Jordan Basketball Management System

Shadip Kumar Joshi

B.Sc. (Hons.) Computing, Softwarica College of IT and E-commerce, Coventry University

ST5007CEM Web Development

Hari Saran Shrestha

August 10, 2023

Acknowledgement

I would like to express my heartfelt gratitude to all those who contributed to the successful completion of this website named Jordan Basketball Management System. Special thanks go to my module leader Hari Saran Shrestha for his valuable guidance, encouragement, and support throughout this project.

Additionally, I extend my appreciation to my peers and seniors for their constructive feedback and insightful discussions. Lastly, I am indebted to the developers and individuals who generously shared their knowledge and resources, enabling me to enhance my understanding and deliver this work.

This report would not have been possible without the collective efforts and cooperation of all involved. Thank you for being an essential part of this endeavour.

Table of Contents

Table of Figures

Jordan Basketball Management System

Introduction

In the basketball world, Michael Jordan's name resonates strongly as an iconic representation of excellence and unparalleled talent. The Jordan Basketball Management System (JBMS) takes its name as a tribute to his impact and enduring legacy.

At its core, the JBMS serves as a comprehensive tool in the form of website that empowers players and coaches alike, enhancing their engagement with the sport and optimizing their basketball journey. This user-friendly digital application seamlessly facilitates a wide array of services viz searching basketball courts, reserving free courts, management of playgrounds, communication, etc at the convenience of one's fingertips.

Figure 1 shows the logo of this project.

Figure 1

Logo of the application

Keywords

Figure 2

Keywords for Jordan Basketball Management System

Aim

Jordan Basketball Management System aims to provide an online platform for easy booking of basketball courts for matches and efficient management for owners.

Objectives

The objectives of “JBMS” are to:

- streamline court-booking process for players and enthusiasts.
- provide a user-friendly platform to check availability of courts in real-time and secure bookings effortlessly.
- deliver a seamless and immersive user-experience to cater to the player’s, coach’s, and court owner’s needs.
- simplify court management tasks like bookings, tournaments scheduling, etc for effective utilization of court facilities.
- provide valuable insights into court usage patterns, player preferences, and review feedback.
- fosters a vibrant basketball community to encourage social interaction among players, coaches, and enthusiasts, and facilitating team formation and friendly matches.
- establish a smooth communication channel between players and court owners to facilitate easy contact for inquiries, feedback, and addressing any concerns promptly.
- provides a review and rating system enabling users to share their experiences and feedback.
- optimize court usage, minimize downtime and maximize revenue for court owners through efficient scheduling and management.
- support and promote local basketball tournaments encouraging community engagement and providing a platform for players to showcase their talent.
- improve its features based on user feedback and technological advancements ensuring it remains at the forefront of basketball management systems.

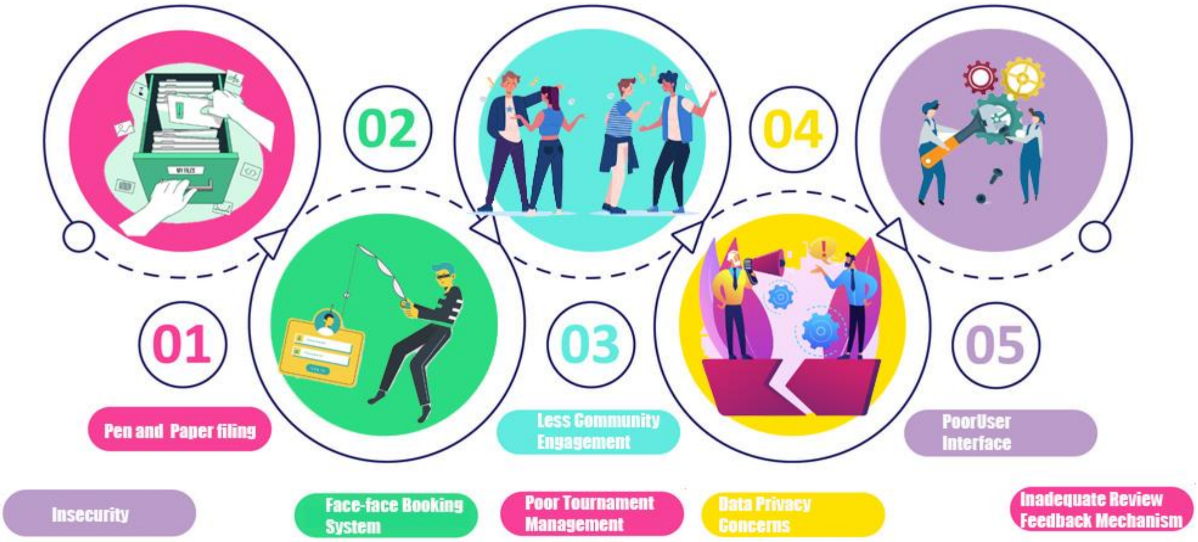
Problem Statement

Problem

Players face various hurdles and problems to book basketball courts that meet their needs and requirements of infrastructures and time. Many courts still use traditional slow and cumbersome court booking procedures through pen and paper booking that leads to frustration among players and court owners. It lacks the ability to provide real-time updates on court availability, leading to miscommunications and missed opportunities for players. There is no provision of any data analytics, leaving players, coaches, and court owners without valuable insights to improve decision-making and court utilization and it even is prone to errors, document loss. Lack of a structured review and feedback system in this method, makes it difficult for players to share experiences and for potential users to make informed decisions. Due to lack of adequate security measures and data privacy policies concerns about the confidentiality of personal information may potentially arise. Additionally, it may lead to potential booking conflicts and inefficient coordination for its absence of effective communication channels between players and court owners. Figure 3 shows some problems in for basketball management system at present.

Figure 3

Problem statement in basketball management system

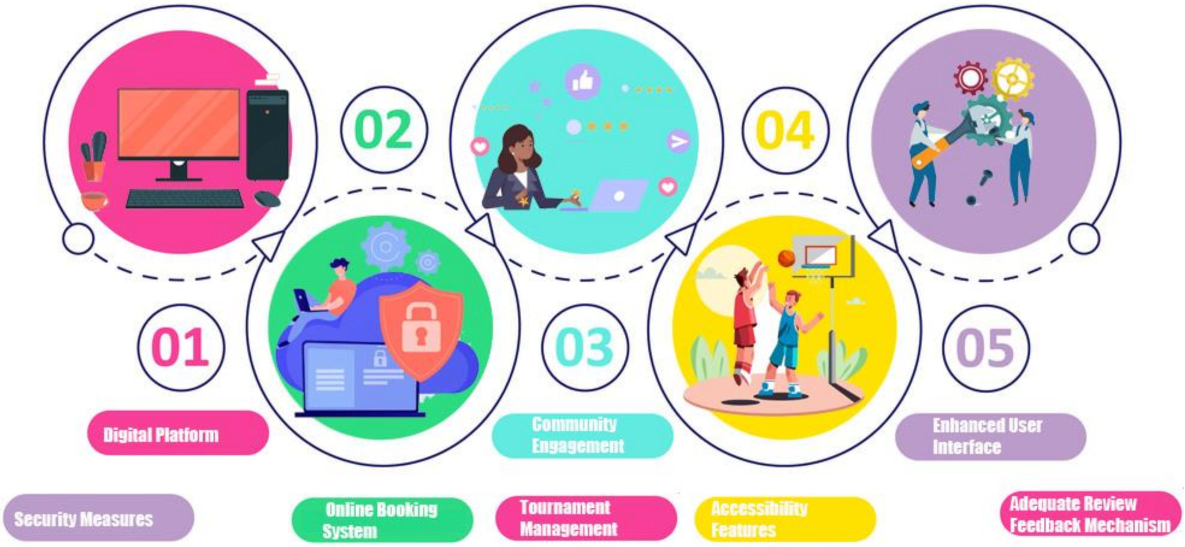


Solution

To address the problems suffered by players in court booking, this JBMS website provides a user-friendly and intuitive online booking platform that allows players to check real-time court availability, make reservations, and manage bookings seamlessly. Implementing a digital system eliminates the slow and cumbersome nature of pen and paper booking. It also incorporates data analytics into its systems to gather insights on court utilization, player preferences, and peak booking times to optimize court allocation and improve the overall user experience. To eliminate miscommunications and enhances coordination it implements a centralized communication system within digital platforms to enable seamless and direct communication between players and court owners, thereby fostering social engagement. Additionally, it has a structured review and feedback system to encourage players to share their experiences and rate court facilities. This information helps potential users make informed decisions. Finally, it incorporates a robust security protocols and data privacy measures to safeguard users' personal information and ensure confidentiality. Figure 4 depicts some solutions for basketball management system.

Figure 4

Solutions to problems in basket management system.



Features and Functionalities

The features of this applications are:

- **User Registration**: New users can register easily by providing essential details, such as name, email, and password.
- **Player Profile Creation**: Players can create personalized profiles, showcasing their achievements, skills, preferred positions, and basketball history.
- **Secure User Login**: The app employs robust security measures to ensure secure user login, protecting user data and preventing unauthorized access.
- **Password Reset**: Users can initiate a password reset process in case they forget their password, receiving a password reset link via email for account recovery.
- **Court Owner Profile Creation**: Court owners can create detailed profiles for their venues, including court images, facilities, rates, and available time slots.
- **Edit Profile:** Users can easily edit and update their profiles, enabling them to keep their information current and relevant.
- **Account Verification:** To ensure the authenticity of user accounts, the app may implement a verification process, such as email verification or mobile number verification.
- **User Dashboard**: Each user is provided with a personalized dashboard that displays relevant information, such as upcoming bookings, training schedules, and recent activity.
- **Booking History**: Users can access their booking history, which includes past and upcoming court reservations, allowing them to track their basketball activities.
- **Multi-Platform Accessibility**: JBMS is accessible across multiple platforms, including web browsers, smartphones, and tablets, ensuring users can use the app from any device with internet connectivity.

Functional Requirements

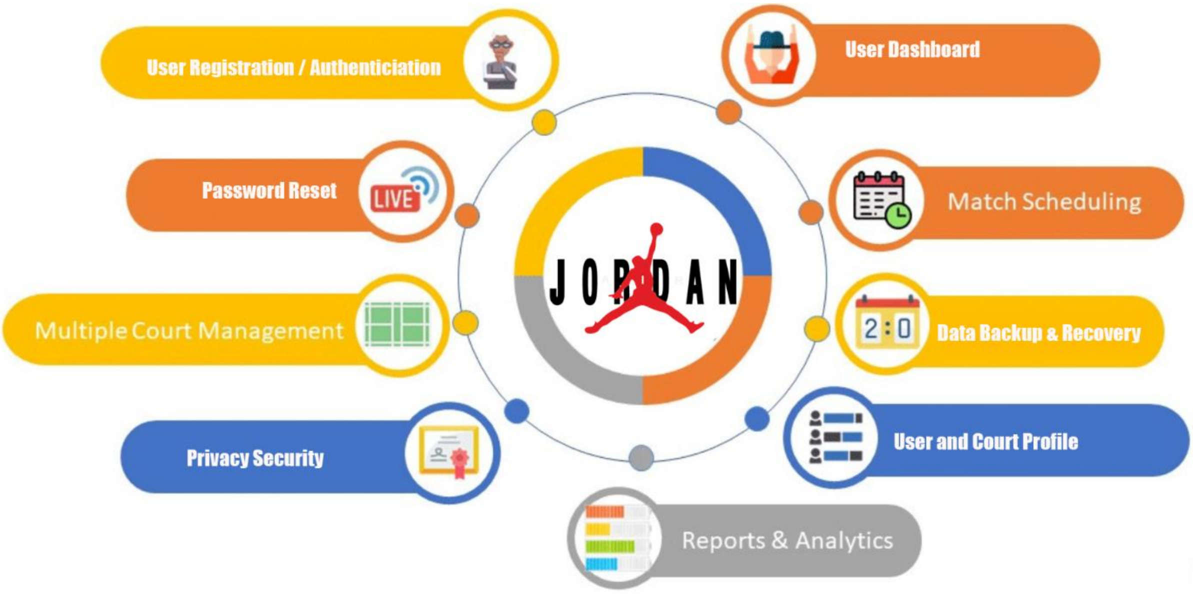
Functional requirements, detail the actions the software needs to perform and describe its capabilities in a comprehensive manner, are as follows:

- **User Registration**: The system should allow new users to register by providing their name, email, and password.
- **Authentication:** User registration should be verified through email or mobile number verification to ensure authenticity.
- **Player Profile Management**: Users should be able to create and edit personalized player profiles.
- **Court Owner Profile Management:** Court owners should have the ability to create and modify detailed profiles for their venues. Court profiles should include court images, facilities, rates, and available time slots.
- **Password Reset**: Users should be able to initiate a password reset process if they forget their password through email.
- **Account Deletion:** Users should have the option to delete their account if they wish to discontinue using the JBMS.
- **Privacy Security**: The software should employ robust security measures to protect user data, including encryption of sensitive information and secure transmission of data.

Figure 5 depicts some functional requirements of a proper basketball management system.

Figure 5

Functional Requirements of Basketball Management System.



Non-functional Requirements

Non-functional requirements, which define how the system should perform, are as follows:

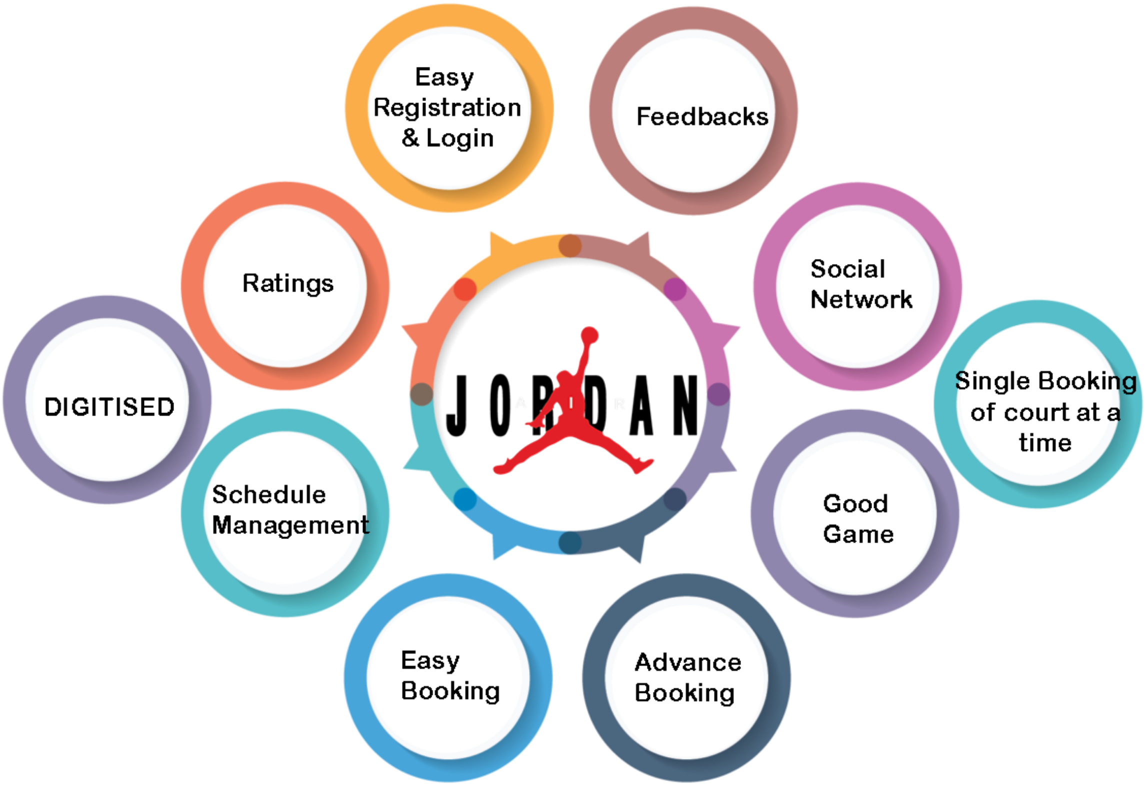
- **Usability:** JBMS should have an intuitive and user-friendly graphical user interface (GUI), ensuring that users can easily navigate and interact with the application.
- **Security**: The system should implement robust security measures to safeguard user data and ensure secure login and logout mechanisms. Users should only have access to their own credentials and data.
- **Performance**: Transitions between different interfaces and actions triggered by buttons should be smooth, with response times not exceeding 10 seconds.
- **Data Changeability**: The system should efficiently handle modifications to the database, ensuring that updates and changes are processed swiftly without any data inconsistencies.
- **Capacity**: JBMS should be capable of handling concurrent usage by a minimum of 200 users without significant degradation in performance, ensuring smooth operations during peak usage times.
- **Scalability**: The application should be designed with scalability in mind, allowing for future upgrades and expansions to accommodate growing user demands and additional features.
- **Portability**: JBMS should be compatible with various operating systems and mobile devices, enabling users to access and utilize the application seamlessly on their preferred platforms.
- **Compatibility**: The system's requirements should not be overly demanding, allowing it to run smoothly on different hardware configurations without compromising performance.
- **Connectivity**: The software should maintain a reliable connection to the internet, ensuring continuous accessibility and functionality even in low-bandwidth environments.

Scope

The scope of this program is defined by its aim, objectives, problem statements, features, functional requirements, and non-functional requirements outlined earlier, as depicted in Figure 6.

Figure 6

Scope of Jordan Basketball Management System



Development Methodology

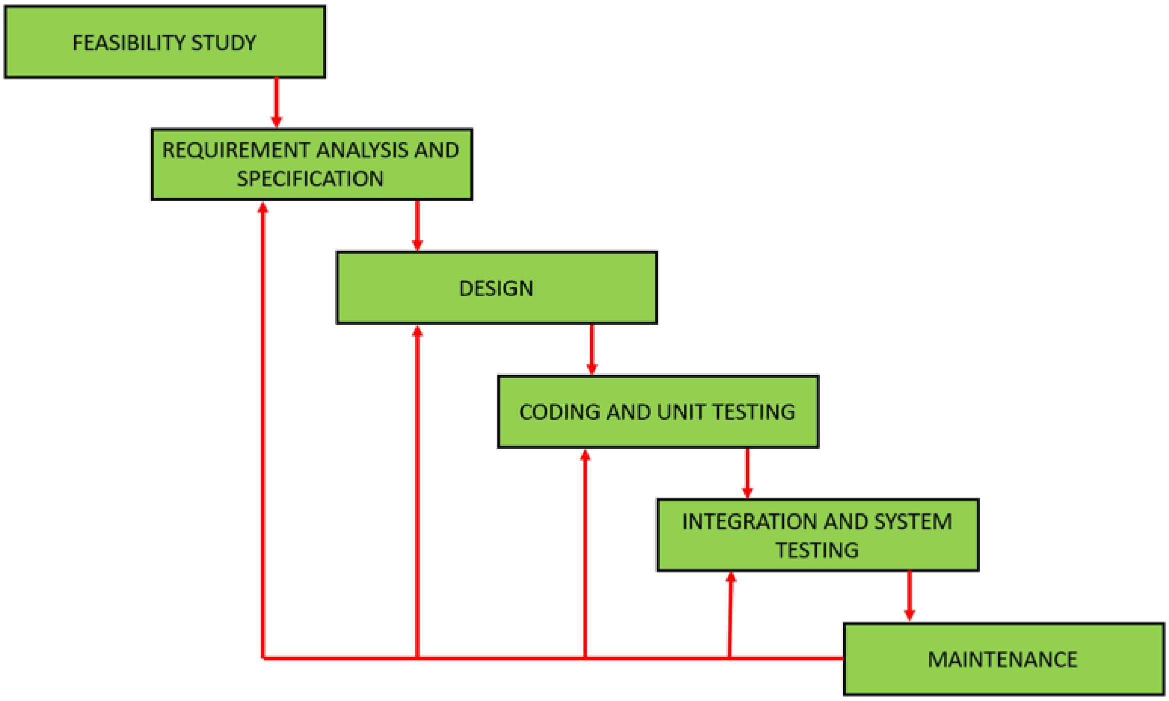
Development methodology pertains to the process known as the Software Development Life Cycle (SDLC), which outlines the sequential stages involved in creating software from its inception to completion.

Methodology

The Modern Waterfall Model, also referred to as the Iterative Model, is employed for the development of this software. It is preferred due to its systematic and easily understandable phases, making it suitable for individuals new to the IT field. As novice web developers in the realm of Software Engineering, this method is well-suited for us. Figure 7 illustrates all the stages in this model.

Figure 7

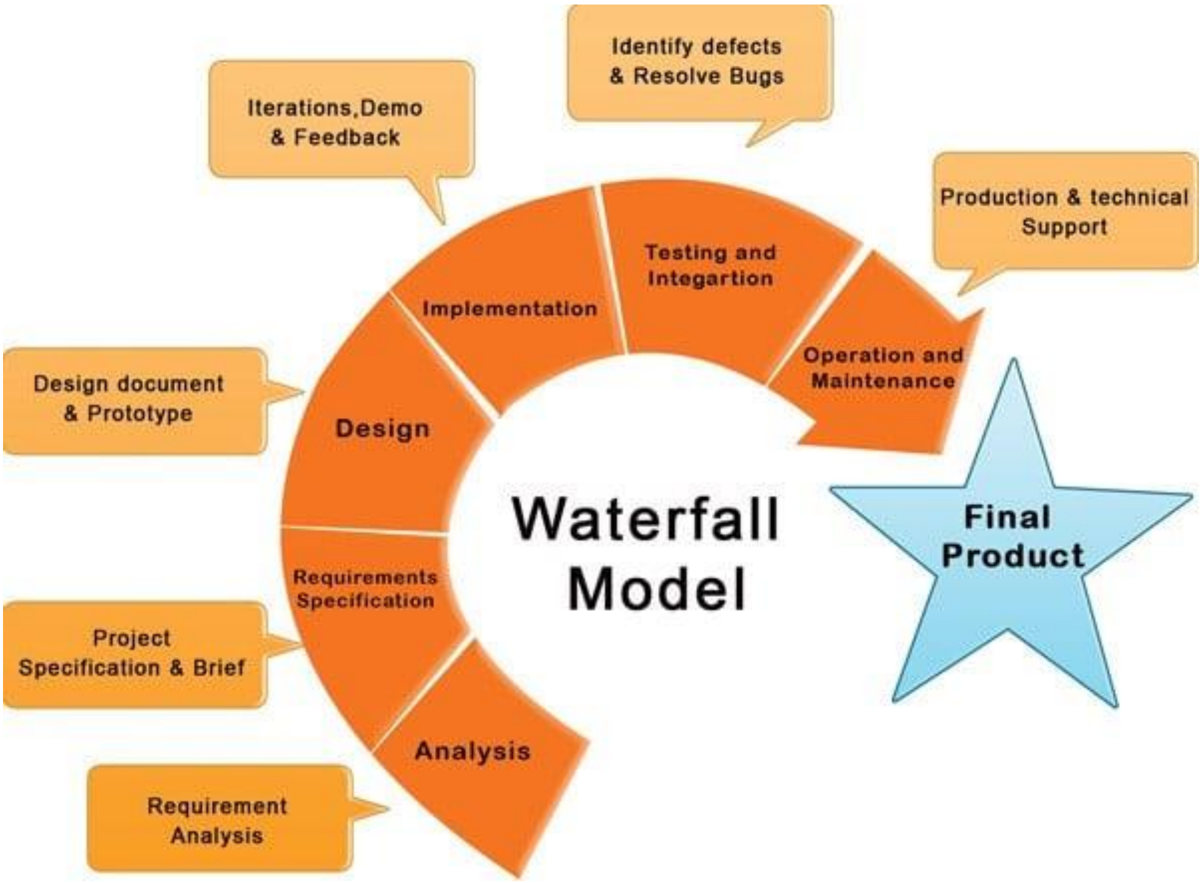
Stages of waterfall method *(Singhal, 2021).*


FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

The developing began by conducting a 'Feasibility study' to assess the practicality of the software through discussions and research. Following that, the 'Requirement Analysis and Specification' phase involved gathering and analyzing requirements, leading to the creation of a Software Requirement Specification (SRS) document. This document detailed functional and non-functional requirements, objectives, features, problem statements, scope, and other software specifics. Next, a structured 'Design' was iteratively finalized using an object-oriented approach, considering various objects in both the problem domain and solution domain, along with their relationships. The design was then implemented into source code using the Java programming language during the 'Development' phase, resulting in a functional software that underwent unit testing and incremental integration in the subsequent 'Integration and System-testing' phase. With the completion of testing, the software was ready for release to users and was regularly maintained during the 'Deployment and Maintenance' stage (Nitika, 2018). Throughout the process, the various phases were revisited as needed to accommodate new user requirements and feedback that arose during development (Singhal, 2021). Figure 8 provides an overview of all the phases of the waterfall SDLC, along with their respective characteristics.

Figure 8

Phases and their characteristics of Waterfall Model (Nitika, 2018).

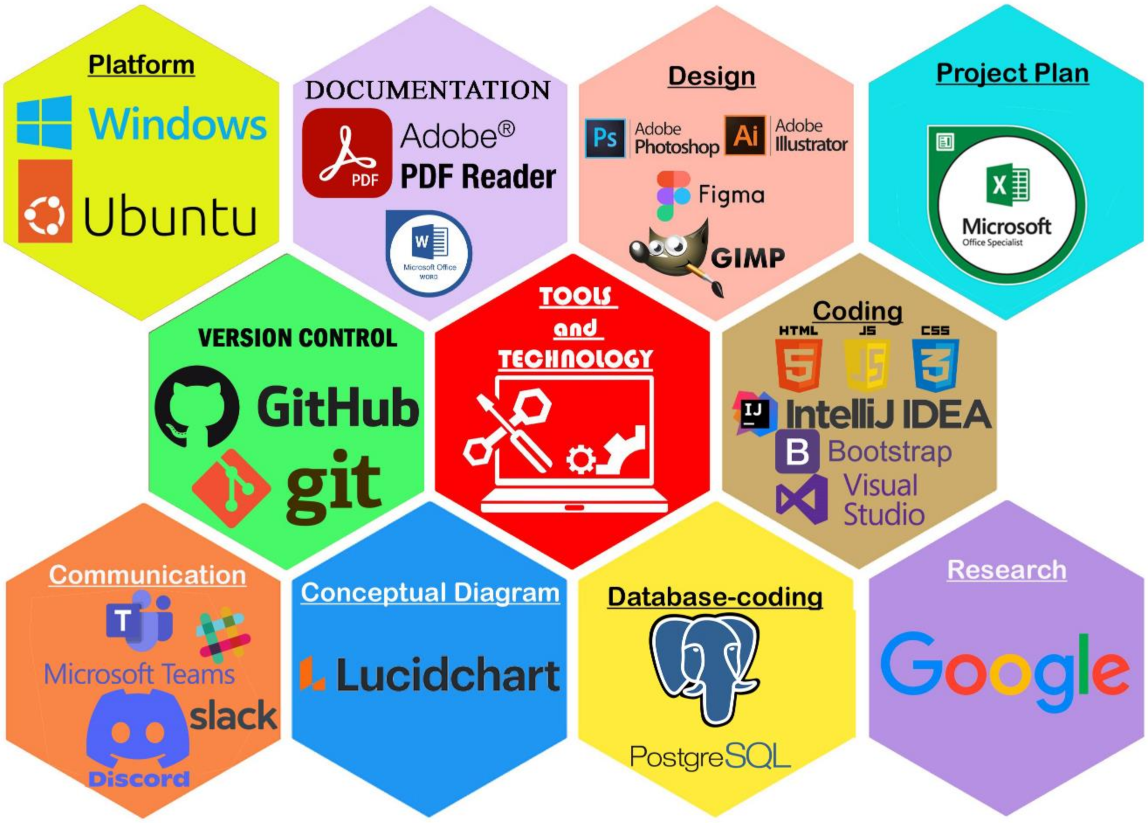

Identify defects & Resolve Bugs
Iterations,Demo & Feedback
Production & technical Support
Testing and Integartion
Implementation
Operation and Maintenance
Design document & Prototype
Design
Waterfall Model
Final Product
Requirements Specification
Project Specification & Brief
Analysis
Requirement Analysis

Tools and Technologies

Various tools and technologies were used in the entire SDLC as shown in figure 9.

Figure 9

Tools and Technologies used in JBMS

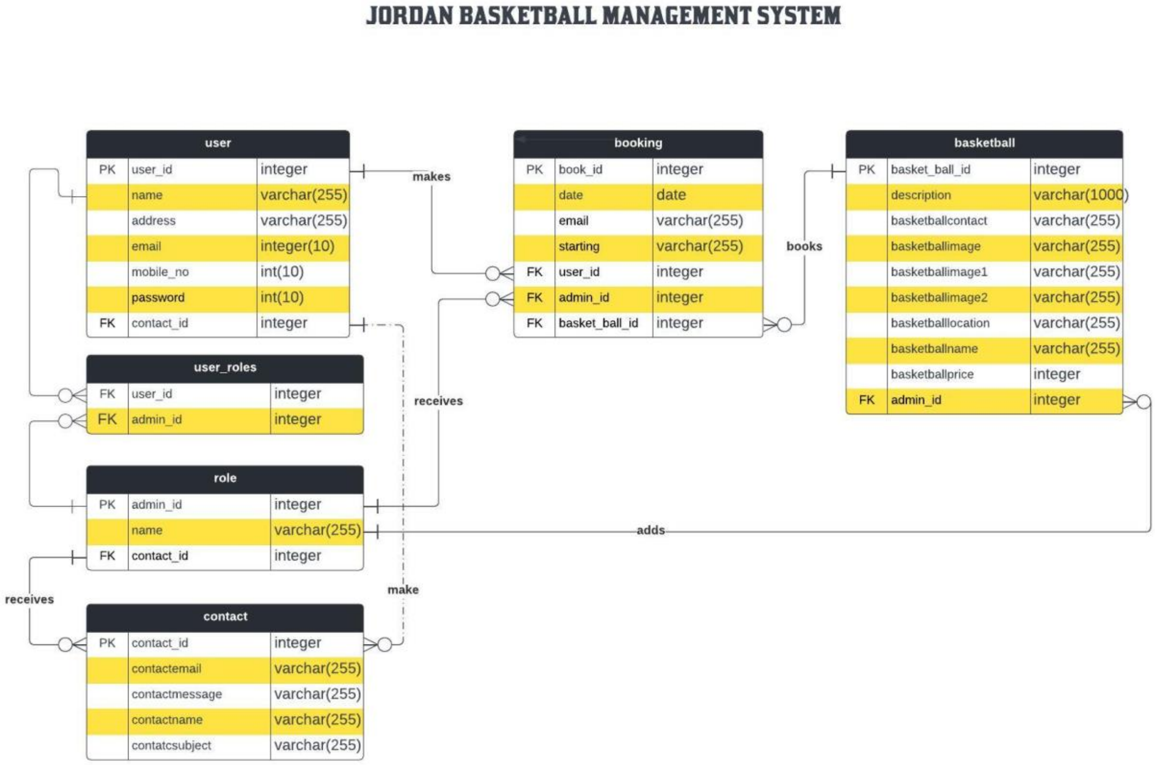


Conceptual Diagram

Entity-Relationship Diagram explains the relationship between entities in a system diagrammatically. Figure 10 and 11 demonstrates ER diagram between entities and use case diagram respectively in this project.

Figure 10

ERD of JBMS



user		
PK	user_id	integer
	name	varchar(255)
	address	varchar(255)
	email	integer(10)
	mobile_no	int(10)
	password	int(10)
FK	contact_id	integer

booking		
PK	book_id	integer
	date	date
	email	varchar(255)
	starting	varchar(255)
FK	user_id	integer
FK	admin_id	integer
FK	basket_ball_id	integer

basketball		
PK	basket_ball_id	integer
	description	varchar(1000)
	basketballcontact	varchar(255)
	basketballimage	varchar(255)
	basketballimage1	varchar(255)
	basketballimage2	varchar(255)
	basketballlocation	varchar(255)
	basketballname	varchar(255)
	basketballprice	integer
FK	admin_id	integer

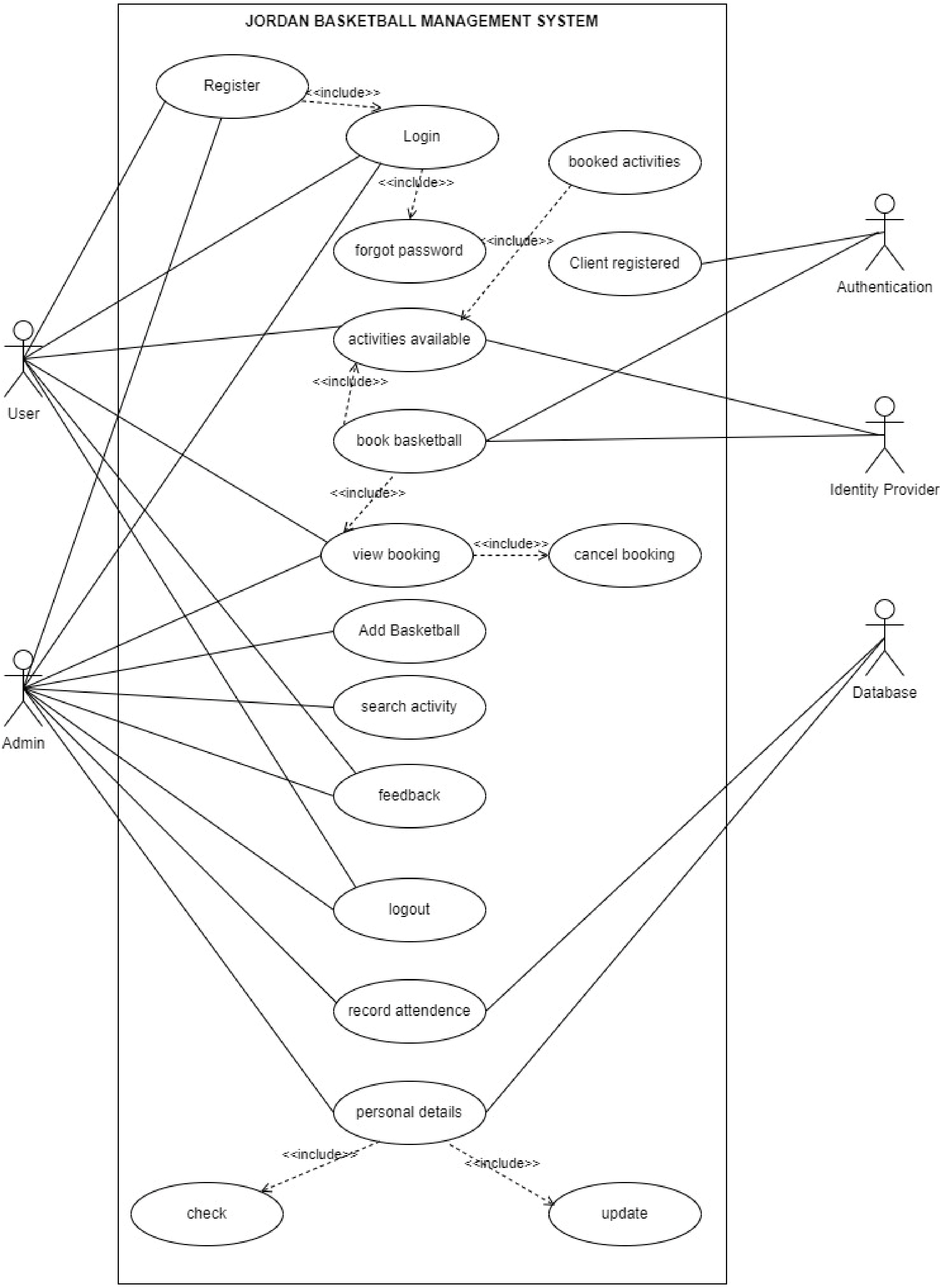
user_roles		
FK	user_id	integer
FK	admin_id	integer

role		
PK	admin_id	integer
	name	varchar(255)
FK	contact_id	integer

contact		
PK	contact_id	integer
	contactemail	varchar(255)
	contactmessage	varchar(255)
	contactname	varchar(255)
	contatcsubject	varchar(255)

Figure 11

Use case diagram



System Architecture

The structure and behavior of a good software are defined by its appropriate system architecture.

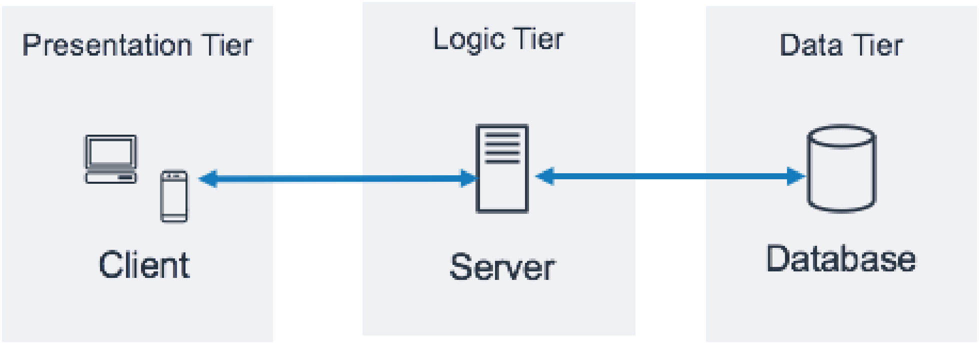
Three Tier Architecture

The three-tier architecture, illustrated in Figure 12, is employed to organize an application into three separate computing tiers: the presentation tier, the application tier, and the data tier. This approach allows each tier to function independently on different infrastructures, providing benefits such as isolated development, updates, and scalability without affecting the other tiers *(IBM, n.d.).*

The presentation tier, also known as the front-end, serves as the user interface and communication layer of the software. It facilitates user interactions, displaying information and collecting input. For this project, the presentation tier is developed using HTML, CSS, JavaScript and Bootstrap. The application tier, often referred to as the logic tier or middle tier, is the core of the application. It processes data received from the presentation tier and performs various operations such as data processing, addition, deletion, or modification. In this project, the application tier is developed using Java. The data tier, also known as the database tier or back-end, is responsible for storing and managing the processed information. PostgreSQL is utilized as the data tier for this software. One notable advantage of the three-tier architecture is improved security, as the presentation tier and data tier cannot communicate directly; all communication between these tiers must pass through the application tier *(IBM, n.d.).*

Figure 12

3-Tier Architecture *(AWS Whitepaper, n.d.).*

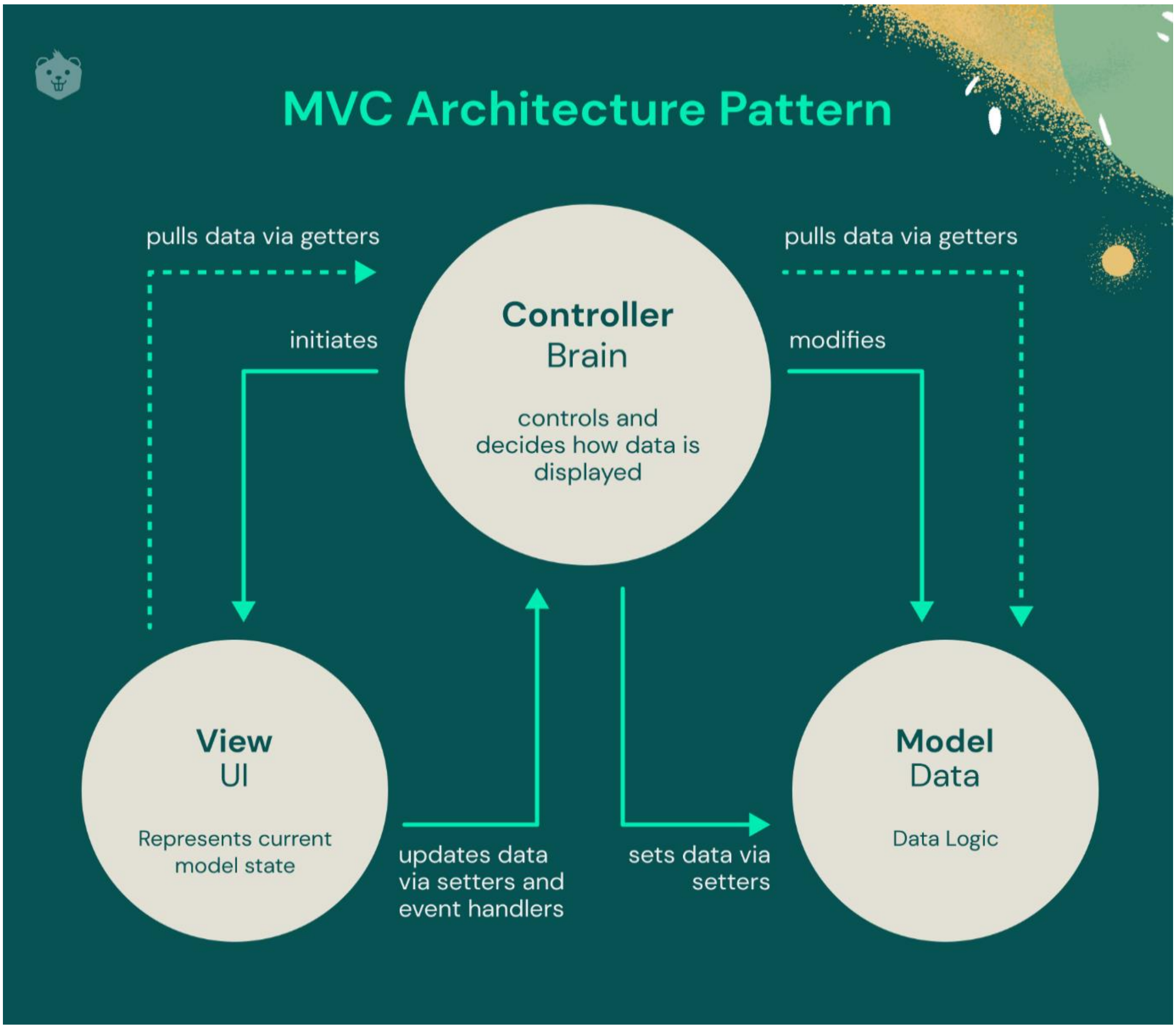


MVC Pattern

The MVC (Model-view-controller) pattern simplifies complex application development by dividing it into manageable parts, separating the frontend and backend codes. This enables multiple developers to work on the software simultaneously without causing conflicts. The pattern consists of three components: the model, view, and controller, as depicted in figure 13. The model represents the backend, containing data and related logic. The view serves as the frontend or graphical user interface (GUI), displaying the data from the model to the user. The controller acts as the application's central control unit, handling user input and converting it into commands for the application. *(Hernandez, 2021)*.

Figure 13

MVC Pattern mvc (Singh Bhui, 2022).



Software Quality

In my individual project, I emphasized software quality by incorporating both SOLID principles and Lean methodology.

SOLID principles

By following SOLID principles, I ensured that my codebase was robust, maintainable, and easy to understand. The Single Responsibility principle allowed me to divide responsibilities among different modules, enhancing code readability and reducing the likelihood of errors. Implementing the Open/Closed principle allowed me to extend the software's functionality seamlessly without modifying existing code, reducing the risk of introducing bugs. I also adhered to the Liskov Substitution principle, ensuring that different components could be interchanged without compromising the overall system's integrity. Emphasizing Interface Segregation led me to create precise interfaces, promoting efficient communication between different parts of the application and minimizing the impact of changes. Lastly, applying the Dependency Inversion principle resulted in a loosely coupled architecture, making testing easier and enhancing the software's stability *(Oloruntoba, 2021).*

Lean Methodology

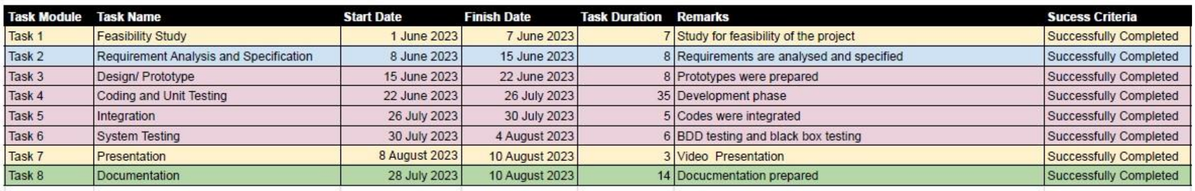
To further elevate software quality, I incorporated Lean methodology into my development process. Frequent iterations and a continuous improvement mindset allowed me to quickly identify and address any issues or changes in requirements. Regular feedback loops enabled timely adjustments, leading to a more refined and user-friendly end product. By prioritizing customer value, I focused on implementing the most critical features and eliminating unnecessary complexities, ensuring that the software aligned perfectly with the needs and preferences of end-users. Lean's emphasis on resource optimization and waste reduction streamlined my development efforts, maximizing productivity and minimizing delays (Simplilearn, 2023).

Project Plan

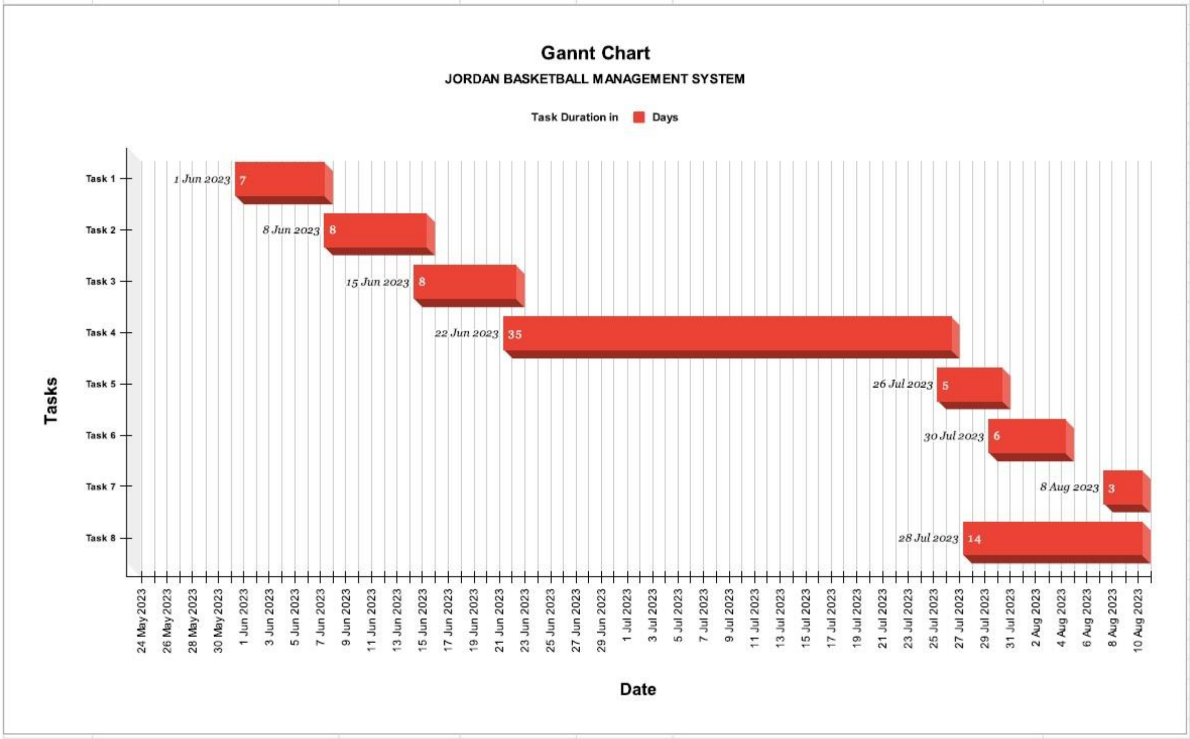
In software engineering, a progress chart is essential for effectively monitoring the time, effort, and advancement of software development. Gannt Chart was utilized to document the progress plan for all phases of the modern waterfall SDLC, as depicted in figure 14. This chart allows for a visual representation of the project's timeline and tasks, helping the team stay organized and track their development milestones.

Figure 14

Milestone and Gannt Chart of JBMS



Task Module	Task Name	Start Date	Finish Date	Task Duration	Remarks	Sucess Criteria
Task 1	Feasibility Study	1 June 2023	7 June 2023	7	Study for feasibility of the project	Successfully Completed
Task 2	Requirement Analysis and Specification	8 June 2023	15 June 2023	8	Requirements are analysed and specified	Successfully Completed
Task 3	Design/ Prototype	15 June 2023	22 June 2023	8	Prototypes were prepared	Successfully Completed
Task 4	Coding and Unit Testing	22 June 2023	26 July 2023	35	Development phase	Successfully Completed
Task 5	Integration	26 July 2023	30 July 2023	5	Codes were integrated	Successfully Completed
Task 6	System Testing	30 July 2023	4 August 2023	6	BDD testing and black box testing	Successfully Completed
Task 7	Presentation	8 August 2023	10 August 2023	3	Video Presentation	Successfully Completed
Task 8	Documentation	28 July 2023	10 August 2023	14	Docucmentation prepared	Successfully Completed



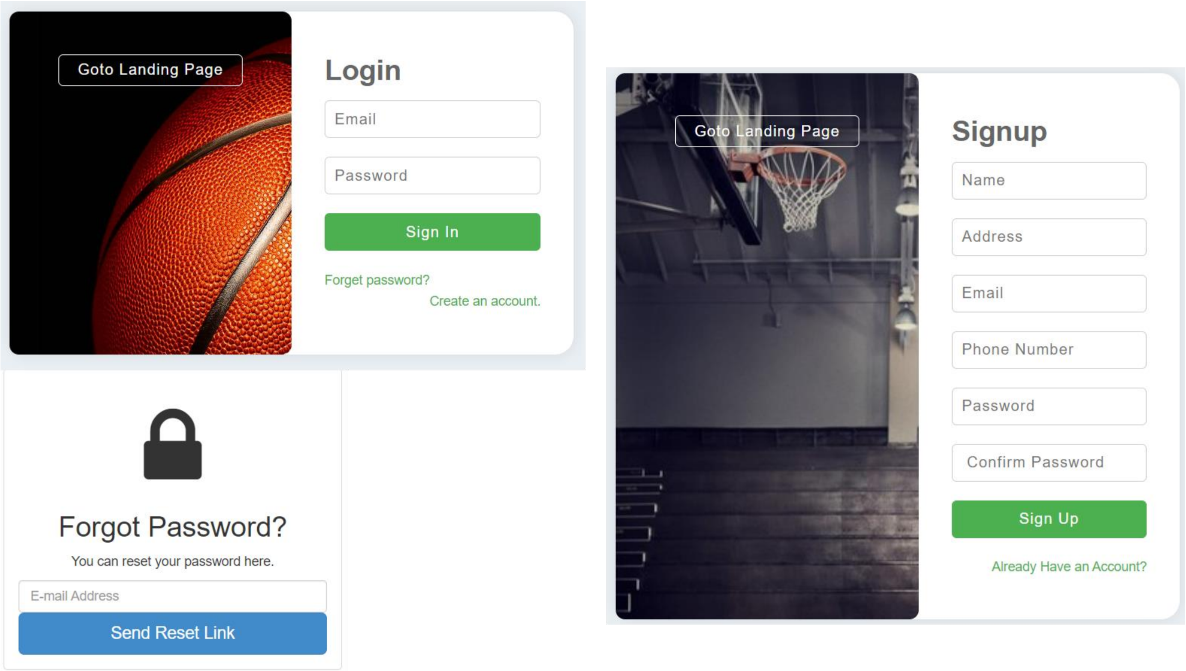
Prototypes

Figures 15-17 illustrates several types of prototypes, which are early models, or releases of software built for testing a concept or process, in this project *(Simplilearn, 2022).*

High Fidelity

Figure 15

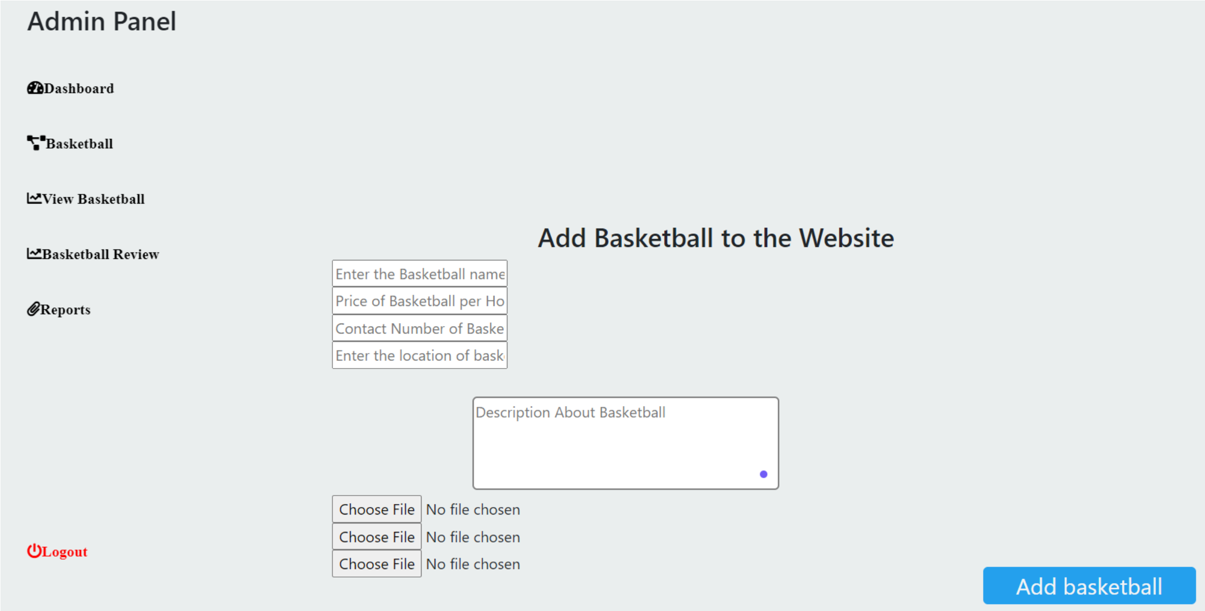
High fidelity prototype of Login, Signup and Forgot Password



You can reset your password here.

Figure 16

Prototype of Admin Dashboard



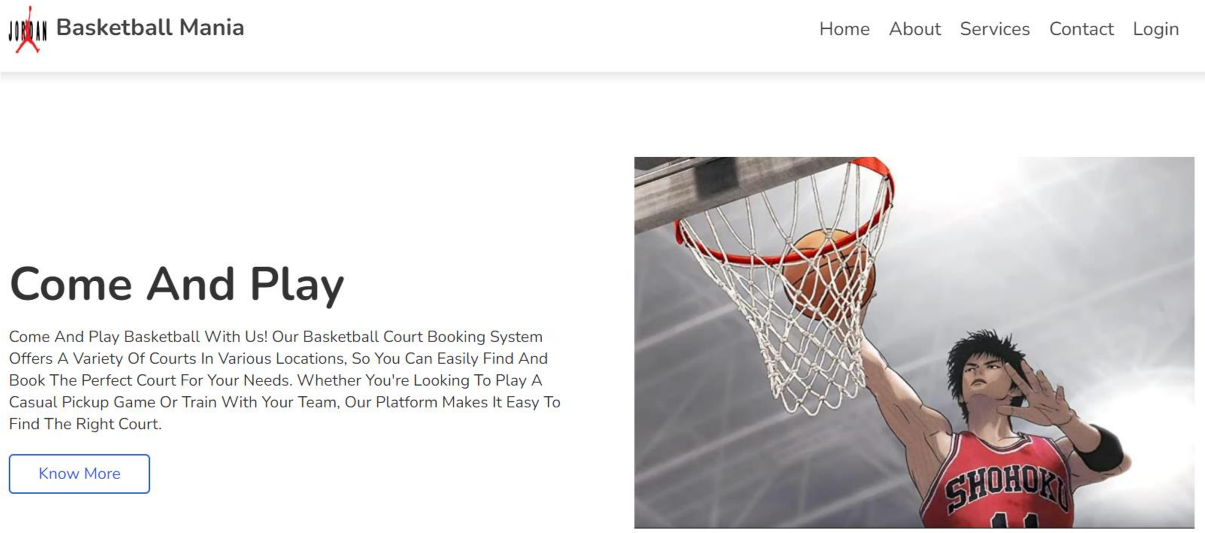
Admin Panel

Add Basketball to the Website

Dashboard
Basketball
View Basketball
Basketball Review
Reports
Logout
Enter the Basketball name
Price of Basketball per Ho
Contact Number of Baske
Enter the location of bask
Description About Basketball
Choose File No file chosen
Choose File No file chosen
Choose File No file chosen
Add basketball

Figure 17

High fidelity prototype of Landing Page



Basketball Mania

Come And Play

Come And Play Basketball With Us! Our Basketball Court Booking System Offers A Variety Of Courts In Various Locations, So You Can Easily Find And Book The Perfect Court For Your Needs. Whether You're Looking To Play A Casual Pickup Game Or Train With Your Team, Our Platform Makes It Easy To Find The Right Court.

Developed System

The software was built with responsive design in mind. Responsive design is essential for the JBMS, ensuring seamless access across devices. It adapts its layout and content to fit different screen sizes, offering intuitive features like real-time court availability and appointment scheduling. This approach eliminates the need for separate versions for various devices, streamlining development and maintenance. The system optimizes loading times, delivering appropriate content and touch-friendly interfaces for smooth browsing, which is useful for mobile users having limited internet connectivity. JBMS also supports different screen orientations, allowing users to switch effortlessly between portrait and landscape modes. Comprehensive testing ensures consistent functionality and a visually appealing experience, resulting in a robust platform for basketball enthusiasts, players, and management alike. Figures 18-20 show JBMS's responsiveness.

Figure 18

Responsiveness in WEB browser

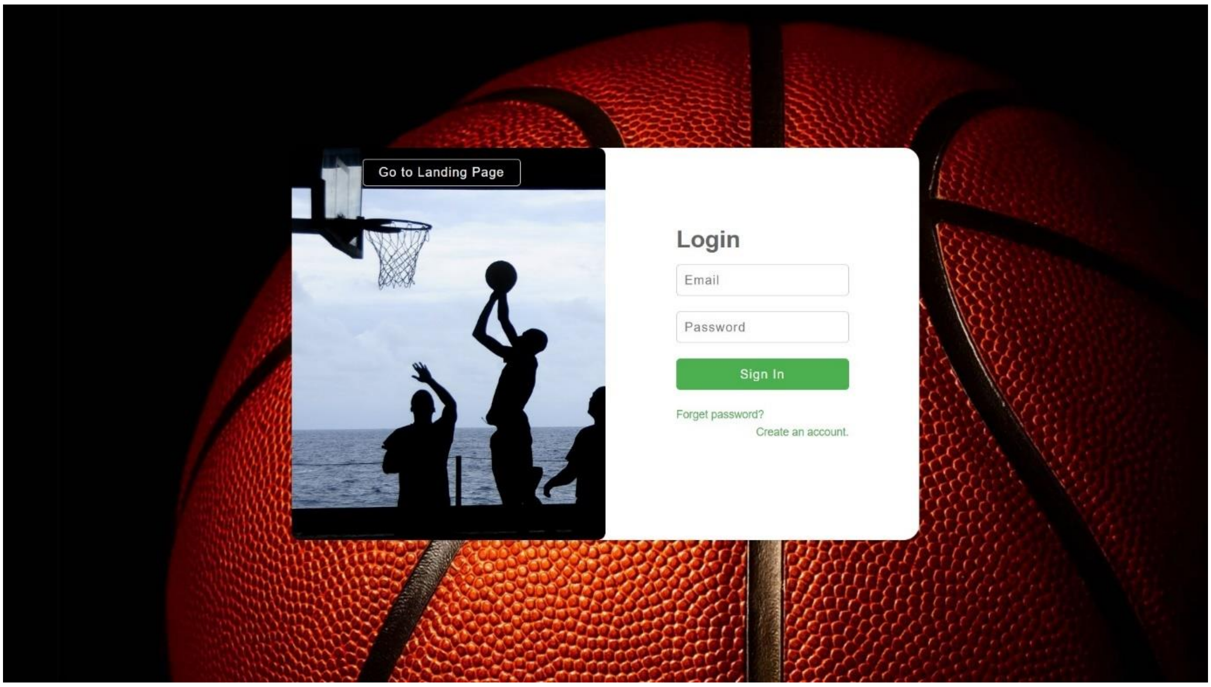
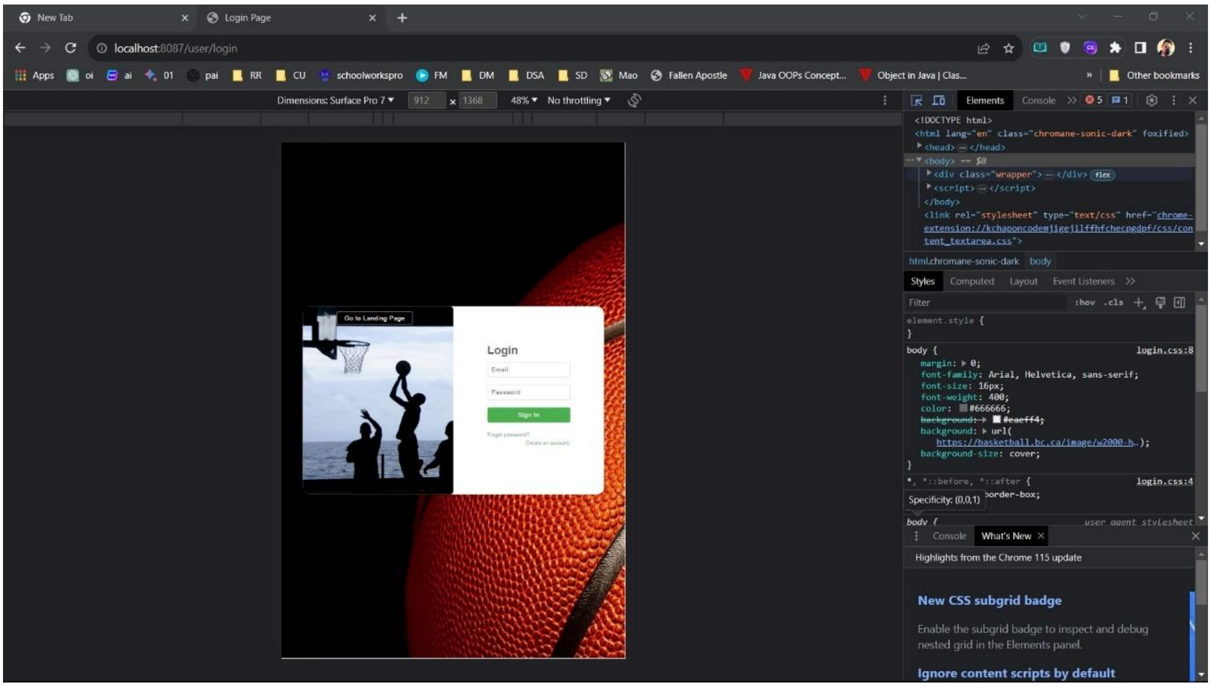


Figure 19

Responsiveness in WEB tablet



```
<!DOCTYPE html>
<html lang="en" class="chromane-sonic-dark" foxified>
 <head> ⋯ </head>
··· <body> == $0
   <div class="wrapper"> ⋯ </div> flex
   <script> ⋯ </script>
  </body>
  <link rel="stylesheet" type="text/css" href="chrome-extension://kchaponcodemjigejilffhfchecpgdpf/css/content_textarea.css">
```

```
element.style {
}
body {                                            login.css:8
  margin: ▸ 0;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 16px;
  font-weight: 400;
  color: #666666;
  ~~background: → #eaeff4;~~
  background: ▸ url(
    https://basketball.bc.ca/image/w2000-h…);
  background-size: cover;
}
*, *::before, *::after {                          login.css:4
  box-sizing: border-box;
body {                                   user agent stylesheet
```

Highlights from the Chrome 115 update

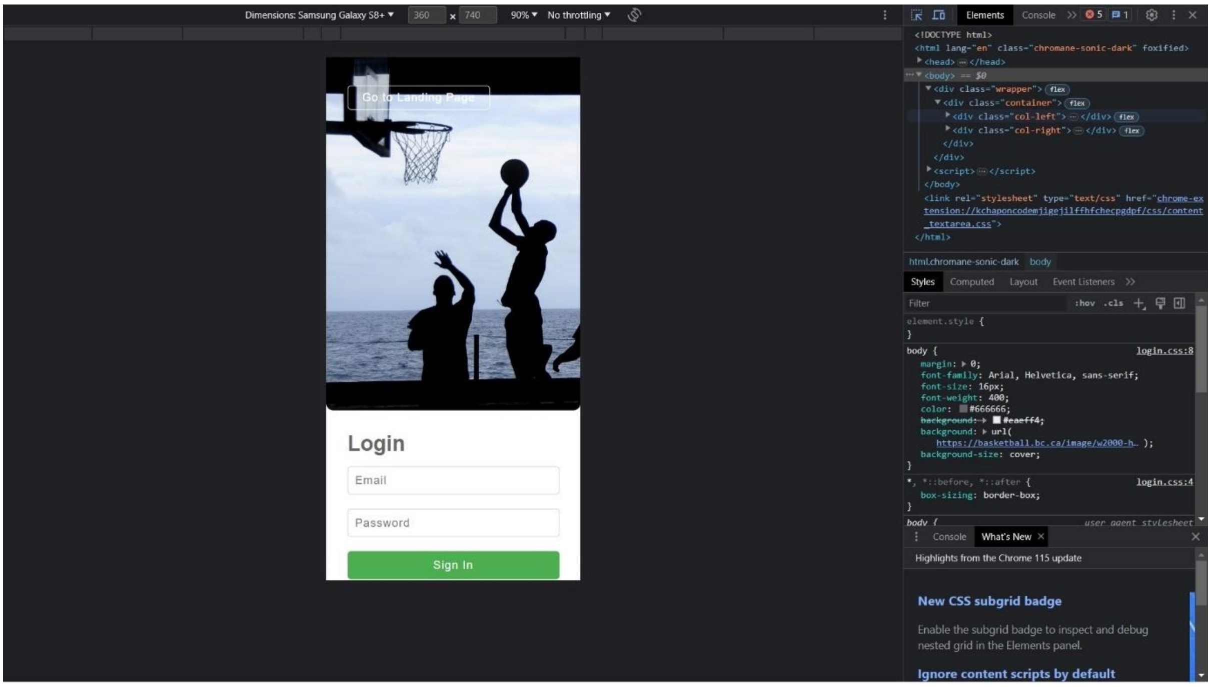
New CSS subgrid badge

Enable the subgrid badge to inspect and debug nested grid in the Elements panel.

Ignore content scripts by default

Figure 20

Responsiveness in smartphone



```
<!DOCTYPE html>
<html lang="en" class="chromane-sonic-dark" foxified>
 <head> ⋯ </head>
··· <body> == $0
   <div class="wrapper"> flex
     <div class="container"> flex
       <div class="col-left"> ⋯ </div> flex
       <div class="col-right"> ⋯ </div> flex
     </div>
   </div>
   <script> ⋯ </script>
  </body>
  <link rel="stylesheet" type="text/css" href="chrome-extension://kchaponcodemjigejilffhfchecpgdpf/css/content_textarea.css">
</html>
```

```
element.style {
}
body {                                            login.css:8
  margin: ▸ 0;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 16px;
  font-weight: 400;
  color: #666666;
  ~~background: → #eaeff4;~~
  background: ▸ url(
    https://basketball.bc.ca/image/w2000-h… );
  background-size: cover;
}
*, *::before, *::after {                          login.css:4
  box-sizing: border-box;
}
body {                                   user agent stylesheet
```

Highlights from the Chrome 115 update

New CSS subgrid badge

Enable the subgrid badge to inspect and debug nested grid in the Elements panel.

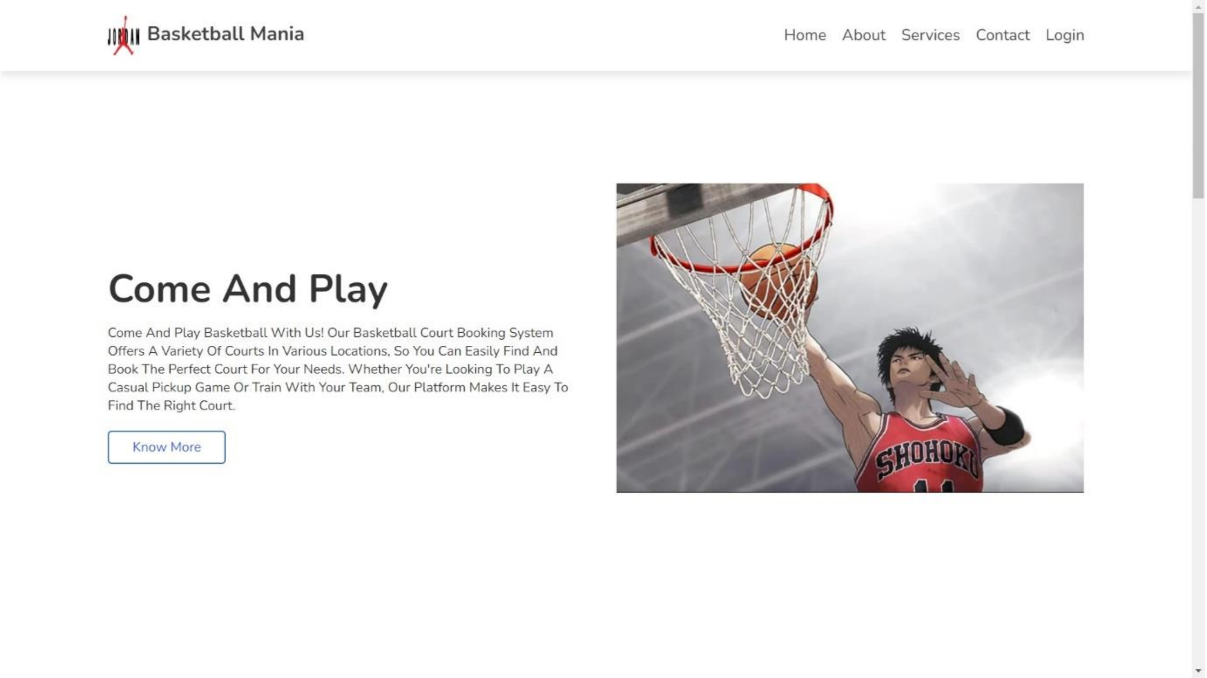
Ignore content scripts by default

Landing Page

It opens when the website of JBMS is searched on the internet and provides all the information regarding this software, it services, and contact information for the players to check out without registering to the site. It also has option to navigate to the login page as shown in figure 21.

Figure 21

Landing page



Come And Play

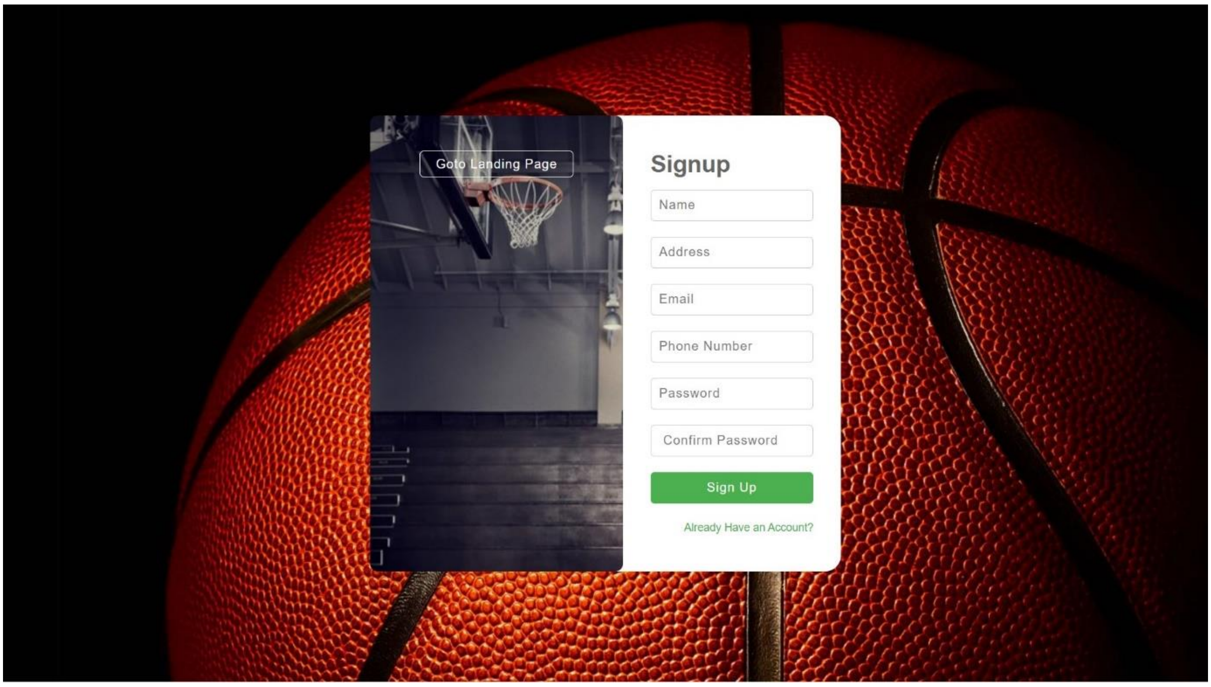
Come And Play Basketball With Us! Our Basketball Court Booking System Offers A Variety Of Courts In Various Locations, So You Can Easily Find And Book The Perfect Court For Your Needs. Whether You're Looking To Play A Casual Pickup Game Or Train With Your Team, Our Platform Makes It Easy To Find The Right Court.

Signup Page

New users can register to this site to enjoy the services provided by JBMS by providing their credentials. Incorrect or empty data doesn't create any accounts. Users can also return to the landing page. They also have option to navigate to login page if they have already registered as shown in figure 22.

Figure 22

Signup page

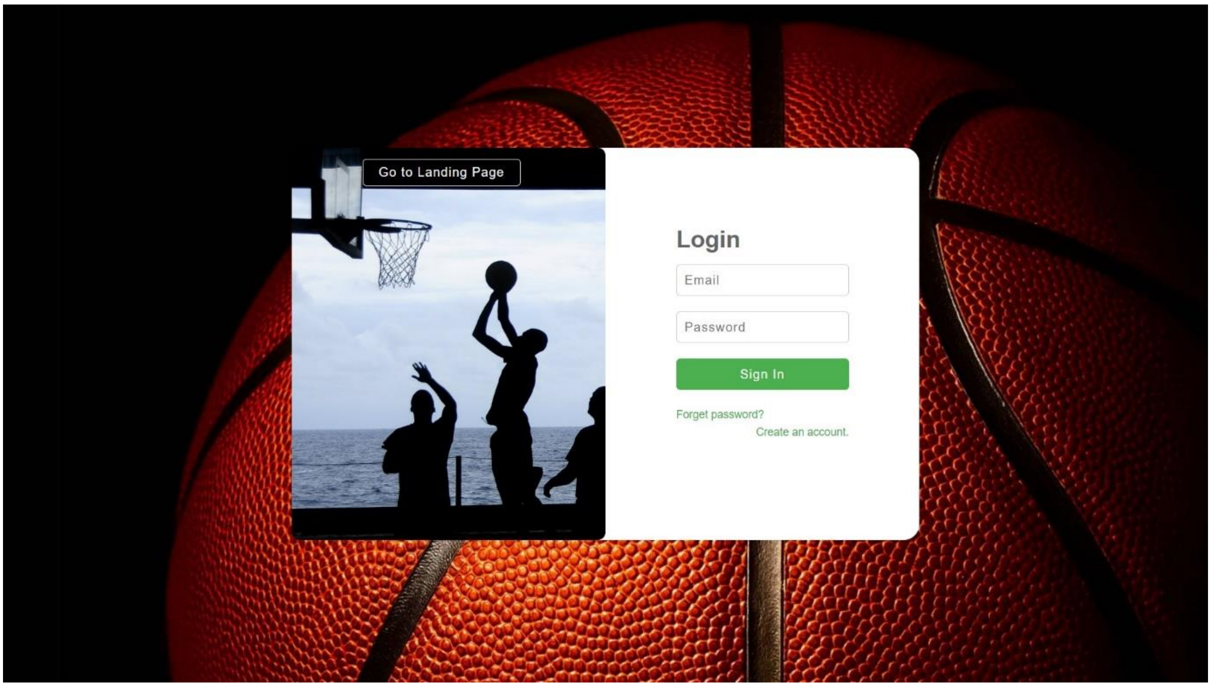


Login Page

Registered players can directly login to their personal dashboard by entering their correct login information on this page. Admin can also login to the admin panel through this page by assigning admin role in database. Users can navigate to landing page, forgot password page or signup page from here too as shown in figure 23.

Figure 23

Login page

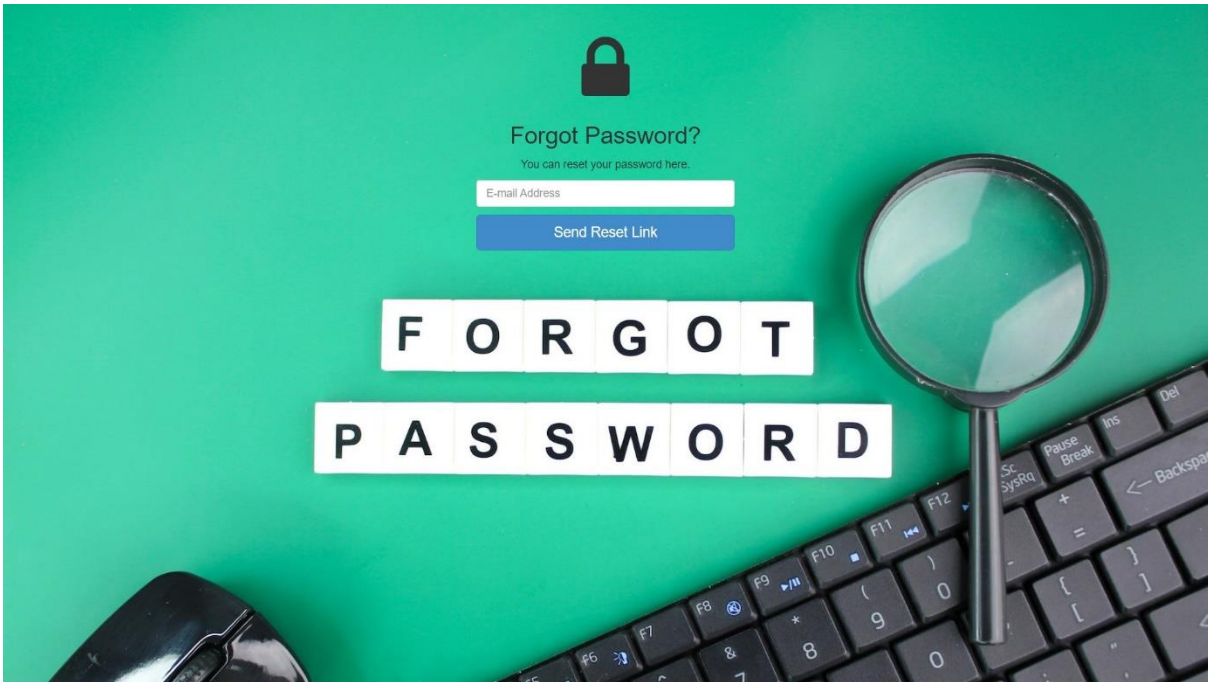


Forgot Password

Patrons who forgot their password can reset password by providing email which sends verification link to the respective email for authentication purpose as shown in figure 24. User can then assign new password.

Figure 24

Forget password page



Forgot Password?

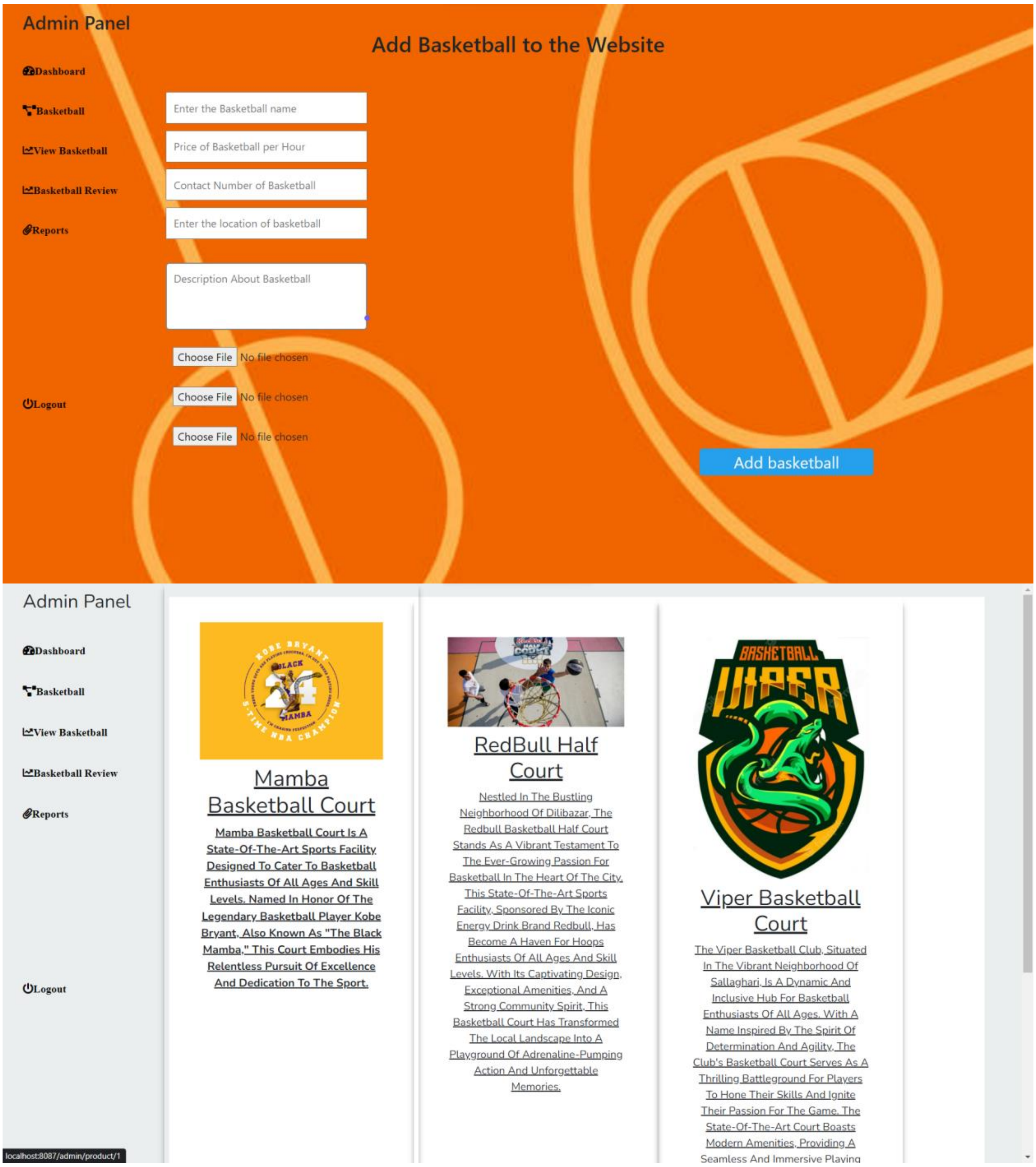
You can reset your password here.

Admin Panel

In this page user can add new courts to the database and manage the bookings from players. Admin can check messages and reviews from players on this page too. For security purpose admin can always logout anytime as per need as in figure 25.

Figure 25

Admin Page


Admin Panel
Dashboard
Basketball
View Basketball
Basketball Review
Reports
Logout
Add Basketball to the Website
Enter the Basketball name
Price of Basketball per Hour
Contact Number of Basketball
Enter the location of basketball
Description About Basketball
Choose File No file chosen
Choose File No file chosen
Choose File No file chosen
Add basketball
Admin Panel
Dashboard
Basketball
View Basketball
Basketball Review
Reports
Logout
localhost:8087/admin/product/1

Mamba Basketball Court

Mamba Basketball Court Is A State-Of-The-Art Sports Facility Designed To Cater To Basketball Enthusiasts Of All Ages And Skill Levels. Named In Honor Of The Legendary Basketball Player Kobe Bryant, Also Known As "The Black Mamba," This Court Embodies His Relentless Pursuit Of Excellence And Dedication To The Sport.

RedBull Half Court

Nestled In The Bustling Neighborhood Of Dilibazar, The Redbull Basketball Half Court Stands As A Vibrant Testament To The Ever-Growing Passion For Basketball In The Heart Of The City. This State-Of-The-Art Sports Facility, Sponsored By The Iconic Energy Drink Brand Redbull, Has Become A Haven For Hoops Enthusiasts Of All Ages And Skill Levels. With Its Captivating Design, Exceptional Amenities, And A Strong Community Spirit, This Basketball Court Has Transformed The Local Landscape Into A Playground Of Adrenaline-Pumping Action And Unforgettable Memories.

Viper Basketball Court

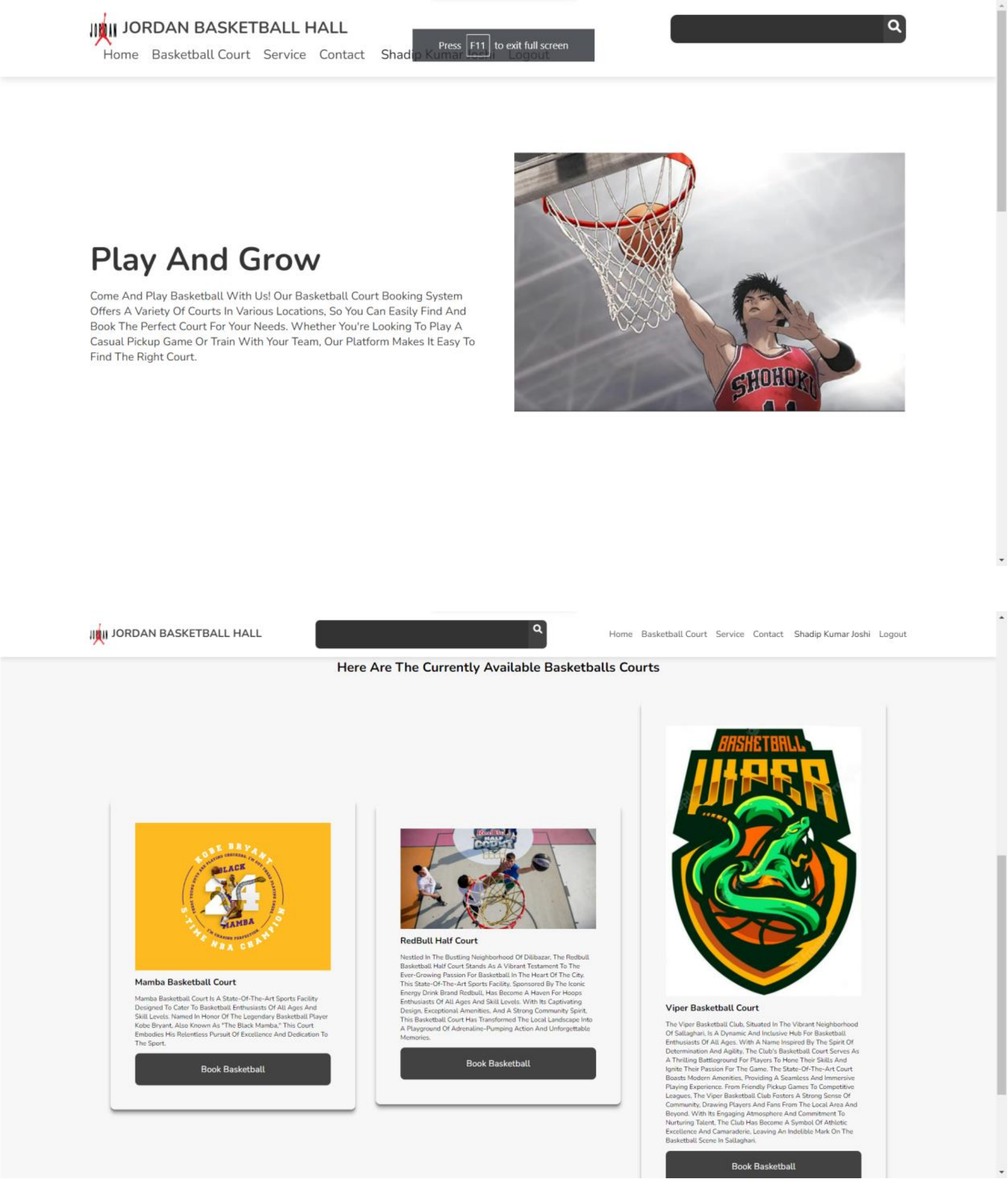
The Viper Basketball Club, Situated In The Vibrant Neighborhood Of Sallaghari, Is A Dynamic And Inclusive Hub For Basketball Enthusiasts Of All Ages. With A Name Inspired By The Spirit Of Determination And Agility, The Club's Basketball Court Serves As A Thrilling Battleground For Players To Hone Their Skills And Ignite Their Passion For The Game. The State-Of-The-Art Court Boasts Modern Amenities, Providing A Seamless And Immersive Playing

User Page

This is the personal page of registered users where users can browse available courts and book them. They have facilities for messaging the management for queries or leaving their feedback as shown in figure 26. Users can update their profile or delete it if needed.

Figure 26

User page



Play And Grow

Come And Play Basketball With Us! Our Basketball Court Booking System Offers A Variety Of Courts In Various Locations, So You Can Easily Find And Book The Perfect Court For Your Needs. Whether You're Looking To Play A Casual Pickup Game Or Train With Your Team, Our Platform Makes It Easy To Find The Right Court.

Here Are The Currently Available Basketballs Courts

Mamba Basketball Court

Mamba Basketball Court Is A State-Of-The-Art Sports Facility Designed To Cater To Basketball Enthusiasts Of All Ages And Skill Levels. Named In Honor Of The Legendary Basketball Player Kobe Bryant, Also Known As "The Black Mamba," This Court Embodies His Relentless Pursuit Of Excellence And Dedication To The Sport.

RedBull Half Court

Nestled In The Bustling Neighborhood Of Dilibazar, The Redbull Basketball Half Court Stands As A Vibrant Testament To The Ever-Growing Passion For Basketball In The Heart Of The City. This State-Of-The-Art Sports Facility, Sponsored By The Iconic Energy Drink Brand Redbull, Has Become A Haven For Hoops Enthusiasts Of All Ages And Skill Levels. With Its Captivating Design, Exceptional Amenities, And A Strong Community Spirit, This Basketball Court Has Transformed The Local Landscape Into A Playground Of Adrenaline-Pumping Action And Unforgettable Memories.

Viper Basketball Court

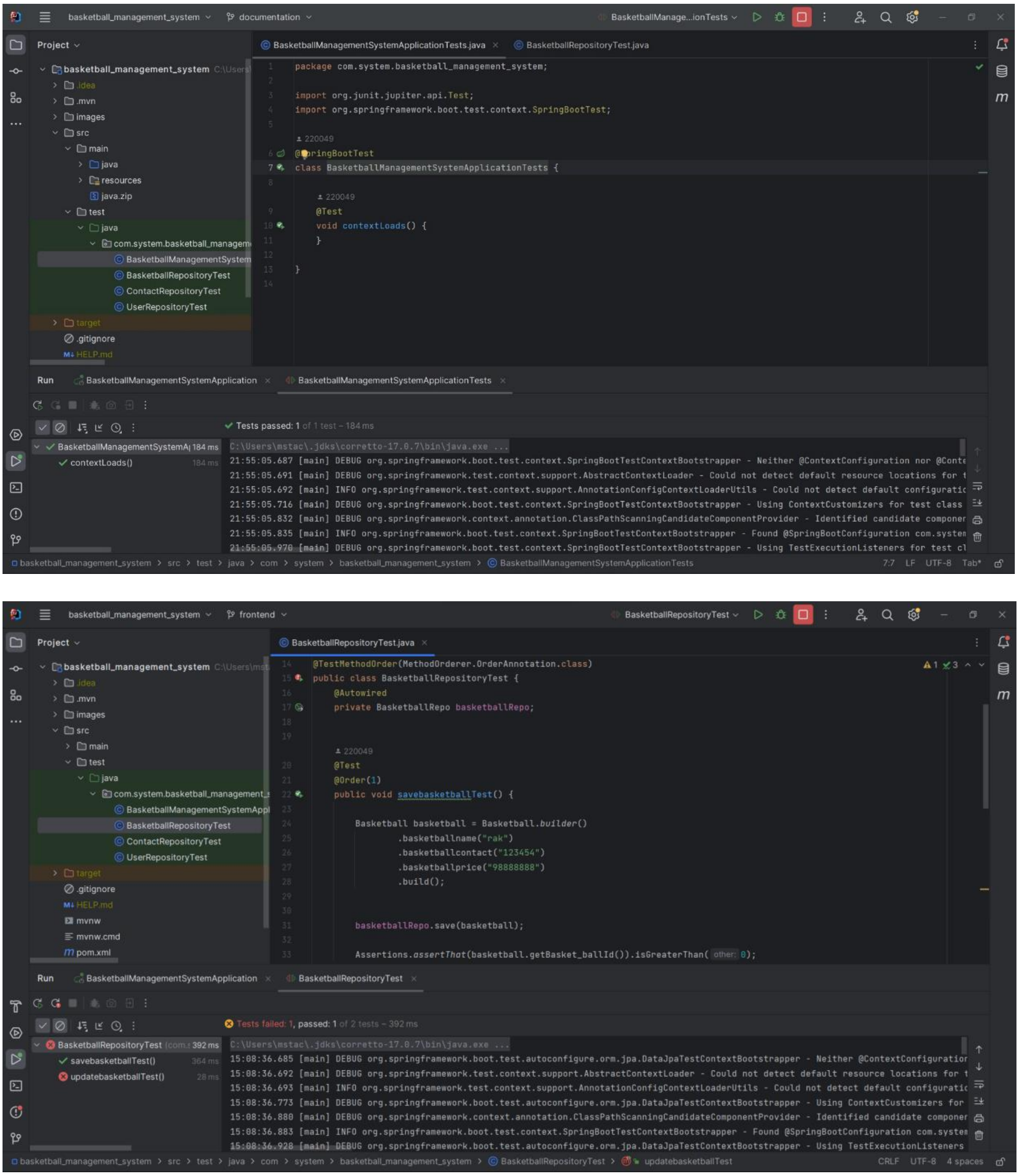
The Viper Basketball Club, Situated In The Vibrant Neighborhood Of Sallaghari, Is A Dynamic And Inclusive Hub For Basketball Enthusiasts Of All Ages. With A Name Inspired By The Spirit Of Determination And Agility, The Club's Basketball Court Serves As A Thrilling Battleground For Players To Hone Their Skills And Ignite Their Passion For The Game. The State-Of-The-Art Court Boasts Modern Amenities, Providing A Seamless And Immersive Playing Experience. From Friendly Pickup Games To Competitive Leagues, The Viper Basketball Club Fosters A Strong Sense Of Community, Drawing Players And Fans From The Local Area And Beyond. With Its Engaging Atmosphere And Commitment To Nurturing Talent, The Club Has Become A Symbol Of Athletic Excellence And Camaraderie, Leaving An Indelible Mark On The Basketball Scene In Sallaghari.

Testing

The system testing was performed through BDD after software development to check for functionality of the project as shown in figure 27 and 28. JBMS was also black-box-tested where its outputs were recorded. Errors were mended until all test cases were fulfillef.

Figure 27

BDD tests of JBMS


basketball_management_system
documentation
BasketballManage...ionTests
Project
BasketballManagementSystemApplicationTests.java
BasketballRepositoryTest.java
basketball_management_system
.idea
.mvn
images
src
main
java
resources
java.zip
test
java
com.system.basketball_managem
BasketballManagementSystem
BasketballRepositoryTest
ContactRepositoryTest
UserRepositoryTest
target
.gitignore
HELP.md
Run
BasketballManagementSystemApplication
BasketballManagementSystemApplicationTests
basketball_management_system > src > test > java > com > system > basketball_management_system > BasketballManagementSystemApplicationTests
7:7 LF UTF-8 Tab*

```
package com.system.basketball_management_system;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class BasketballManagementSystemApplicationTests {

    @Test
    void contextLoads() {
    }

}
```

✓ Tests passed: 1 of 1 test – 184 ms

✓ BasketballManagementSystemA| 184 ms
✓ contextLoads() 184 ms

```
C:\Users\mstac\.jdks\corretto-17.0.7\bin\java.exe ...
21:55:05.687 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Neither @ContextConfiguration nor @Conte
21:55:05.691 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Could not detect default resource locations for t
21:55:05.692 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuratic
21:55:05.716 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Using ContextCustomizers for test class
21:55:05.832 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate componer
21:55:05.835 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootConfiguration com.systen
21:55:05.970 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Using TestExecutionListeners for test cl
```

basketball_management_system
frontend
BasketballRepositoryTest
Project
BasketballRepositoryTest.java
basketball_management_system
.idea
.mvn
images
src
main
test
java
com.system.basketball_management_s
BasketballManagementSystemAppl
BasketballRepositoryTest
ContactRepositoryTest
UserRepositoryTest
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
Run
BasketballManagementSystemApplication
BasketballRepositoryTest
basketball_management_system > src > test > java > com > system > basketball_management_system > BasketballRepositoryTest > updatebasketballTest
CRLF UTF-8 4 spaces

```
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class BasketballRepositoryTest {
    @Autowired
    private BasketballRepo basketballRepo;


    @Test
    @Order(1)
    public void savebasketballTest() {

        Basketball basketball = Basketball.builder()
                .basketballname("rak")
                .basketballcontact("123454")
                .basketballprice("98888888")
                .build();


        basketballRepo.save(basketball);

        Assertions.assertThat(basketball.getBasket_ballId()).isGreaterThan(0);
```

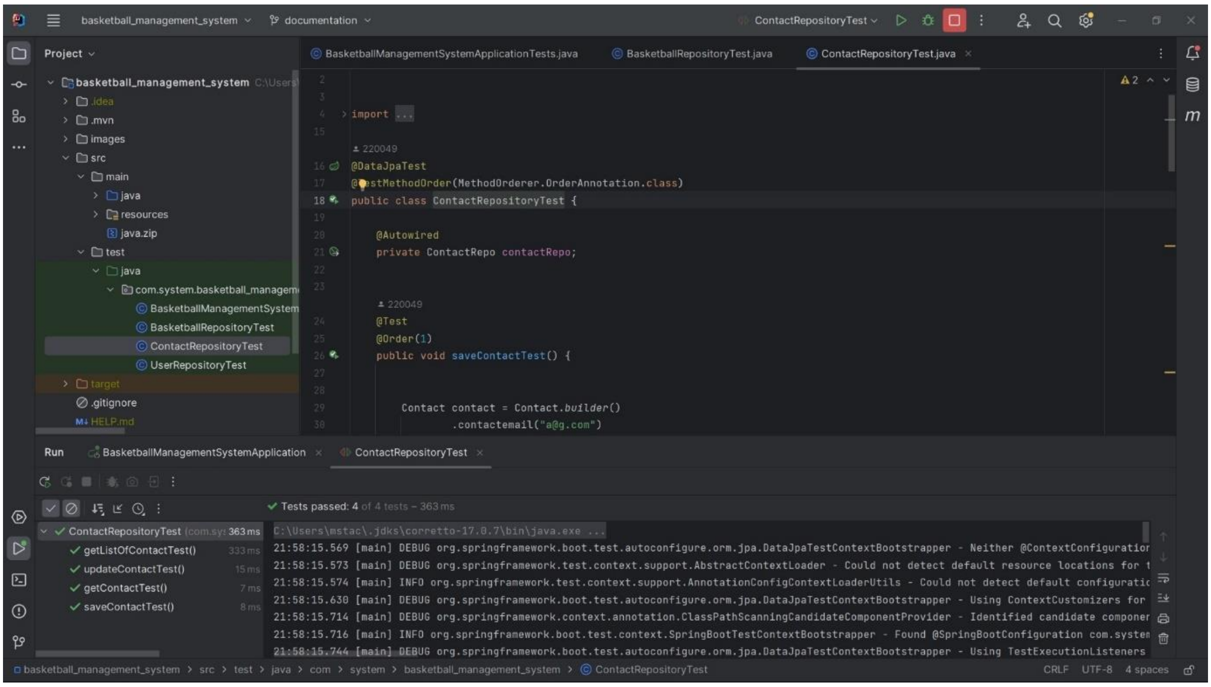
Tests failed: 1, passed: 1 of 2 tests – 392 ms

BasketballRepositoryTest (com.s 392 ms
✓ savebasketballTest() 364 ms
updatebasketballTest() 28 ms

```
C:\Users\mstac\.jdks\corretto-17.0.7\bin\java.exe ...
15:08:36.685 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Neither @ContextConfiguratior
15:08:36.692 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Could not detect default resource locations for t
15:08:36.693 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuratic
15:08:36.773 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using ContextCustomizers for
15:08:36.880 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate componer
15:08:36.883 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootConfiguration com.systen
15:08:36.928 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using TestExecutionListeners
```

Figure 28

BDD tests of JBMS



basketball_management_system
documentation
ContactRepositoryTest
Project
BasketballManagementSystemApplicationTests.java
BasketballRepositoryTest.java
ContactRepositoryTest.java
basketball_management_system C:\Users\
.idea
.mvn
images
src
main
java
resources
java.zip
test
java
com.system.basketball_managem
BasketballManagementSystem
BasketballRepositoryTest
ContactRepositoryTest
UserRepositoryTest
target
.gitignore
HELP.md
Run
BasketballManagementSystemApplication
ContactRepositoryTest
basketball_management_system > src > test > java > com > system > basketball_management_system > ContactRepositoryTest
CRLF UTF-8 4 spaces

```java
import ...

220049
@DataJpaTest
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class ContactRepositoryTest {

    @Autowired
    private ContactRepo contactRepo;


    220049
    @Test
    @Order(1)
    public void saveContactTest() {


        Contact contact = Contact.builder()
                .contactemail("a@g.com")
```

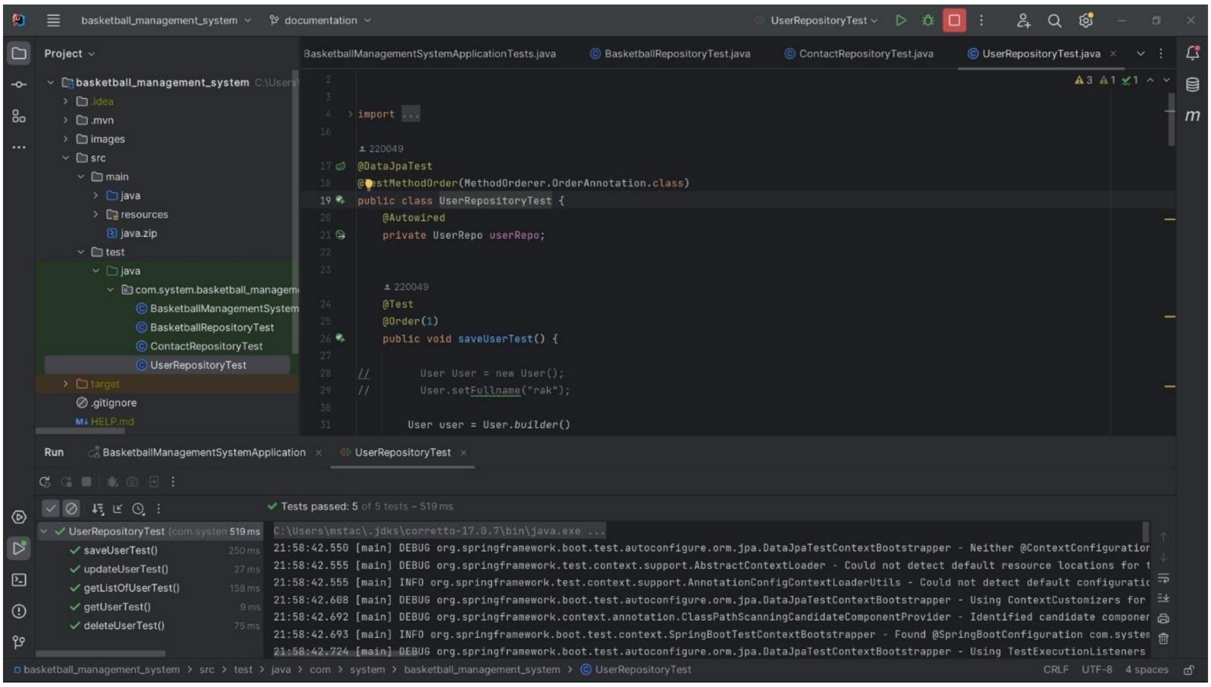
```
✓ Tests passed: 4 of 4 tests – 363 ms

✓ ContactRepositoryTest (com.sys 363 ms
  ✓ getListOfContactTest()   333 ms
  ✓ updateContactTest()       15 ms
  ✓ getContactTest()           7 ms
  ✓ saveContactTest()          8 ms

C:\Users\mstac\.jdks\corretto-17.0.7\bin\java.exe ...
21:58:15.569 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Neither @ContextConfiguratior
21:58:15.573 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Could not detect default resource locations for t
21:58:15.574 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuratic
21:58:15.630 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using ContextCustomizers for
21:58:15.714 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate componer
21:58:15.716 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootConfiguration com.systen
21:58:15.744 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using TestExecutionListeners
```

Figure 29

BDD tests of JBMS



basketball_management_system
documentation
UserRepositoryTest
Project
BasketballManagementSystemApplicationTests.java
BasketballRepositoryTest.java
ContactRepositoryTest.java
UserRepositoryTest.java
basketball_management_system C:\Users\
.idea
.mvn
images
src
main
java
resources
java.zip
test
java
com.system.basketball_managem
BasketballManagementSystem
BasketballRepositoryTest
ContactRepositoryTest
UserRepositoryTest
target
.gitignore
HELP.md
Run
BasketballManagementSystemApplication
UserRepositoryTest
basketball_management_system > src > test > java > com > system > basketball_management_system > UserRepositoryTest
CRLF UTF-8 4 spaces

```java
import ...

220049
@DataJpaTest
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class UserRepositoryTest {
    @Autowired
    private UserRepo userRepo;


    220049
    @Test
    @Order(1)
    public void saveUserTest() {

//        User User = new User();
//        User.setFullname("rak");

      User user = User.builder()
```

```
✓ Tests passed: 5 of 5 tests – 519 ms

✓ UserRepositoryTest (com.systen 519 ms
  ✓ saveUserTest()          250 ms
  ✓ updateUserTest()         27 ms
  ✓ getListOfUserTest()     158 ms
  ✓ getUserTest()             9 ms
  ✓ deleteUserTest()         75 ms

C:\Users\mstac\.jdks\corretto-17.0.7\bin\java.exe ...
21:58:42.550 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Neither @ContextConfiguratior
21:58:42.555 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Could not detect default resource locations for t
21:58:42.555 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuratic
21:58:42.608 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using ContextCustomizers for
21:58:42.692 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate componer
21:58:42.693 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootConfiguration com.systen
21:58:42.724 [main] DEBUG org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTestContextBootstrapper - Using TestExecutionListeners
```

Project Issue

In Development Phase

Being my first ever web project, I faced a lot of hurdles mainly in backend. Database was easy but attempting to follow the guidelines of SOLID principles, lean framework, along with mvc architecture with many directories like pojo, services, repository, config, security, etch were mind-boggling especially their interrelation. The frontend part was also a headache to work hand-in-hand with different languages like javascript, html, css with bootstrap to make a responsive and user-friendly UI. A lot of errors were made during development causing lots of errors and bugs, which required a lot of time to rectify them using BDD testing and blackbox testing.

Project Limitation

JBMS was intended to include a huge array of functionalities. However, many features couldn't be successfully included in it because of lack of time and expertise. Google maps don't properly show the location of the courts, neither does it support live location of users. This software still lacks proper back-to-back messaging features. Notification system couldn't be integrated for mobile devices. Online payment system couldn't be integrated in the first version of this project. The greatest drawback of this software at present is its inability to check automatically if more bookings are made at the same time and response to court booking automatically.

Future Plans

All the limitations of this project are planned to be solved in later upgrades. Live location of users and court locations will be focused on future versions. Automatic booking for empty appointment will be given priority while overlapping bookings will be informed to users on the spot of booking with suggestions for available time of courts. Instant chats will also be introduced in the future revisions with better and streamlined user interface. Many new convenient features like score tracking of matches, players game history, and analytics of their gameplay are planned to be introduced in this project for reviewing purposes. JBMS will also have video upload function for gameplays to be analysed by coaches for training purpose. API for online payment will be introduced in this program soon for advance/full payment for booking purpose.

What I learned?

Firstly, I earned practical experience in the use of all the new frontend languages like javascript, html, css along with bootstrap framework to build a user friendly, responsive and attractive web interface. The use of relational DBMS called postgres helped me to structure my database through tables and relationships to plan my project better, thus utilizing the conceptual diagrams like erd properly. I enjoyed utilising my java language with maven to setup, configure and run an easy and fast web application through springboot. The use of frontend, database and backend aided me to foster my knowledge in MVC architecture and 3-tier database architecture for efficient program development and integration.

Additionally, precious experience of developing a web application from scratch increased my experience in the IT field. Waterfall SDLC made me to plan and work sequentially to complete my project, which I can use in my daily life too. I learned to learn and grow by facing new challenges whenever I faced any kind of hurdles. Friendly interaction with the instructor and programmers in online platforms reminded me of the importance of friendly social networks.

Finally, I learned the usage of SOLID and lean principles to make a software more compact and efficient for understanding my codes and its management. All the theoretical knowledges I earned in my modules like Mathematics, Software design, Programming and Algorithm, Database, Architecture, Object-oriented Programming and DSA were fully utilised in this project.

Conclusion

A functional product was developed in alignment with the project's aim and objectives by incorporating the theoretical concepts acquired during classroom modules and leveraging all available technologies to their maximum potential.

This project successfully streamlines the interaction between basketball court owners and players during the booking process. The software fulfills both functional and non-functional requirements as specified. System testing has yielded positive outcomes, and any bugs identified during implementation were effectively resolved through strong teamwork. The graphical user interfaces (GUIs) are accessible through secure credentials, ensuring user privacy. The team's collaborative efforts resulted in attractive, presentable, and user-friendly GUIs that are self-explanatory and easy to navigate.

The project was built compiling to the IT ethics and it still has the option for further upgrades to mitigate its limitations.

Reference

Abeythilake, U. (2022, June 12). *Agile Methodology*. Medium. Retrieved December 11, 2022, from https://medium.com/@abeythilakeudara3/agile-methodology-106270809c99

Agile Alliance. (2021, October 9). *Agile Manifesto for Software Development*. Agile Alliance |. Retrieved December 13, 2022, from https://www.agilealliance.org/manifesto-download/

AWS Whitepaper. (n.d.). *Three-tier architecture overview*. Amazon AWS. Retrieved December 11, 2022, from https://docs.aws.amazon.com/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/three-tier-architecture-overview.html

Company, I.-. M. a. D., Trętowicz, M., Zalewski, A., & Turkowski, J. (2023, January 26). MVC MVP and MVVM architecture pattern – Introduction. *Mobile & Web App Development Company | USA, UK, Norway*. Retrieved July 10, 2023, from https://itcraftapps.com/blog/mvc-mvp-and-mvvm-architecture-pattern-introduction/#:~:text=In%20MVP%2C%20the%20data%20from,which%20can%20support%20multiple%20Views.

Enalean SAS. (n.d.). *Understanding Agile Scrum in 10 minutes*. Tuleap. Retrieved December 23, 2022, from https://www.tuleap.org/agile/agile-scrum-in-10-minutes

Global App Testing. (n.d.). *Best Practices for Agile Testing*. Global App Testing Retrieved December 13, 2022, from https://www.globalapptesting.com/the-ultimate-guide-to-agile-testing

Hoory, L., Bottorff, C., & Watts, R. (2022, March 27). *What Is A Burndown Chart? Everything You Need To Know*. Forbes Advisor. Retrieved December 13, 2022, from https://www.forbes.com/advisor/business/what-is-a-burndown-chart/

IBM. (n.d.). *What is three-tier architecture?* IBM. Retrieved December 24, 2022, from https://www.ibm.com/topics/three-tier-architecture

Lima, A. (2020). MVP vs MVVM: principais diferenças, vantagens e desvantagens. *Zup*. https://www.zup.com.br/blog/mvp-vs-mvvm

Nitika. (2018). *WHAT IS THE ROLE OF WATERFALL MODEL IN SDLC - Loginworks*. Loginworks. Retrieved 21 July 2023, from https://www.loginworks.com/blogs/role-waterfall-model-sdlc/.

Oloruntoba, S. (2021). SOLID: The First 5 Principles of Object Oriented Design. *DigitalOcean*. Retrieved July 20, 2023, from https://www.digitalocean.com/community/conceptual-articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design

Pm admin. (2021, June 13). *The Agile Journey: A Scrum overview*. PM Partners. Retrieved December 11, 2022, from https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/

Shoaib A. (n.d.). *You will get Hostel Management System (A complete Web Solution)*. https://www.upwork.com. Retrieved July 9, 2023, from https://www.upwork.com/services/product/development-it-hostel-management-system-a-complete-web-solution-1666480411133214720

Simplilearn. (2022, November 9). *A Complete Guide to Prototyping In Design Thinking With Benefits & Examples*. Simplilearn.com. Retrieved November 7, 2022, from https://www.simplilearn.com/prototyping-in-design-thinking-article

Simplilearn. (2023, March 7). Fundamentals of lean methodology explained with examples. *Simplilearn.com*. Retrieved July 21, 2023, from https://www.simplilearn.com/lean-methodology-article#:~:text=Lean%20is%20a%20continuous%20improvement,to%20customers%20with%20fewer%20resources

Singh Bhui, J. (2022, October 6). *Understand MVC Architecture in 5 mins*. Crio. Retrieved July 21, 2023, from https://www.crio.do/blog/understand-mvc-architecture/

Singhal, P. (2021). *Software Engineering | Iterative Waterfall Model - GeeksforGeeks*. GeeksforGeeks. Retrieved 21 July 2023, from https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/.

Appendix

Version Control

Github Link: https://github.com/220049/JORDAN-Basketball-Management-System

Presentation

Youtube Video Link: https://youtu.be/y0c1B1yD38c

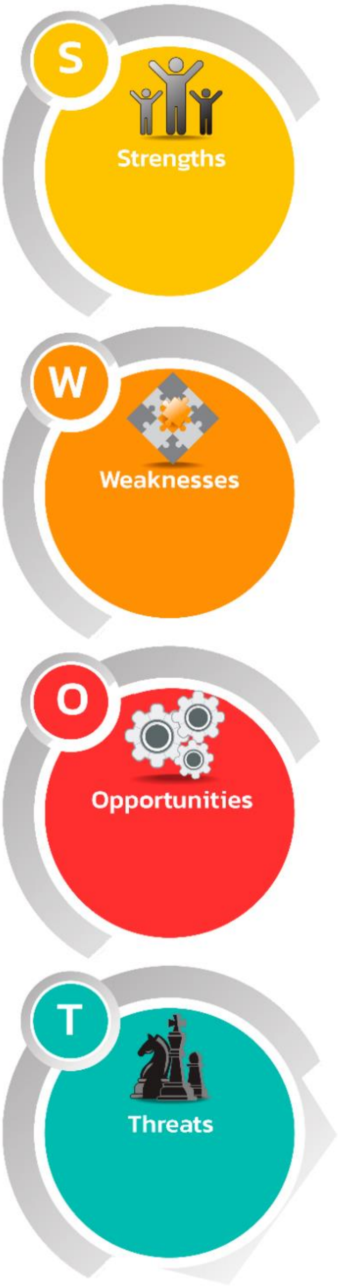
Spreadsheet Link:

https://docs.google.com/spreadsheets/d/14w1F8OdkbJTEbe7ATguLa30gw1ah1043nPeJeMGX6xo/edit#gid=1198600336

SWOT Analysis

Figure 30

Swot Analysis



- User-Friendly Platform
- Real-Time Court Availability
- Data Analytics
- Data Analytics
- Structured Review and Feedback System

- Limited Reach
- Initial Setup and Adoption
- Potential Technical Glitches

- Expansion to New Locations
- Partnerships with Basketball Organizations
- Mobile App Development
- Integration of Payment Solutions

- Competition from Similar Platforms
- Changing User Preferences
- Data Security Concerns
- Economic and External Factors