paul**		
Test Title: Edit profile			**Test Execution Date: 23**-10-2023		
Description: change 2 nd contact number					
Pre-Condition: User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to home page		Directed to homepage	home page displayed	Pass
2	Select profile section		User should be able edit	User Logged in and navigated to the	Pass

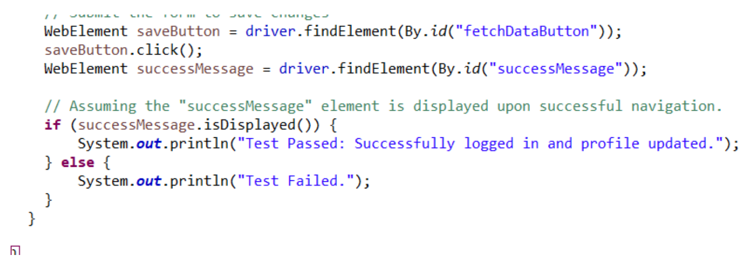
3	Provide Valid Password	Number: “8330085575”	2 nd contact number	dashboard with records	
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 3:

Code

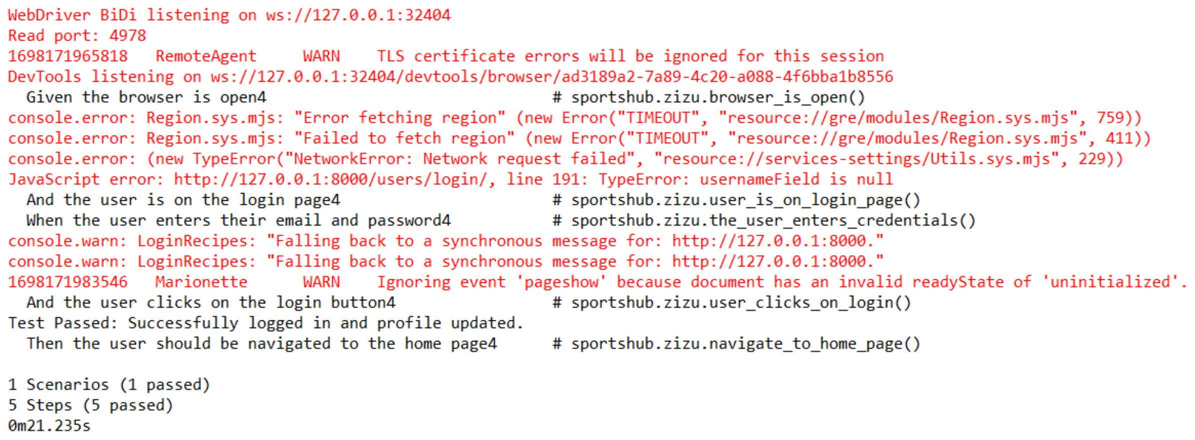


```
package sportshub;

  import org.openqa.selenium.By;
  import org.openqa.selenium.WebDriver;
  import org.openqa.selenium.WebElement;
  import org.openqa.selenium.firefox.FirefoxDriver;
  import io.cucumber.java.en.And;
  import io.cucumber.java.en.Given;
  import io.cucumber.java.en.Then;
  import io.cucumber.java.en.When;

  public class zizu {

    WebDriver driver;

    @Given("the browser is open4")
    public void browser_is_open() {
    System.setProperty("webdriver.gecko.driver", "C:\\Users\\acer\\eclipse-workspace\\v13\\src\\test\\resources\\driver\\geckodriver.exe");
      driver = new FirefoxDriver();
      driver.manage().window().maximize();
    }

    @And("the user is on the login page4")
    public void user_is_on_login_page() throws Exception {
      driver.navigate().to("http://127.0.0.1:8000/users/login/");
      Thread.sleep(2000);
    }

    @When("the user enters their email and password4")
    public void the_user_enters_credentials() {
        driver.findElement(By.id("form2Example17")).sendKeys("vidhukrishnanvinod@gmail.com");
        driver.findElement(By.id("form2Example27")).sendKeys("aryanraman");
    }

    @And("the user clicks on the login button4")
    public void user_clicks_on_login() {
      driver.findElement(By.id("submit")).click();
    }

    @Then("the user should be navigated to the home page4")
    public void navigate_to_home_page() throws Exception {
      // Assuming you have successfully logged in at this point.
      // Now, you can interact with the dropdown menu and click on "All Property."
// Add a delay to allow the dropdown to appear.
      driver.findElement(By.id("dasb")).click();
      driver.findElement(By.id("zizu")).click();

      // Now, you should be on the "All Property" page.
      // You can add additional assertions or actions here as needed.

      // Don't forget to close the WebDriver when done.


      // Submit the form to save changes
      WebElement saveButton = driver.findElement(By.id("fetchDataButton"));
```



```
    // Submit the form to save changes
    WebElement saveButton = driver.findElement(By.id("fetchDataButton"));
    saveButton.click();
    WebElement successMessage = driver.findElement(By.id("successMessage"));

    // Assuming the "successMessage" element is displayed upon successful navigation.
    if (successMessage.isDisplayed()) {
        System.out.println("Test Passed: Successfully logged in and profile updated.");
    } else {
        System.out.println("Test Failed.");
    }
  }
```

Screenshot



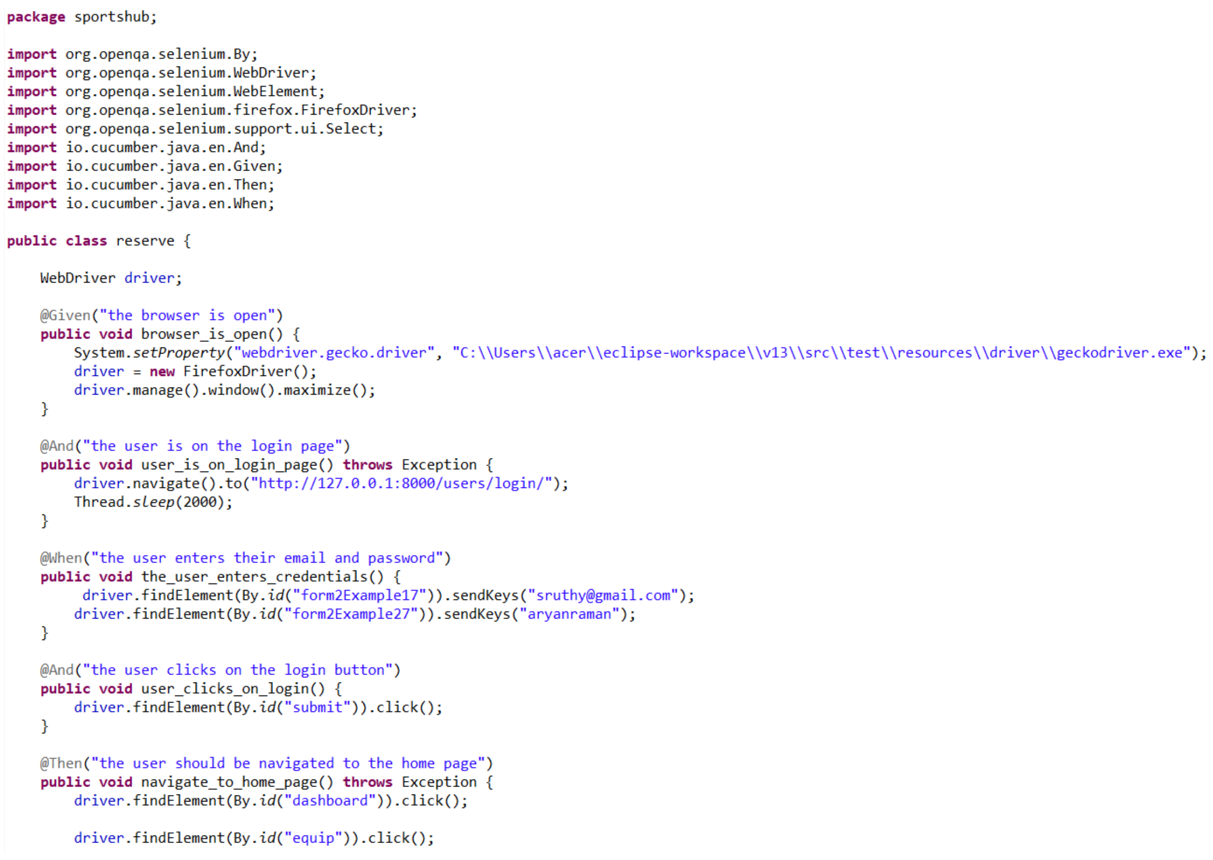
```
WebDriver BiDi listening on ws://127.0.0.1:32404
Read port: 4978
1698171965818   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:32404/devtools/browser/ad3189a2-7a89-4c20-a088-4f6bba1b8556
  Given the browser is open4                           # sportshub.zizu.browser_is_open()
console.error: Region.sys.mjs: "Error fetching region" (new Error("TIMEOUT", "resource://gre/modules/Region.sys.mjs", 759))
console.error: Region.sys.mjs: "Failed to fetch region" (new Error("TIMEOUT", "resource://gre/modules/Region.sys.mjs", 411))
console.error: (new TypeError("NetworkError: Network request failed", "resource://services-settings/Utils.sys.mjs", 229))
JavaScript error: http://127.0.0.1:8000/users/login/, line 191: TypeError: usernameField is null
  And the user is on the login page4                   # sportshub.zizu.user_is_on_login_page()
  When the user enters their email and password4       # sportshub.zizu.the_user_enters_credentials()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
1698171983546   Marionette      WARN    Ignoring event 'pageshow' because document has an invalid readyState of 'uninitialized'.
  And the user clicks on the login button4             # sportshub.zizu.user_clicks_on_login()
Test Passed: Successfully logged in and profile updated.
  Then the user should be navigated to the home page4      # sportshub.zizu.navigate_to_home_page()

1 Scenarios (1 passed)
5 Steps (5 passed)
0m21.235s
```

Project Name : SoulCure					
Login Test Case					
Test Case ID: 3			**Test Designed By: Vidhu krishnan vinod**		
Test Priority (Low/Medium/High): High			**Test Designed Date: 23-**10-2023		
Module Name: Zizu			**Test Executed By: Dr.Paulin Paul**		
Test Title: Open Ai gym assistant			**Test Execution Date: 23-**10-2023		
Description: Open Ai gym assistant					
Pre-Condition: User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to home page		Directed to homepage	home page displayed	Pass

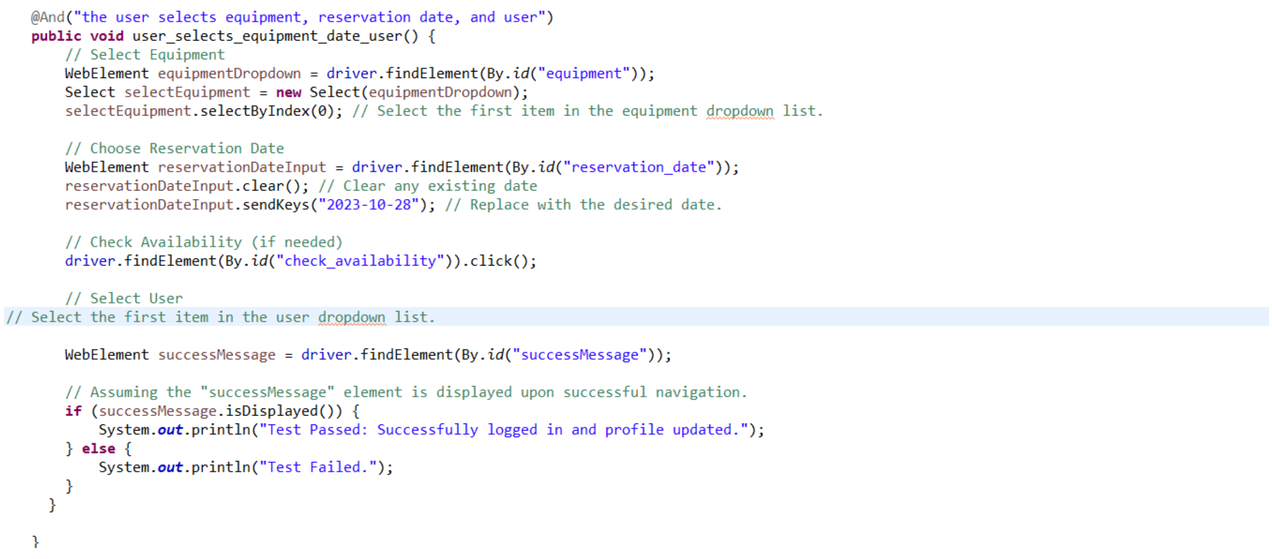
2	Select Zizu page		User should be able acces ai trainer	the user successfully accessed the AI trainer without encountering any errors or issues. The AI trainer's interface and functionalities were accessible and functional.	Pass
3	Select fetch data button to intiate ai gym				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 4:

Code

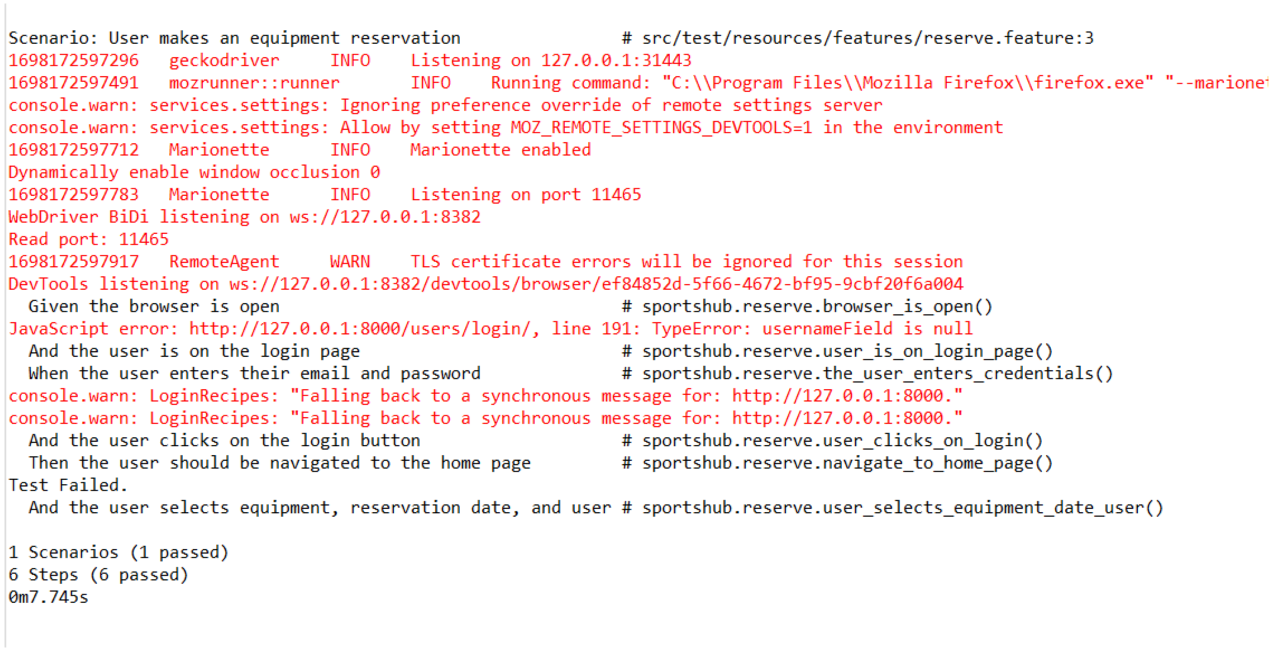


```
package sportshub;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;

public class reserve {

    WebDriver driver;

    @Given("the browser is open")
    public void browser_is_open() {
        System.setProperty("webdriver.gecko.driver", "C:\\Users\\acer\\eclipse-workspace\\v13\\src\\test\\resources\\driver\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
    }

    @And("the user is on the login page")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://127.0.0.1:8000/users/login/");
        Thread.sleep(2000);
    }

    @When("the user enters their email and password")
    public void the_user_enters_credentials() {
         driver.findElement(By.id("form2Example17")).sendKeys("sruthy@gmail.com");
        driver.findElement(By.id("form2Example27")).sendKeys("aryanraman");
    }

    @And("the user clicks on the login button")
    public void user_clicks_on_login() {
        driver.findElement(By.id("submit")).click();
    }

    @Then("the user should be navigated to the home page")
    public void navigate_to_home_page() throws Exception {
        driver.findElement(By.id("dashboard")).click();

        driver.findElement(By.id("equip")).click();
```



```
    @And("the user selects equipment, reservation date, and user")
    public void user_selects_equipment_date_user() {
        // Select Equipment
        WebElement equipmentDropdown = driver.findElement(By.id("equipment"));
        Select selectEquipment = new Select(equipmentDropdown);
        selectEquipment.selectByIndex(0); // Select the first item in the equipment dropdown list.

        // Choose Reservation Date
        WebElement reservationDateInput = driver.findElement(By.id("reservation_date"));
        reservationDateInput.clear(); // Clear any existing date
        reservationDateInput.sendKeys("2023-10-28"); // Replace with the desired date.

        // Check Availability (if needed)
        driver.findElement(By.id("check_availability")).click();

        // Select User
// Select the first item in the user dropdown list.

        WebElement successMessage = driver.findElement(By.id("successMessage"));

        // Assuming the "successMessage" element is displayed upon successful navigation.
        if (successMessage.isDisplayed()) {
            System.out.println("Test Passed: Successfully logged in and profile updated.");
        } else {
            System.out.println("Test Failed.");
        }
      }

    }
```

Screenshot



```
Scenario: User makes an equipment reservation                # src/test/resources/features/reserve.feature:3
1698172597296   geckodriver     INFO    Listening on 127.0.0.1:31443
1698172597491   mozrunner::runner       INFO    Running command: "C:\\Program Files\\Mozilla Firefox\\firefox.exe" "--marione
console.warn: services.settings: Ignoring preference override of remote settings server
console.warn: services.settings: Allow by setting MOZ_REMOTE_SETTINGS_DEVTOOLS=1 in the environment
1698172597712   Marionette      INFO    Marionette enabled
Dynamically enable window occlusion 0
1698172597783   Marionette      INFO    Listening on port 11465
WebDriver BiDi listening on ws://127.0.0.1:8382
Read port: 11465
1698172597917   RemoteAgent     WARN    TLS certificate errors will be ignored for this session
DevTools listening on ws://127.0.0.1:8382/devtools/browser/ef84852d-5f66-4672-bf95-9cbf20f6a004
  Given the browser is open                                  # sportshub.reserve.browser_is_open()
JavaScript error: http://127.0.0.1:8000/users/login/, line 191: TypeError: usernameField is null
  And the user is on the login page                          # sportshub.reserve.user_is_on_login_page()
  When the user enters their email and password              # sportshub.reserve.the_user_enters_credentials()
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
console.warn: LoginRecipes: "Falling back to a synchronous message for: http://127.0.0.1:8000."
  And the user clicks on the login button                    # sportshub.reserve.user_clicks_on_login()
  Then the user should be navigated to the home page         # sportshub.reserve.navigate_to_home_page()
Test Failed.
  And the user selects equipment, reservation date, and user # sportshub.reserve.user_selects_equipment_date_user()

1 Scenarios (1 passed)
6 Steps (6 passed)
0m7.745s
```

Project Name : SoulCure	
Login Test Case	
Test Case ID: 2	**Test Designed By: Vidhu krishnan vinod**
Test Priority (Low/Medium/High): High	**Test Designed Date: 23**-10-2023

Module Name: Equipment reservation			**Test Executed By: Dr.Paulin Paul**		
Test Title: Reserve equipment			**Test Execution Date: 23-**10-2023		
Description: : Check Availability for equipments					
Pre-Condition: User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation to home page		Directed to homepage	home page displayed	Pass
2	Select profile section		User should be able edit 2 nd contact number	User Logged in and navigated to the dashboard with records	Pass
3	Provide Valid Password	Number: “8330085575”			
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

CHAPTER 6
IMPLEMENTATION

6.1INTRODUCTION

During the implementation stage of a project, the theoretical design is transformed into a functional system, which is essential in achieving a successful outcome and user confidence in the system's effectiveness and accuracy. User training and documentation are the primary focus during this stage, and conversion typically takes place around the same time as user training or afterwards. Implementation involves converting the new system design into an operational one and can create chaos and confusion if not carefully planned or controlled. The implementation process encompasses all activities necessary to convert from the existing system to the new system, whether it be a totally new or modified system. It is crucial to provide a reliable system that meets organizational requirements. The system can only be implemented after thorough testing is done, and it is found to be working according to specifications. Feasibility checks are performed by system personnel, and the more complex the system being implemented, the more involved the system analysis and design effort required for 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 have to ensure that the resistance does not build up, as one has to 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 won’t take place.

6.2.1 User Training

The goal of user training is to give users the confidence to test and alter computer-based systems in order to finally accomplish the intended goals. Training becomes more important as systems get more complicated. As part of the user training process, participants are exposed to a variety of crucial tasks like data entering, handling error alerts, querying databases, calling up routines to generate reports, among others.

6.2.2 Training on the Application Software

The new application software requires users to first complete a basic computer literacy training course before utilizing it. They should be shown how to access help resources, traverse the software panels, correct entry errors, and update data that has already been entered. Specific topics pertaining to the user group and their function in using the system or its components should also be covered in the training. The training for the programme should be adapted to the requirements of various user groups and hierarchical levels.

6.2.3 System Maintenance

The software maintenance phase is an essential part of the software development life cycle and occurs after a software product has been successfully implemented and is performing useful work. Maintenance is necessary to ensure that the system remains adaptable to changes in the system environment. While finding mistakes is a part of software maintenance, it is not the only task involved. There are many other activities involved in maintenance, such as updating documentation, fixing bugs, enhancing existing features, and adding new features. Effective maintenance can help ensure that the software remains functional, reliable, and efficient over time.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In conclusion, the proposed project, "Sportshub," is an all-encompassing platform catering to the needs of both sports enthusiasts and fitness-conscious individuals. With a range of professional facilities and equipment available for booking, it ensures a seamless experience for users interested in sports such as football, basketball, cricket, tennis, and volleyball. Additionally, fitness enthusiasts can register and access a variety of fitness resources to achieve their fitness goals.

The platform features certified trainers who offer personalized training and closely monitor the daily activities of members, aligning with their chosen training schemes and payment plans. What sets this project apart is the integration of AI technology, specifically Pose Detection Machine Learning, which powers a virtual assistant gym trainer. Users can partake in virtual training sessions, benefitting from real-time analysis of their workout posture and precision. The system provides immediate feedback on posture violations and offers guidance to enhance performance. Furthermore, detailed progress reports are generated, allowing users to track their fitness journey.

This advanced feature brings the convenience and expertise of a professional gym trainer to users, whether they choose to work out in a gym or at home. The project is divided into four modules: Admin, Registered Sports User, Registered Gym User, and Registered Trainers. This report has covered the overall system, with partial functionality implemented for the Admin, Registered Gym User, and Registered Trainer modules within the scope of the mini-project.

7.2 FUTURE SCOPE

The "Sportshub" project holds substantial potential for further development and expansion. Here are some of the potential areas for future growth and enhancement:

Enhanced AI Integration: Continuous improvement and integration of more advanced AI technologies can lead to more precise fitness monitoring and training assistance.

Mobile Applications: Developing mobile applications for the platform can make it more accessible and convenient for users, enabling them to engage in workouts and track progress on the go.

Social Features: Incorporating social features that allow users to connect, compete, or collaborate with friends and other users can enhance user engagement and motivation.

Nutritional Guidance: Expanding the system to offer nutritional guidance and meal planning for users seeking a holistic approach to fitness.

Wearable Device Integration: Integrating with wearable fitness devices for real-time health and fitness data collection can offer valuable insights for users and trainers.

Community and Forums: Building a community and forums within the platform can create a space for users to share experiences, ask questions, and provide support to one another.

Multi-Language Support: Expanding the platform to support multiple languages can make it accessible to a broader international audience.

Gamification: Implementing gamification elements can make workouts and training more engaging and fun, motivating users to stay active.

Subscription Models: Offering different subscription tiers with varying features and benefits can help generate revenue and cater to a wider range of users.

User Feedback and Improvement: Regularly gathering user feedback and making continuous improvements based on their suggestions and needs.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- Pankaj Jalote, “So*ftware engineering*: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- https:/stackoverflow.com/
- https://www.youtube.com/brototype/
- https://app.diagrams.net/

CHAPTER 9
APPENDIX

Views.py

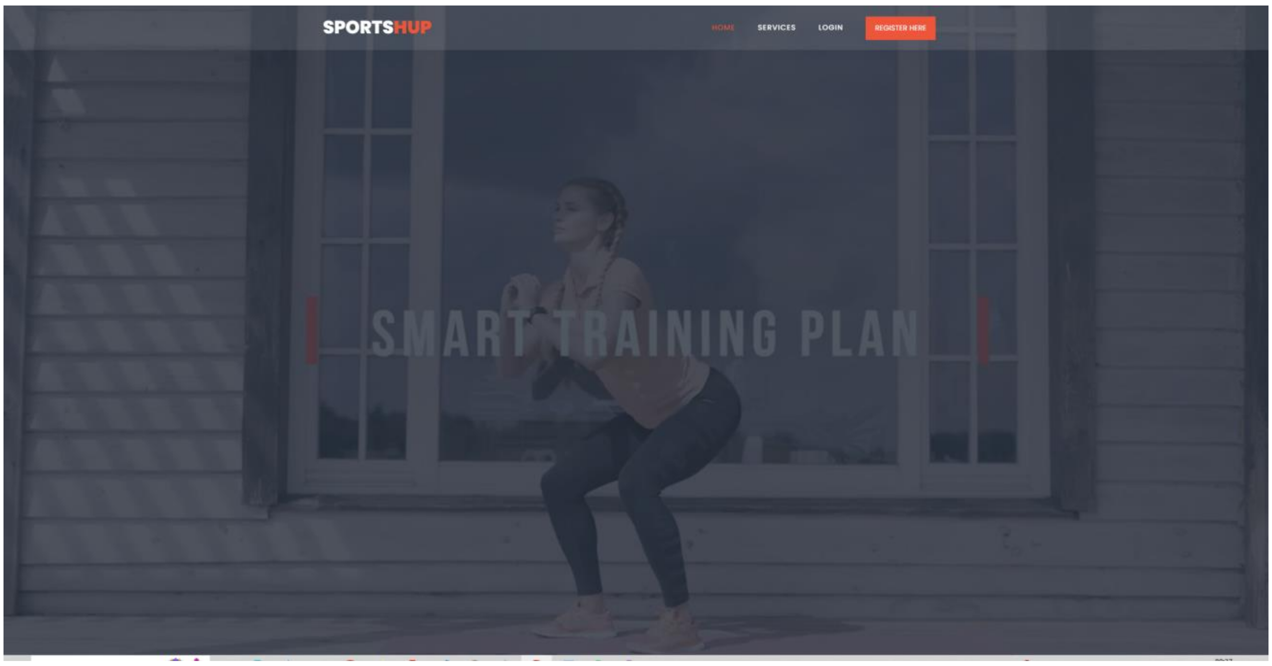
```
from django.shortcuts import render, redirect
from django.contrib.auth import get_user_model, login, logout, authenticate
from django.contrib import messages
from django.contrib.auth.forms import AuthenticationForm
from .decorators import user_not_authenticated
from .forms import RegistrationForm, UserLoginForm, UserUpdateForm
from .models import *
import uuid
from django.http import HttpResponseRedirect
from django.conf import settings
from django.core.mail import send_mail
from django.contrib.auth.decorators import login_required
from .decorators import admin_user_required

from django.urls import reverse
from django.contrib import messages,auth

def signup_redirect(request):
    messages.error(request, "Something wrong here, it may be that you already have account!")
    return redirect("/")



User = get_user_model()

@user_not_authenticated
# def register(request):
#     if request.method == "POST":
#         form = RegistrationForm(request.POST)
#         if form.is_valid():
#             user = form.save()
#             login(request, user, backend='django.contrib.auth.backends.ModelBackend')
#             return redirect('/')
#     else:
#         form = RegistrationForm()

#     return render(request, 'users/register.html', {'form': form})


def register(request):
    if request.method == 'POST':
        uname = request.POST.get('uname', None)
        email = request.POST.get('email', None)
        password = request.POST.get('pass', None)


        if uname and email  and password :
            if User.objects.filter(email=email).exists():
                messages.info(request,"Email already taken")
                return redirect('users:register')


            else:
```

```
            user = User(email=email,username=uname)
            user.set_password(password)  # Set the password securely
            user.save()
            messages.info(request,"registered")
            return redirect('users:login')

    return render(request, 'register.html')



@login_required
def custom_logout(request):
    logout(request)
    messages.info(request, "Logged out successfully!")
    return redirect("/")

from django.contrib.auth import authenticate, login
from django.contrib import messages
from django.contrib.auth.forms import AuthenticationForm
from django.shortcuts import render, redirect

@user_not_authenticated
def custom_login(request):
    if request.user.is_authenticated:
        return redirect("/")

    if request.method == "POST":
        form = AuthenticationForm(request=request, data=request.POST)
        if form.is_valid():
            user = authenticate(
                username=form.cleaned_data["username"],
                password=form.cleaned_data["password"],
            )
            if user is not None:
                login(request, user)
                messages.success(request, f"Hello <b>{user.username}</b>! You have been logged in ")
                # Add a message to indicate that fields are empty (no explicit check)
                messages.warning(request, 'complete your profile if you havent')
                return redirect("/")

        else:
            for error in list(form.errors.values()):
                messages.error(request, error)

    form = AuthenticationForm()

    return render(request,"login.html",context={"form": form})




# def profile(request, username):
#     if request.method == "POST":
#         form = UserUpdateForm(request.POST, instance=request.user)
#         if form.is_valid():
#             form.save()
#             messages.success(request, 'Your profile has been updated!')
#             return redirect("/")   # Redirect to the index (homepage)
#         else:
#             messages.error(request, 'There was an error updating your profile.')

#     form = UserUpdateForm(instance=request.user)
#     return render(
#         request=request,
#         template_name="users/profile.html",
#         context={"form": form}
#     )
# def password_reset_form(request):
#     return redirect("")




from django.shortcuts import render, redirect
from .forms import RoleApplicationForm

def role_application_view(request):
    # Check if the user already has one of the three roles
    has_role = CustomUser.objects.filter(
        username=request.user.username,
        role__in=['FitnessUser', 'FitnessTrainer', 'SportsTrainer']
    ).exists()

    # Check if the user already has a pending application
    existing_application = RoleApplication.objects.filter(user=request.user, is_approved=False).first()

    if has_role:
        return render(request, 'users/has_role.html')

    if existing_application:
        return render(request, 'users/application_pending.html')

    if request.method == 'POST':
        form = RoleApplicationForm(request.user, request.POST)
        if form.is_valid():
            form.save()
            return redirect('index')  # Replace with the appropriate URL name
    else:
        form = RoleApplicationForm(request.user)

    return render(request, 'users/role_application.html', {'form': form})




from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from django.views import View
from .models import CustomUser
from Members.models import FitnessUser, FitnessTrainer, SportsTrainer
from django.http import HttpResponse
from .forms import UserProfileForm, FitnessUserForm, FitnessTrainerForm, SportsTrainerForm

@login_required
def profile(request):
    user = request.user

    if request.method == "POST":
        user_form = UserProfileForm(request.POST, request.FILES, instance=user)
        role = user.role

        if role == "FitnessUser":
            role_form = FitnessUserForm(request.POST, instance=user.fitnessuser)
        elif role == "FitnessTrainer":
            role_form = FitnessTrainerForm(request.POST, instance=user.fitnesstrainer)
        elif role == "SportsTrainer":
            role_form = SportsTrainerForm(request.POST, instance=user.sportstrainer)
        else:
             role_form = None
             role_title = None

        if user_form.is_valid() and (role_form is None or role_form.is_valid()):
            user_form.save()
            if role_form:
                role_form.save()
            return redirect('users:profile')

    else:
        user_form = UserProfileForm(instance=user)
        role = user.role

        if role == "FitnessUser":
            role_form = FitnessUserForm(instance=user.fitnessuser)
        elif role == "FitnessTrainer":
            role_form = FitnessTrainerForm(instance=user.fitnesstrainer)
        elif role == "SportsTrainer":
            role_form = SportsTrainerForm(instance=user.sportstrainer)
        else:
            role_form = None
            role_title = None

    context = {
        'user_form': user_form,
        'role_form': role_form,
        'user':user,
    }

    return render(request, 'users/profile.html', context)
```

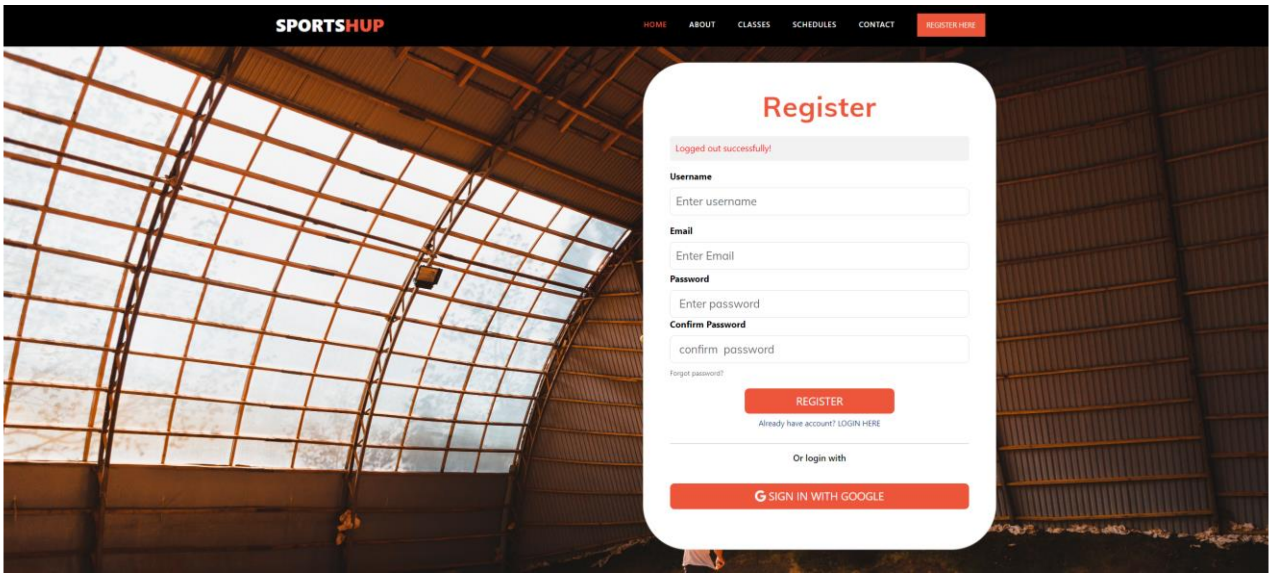
```
from django.shortcuts import render, redirect
from .models import RoleApplication
from Members.models import FitnessUser,FitnessTrainer, SportsTrainer
from .decorators import admin_user_required

@admin_user_required
def role_approval_view(request):
    applications_pending_approval = RoleApplication.objects.filter(is_approved=False)

    if request.method == 'POST':
        application_id = request.POST.get('application_id')
        application = RoleApplication.objects.get(id=application_id)

        # Check which action button was clicked (Approve or Reject)
        action = request.POST.get('action')

        if action == 'approve':
            application.is_approved = True
            application.save()

            # Update the user's role to the approved role
            user = application.user
            user.role = application.role.name
            user.save()

            # If the user's role is 'FitnessUser', create a FitnessUser instance
            if user.role == 'FitnessUser':
                fitness_user = FitnessUser(
                    user=user,
                    fitness_goal=application.fitness_goal,
                    height=application.height,
                    weight=application.weight,
                )
                fitness_user.save()
            elif user.role == 'FitnessTrainer':
                fitness_trainer = FitnessTrainer(
                    user=user,
                    experience=application.experience,
                    certification=application.certification,
                    training_goal=application.specialization_details,
                    certification_link=application.certification_link,
                )
                fitness_trainer.save()
            elif user.role == 'SportsTrainer':
                sports_trainer = SportsTrainer(
                    user=user,
                    specialization=application.specialization_details,
                )
                sports_trainer.save()

        elif action == 'reject':
            application.delete()  # Delete the rejected application

        return redirect('index')  # Replace with the appropriate URL name
    return render(request, 'users/role_approval.html', {'applications_pending_approval':
applications_pending_approval})
```

9.1 Screen Shots

Landing Page



Registration Page



Register

Logged out successfully!

Username

Enter username

Email

Enter Email

Password

Enter password

Confirm Password

confirm password

Forgot password?

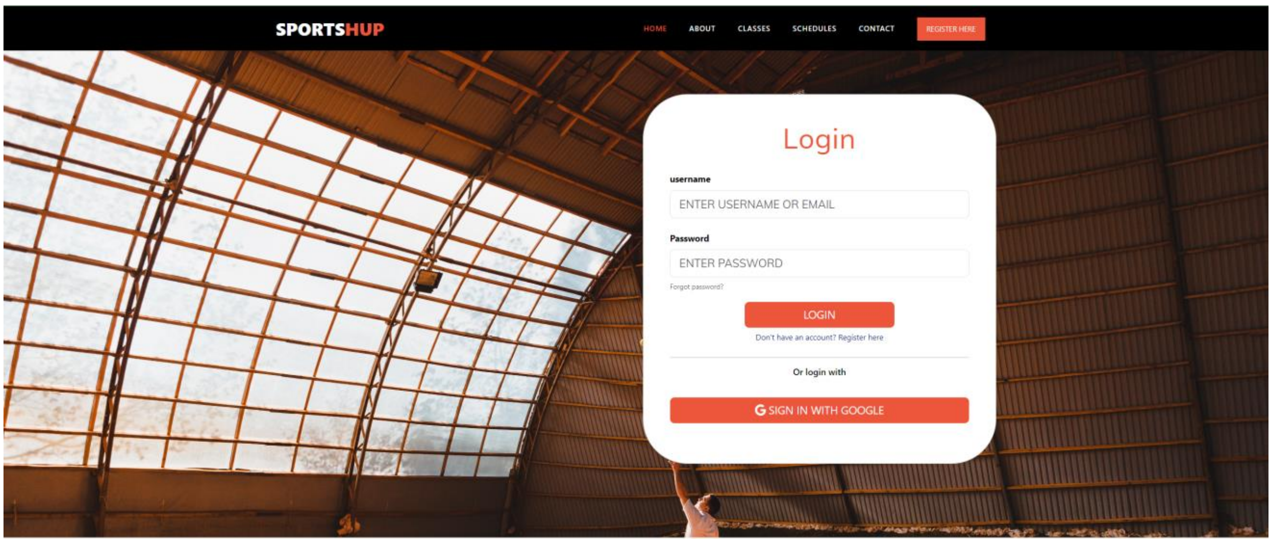
REGISTER

Already have account? LOGIN HERE

Or login with

SIGN IN WITH GOOGLE

Login Page



Login

username

ENTER USERNAME OR EMAIL

Password

ENTER PASSWORD

Forgot password?

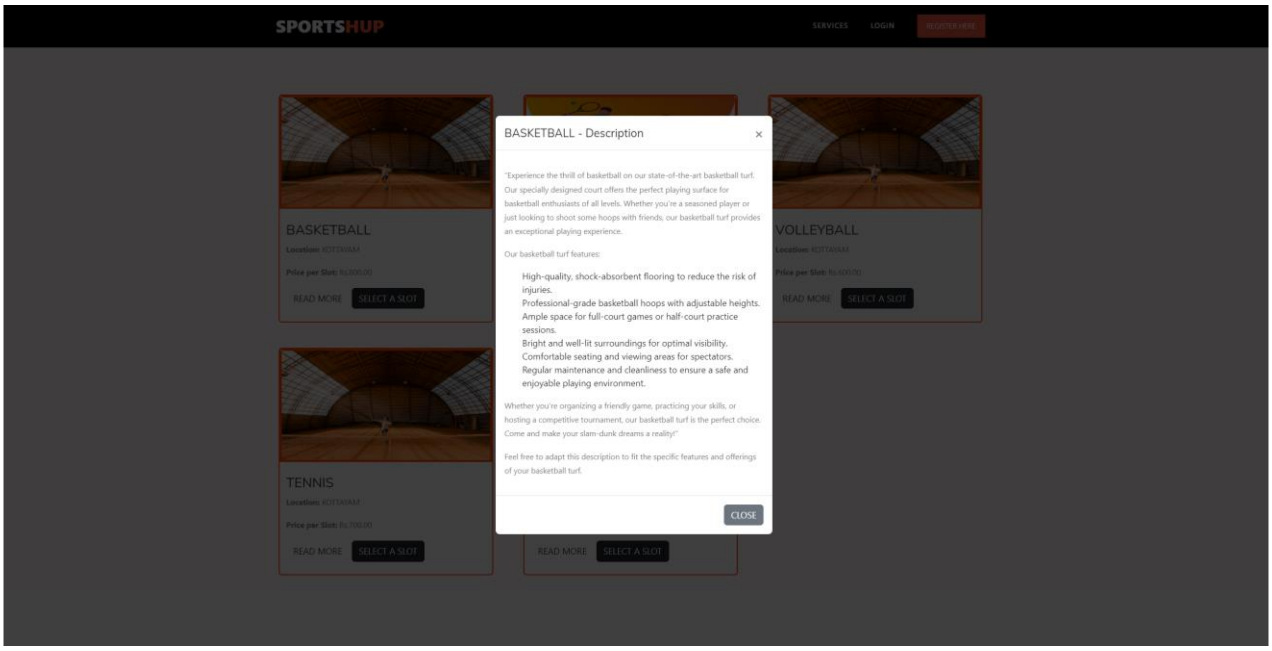
LOGIN

Don't have an account? Register here

Or login with

SIGN IN WITH GOOGLE

Services Page



BASKETBALL - Description

"Experience the thrill of basketball on our state-of-the-art basketball turf. Our specially designed court offers the perfect playing surface for basketball enthusiasts of all levels. Whether you're a seasoned player or just looking to shoot some hoops with friends, our basketball turf provides an exceptional playing experience.

Our basketball turf features:

High-quality, shock-absorbent flooring to reduce the risk of injuries.
Professional-grade basketball hoops with adjustable heights.
Ample space for full-court games or half-court practice sessions.
Bright and well-lit surroundings for optimal visibility.
Comfortable seating and viewing areas for spectators.
Regular maintenance and cleanliness to ensure a safe and enjoyable playing environment.

Whether you're organizing a friendly game, practicing your skills, or hosting a competitive tournament, our basketball turf is the perfect choice. Come and make your slam-dunk dreams a reality!"

Feel free to adapt this description to fit the specific features and offerings of your basketball turf.