

**LA GRANDEE INTERNATIONAL COLLEGE**

**Simalchaur, Pokhara Nepal**

Final Project Report

On

**“JobNexus”**

(Online Job Portal)

**Submitted to:**

Bachelor of Computer Application (BCA) Program

In partial fulfilment of the requirements for the degree of BCA under

Pokhara University

**Submitted by:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | **Course** | **Semester** | **P.U. Registration Number** |
| Akriti Chapagain | BCA | 8th | 2019-01-53-0104 |
| Binaya Koirala | BCA | 8th | 2019-01-53-0111 |

**Date:28/05/2024**

**ACKNOWLEDGEMENT**

We would like to express our gratitude to Pokhara University for developing a highly valuable curriculum and enabling us to undertake this project. Furthermore, we extend our sincere appreciation to La Grandee International College, whose assistance has been instrumental in making our project a reality. Without their support, our project would have been unattainable. We express our sincere regard to our project supervisor Sunil Sapkota, for his valuable time, guidance, encouragement, support, and cooperation throughout the duration of our project. We would sincerely like to thank BCA Department for giving us the opportunity to work on enhancing our technical skills while undergoing this project. This project helped us understand the various parameters involved in the development of a mobile application and the working and integration of frontend along with the backend to create a fully functional mobile application. We would like to express our gratitude to Pokhara University for developing a highly valuable curriculum and enabling us to undertake this project. Furthermore, we extend our sincere appreciation to La Grandee International College, whose assistance has been instrumental in making our project a reality. Without their support, our project would have been unattainable.

With Regards,

Akriti Chapgain (Registration No: 2019-1-53-0104)

Binaya koirala (Registration No: 2019-1-53-0111)

ii

**DECLARATION FOR**

**“JobNexus”**

**STUDENT’S DECLARATION**

We hereby declare that we are the only authors of this work and that no sources other than the mentioned here have been used in this. We assure you that the work we present here is unique to ourselves and resemblances to another similar project are purely coincidental.

Akriti Chapagai (PU Exam Roll no): 20530162

Binaya Koirala (PU Exam Roll no): 20530169

Program: BCA, 8th Semester

Date:

iii

**Supervisor’s Declaration**

I hereby recommend that this project entitled "JobNexus " is done under my supervision by Akriti Chapagain, Binaya Koirala during 8th semester in partial fulfillment of the requirement for the degree of Bachelor of Computer Application (BCA) under Pokhara University is completed to my satisfaction and be processed for final evaluation.

Sunil Sapkota



iv

**Letter of Approval**

We certify that we have examined this report JobNexus ", and are satisfied with the proposal defense. In our opinion it is satisfactory in the scope and qualify as project in partial fulfillment of the requirements for the degree of BCA under Pokhara University.

**Project Supervisor**

**External Invigilator**

**Principal**

**Er. Sunil Sapkota**

**Er. Kiran K.C**

v

**Abstract**

This report is submitted to the Department of Computer Application at LA Grandee International College as part of the BCA 8th Semester Project. The project, titled "JobNexus”, it serves as an introductory overview of the Job Portal App project, showcasing its potential to revolutionize the event planning and venue selection process. “Job Nexus” is being developing for creating an interactive job vacancy form for candidates the app effectively gathers client and candidate information, categorizing them based on job requirements and vacancies. Its unique development methodology helps in acquiring the client and candidate information and separating them according to the job requirements and vacancies. This application would provide details of the job. An employer being registered in the application has the facility to use the services. Being an authorized user he can .Publish vacancy details and can search no of Employees on portal and also he can search candidates on basis of the key skill which employee provides on registration.

.

vi

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Contents** |  |
| [1.](#page11) | [Introduction ............................................................................................................](#page11) | | [1](#page11) |
| [2.](#page12) | [Problem Statement .....................................................................................................](#page12) | | [2](#page12) |
| [3.](#page13) | [Objectives ..................................................................................................................](#page13) | | [3](#page13) |
| [4.Background Study ......................................................................................................](#page14) | | | [4](#page14) |
| [5.](#page16) | [Methodology ..............................................................................................................](#page16) | | [6](#page16) |
|  | [5.1](#page17) | [Requirement Analysis ..........................................................................................](#page17) | [7](#page17) |
|  | [5.2. Problem Analysis: ...............................................................................................](#page17) | | [7](#page17) |
|  | [5.3 Verification and Validation: ..................................................................................](#page18) | | [8](#page18) |
|  | [5.4](#page18) | [Development Plan: ...............................................................................................](#page18) | [8](#page18) |
|  | [5.5](#page18) | [Testing: ................................................................................................................](#page18) | [8](#page18) |
|  | [5.6](#page18) | [Documentation: ...................................................................................................](#page18) | [8](#page18) |
| [6.](#page20) | [Requirement document ............................................................................................](#page20) | | [10](#page20) |
|  | [6.1. Tools and software requirements ......................................................................](#page20) | | [10](#page20) |
|  | [6.2. Functional Requirements ..................................................................................](#page20) | | [10](#page20) |
|  | [6.3](#page21) | [Requirement Matrix: ..........................................................................................](#page21) | [11](#page21) |
| [7.](#page21) | [System Analysis and Design: ..................................................................................](#page21) | | [12](#page21) |
|  | [7.1](#page22) | [ER-Diagram .......................................................................................................](#page22) | [12](#page22) |
|  | [7.2](#page23) | [Class-Diagram ....................................................................................................](#page23) | [13](#page23) |
|  | [7.3](#page24) | [DFD (Data Flow Diagram) ................................................................................](#page24) | [14](#page24) |
| [8.](#page26) | [Development ............................................................................................................](#page26) | | [16](#page26) |
| [9. TESTING .................................................................................................................](#page29) | | | [19](#page29) |
|  | [9.1. Test cases ...........................................................................................................](#page29) | | [19](#page29) |
| [10. Time schedule ........................................................................................................](#page42) | | | [32](#page42) |
| [11. Project Results: .....................................................................................................](#page43) | | | [33](#page43) |
| [12.Future Enhancement ...............................................................................................](#page45) | | | [35](#page45) |
| [13.Conclusion ..............................................................................................................](#page46) | | | [36](#page46) |
| [14. References .............................................................................................................](#page47) | | | [37](#page47) |
| [Annexure .....................................................................................................................](#page49) | | | [39](#page49) |

vii

**List of Figures**

[FIGURE 1: ITERATION MODEL](file:///G:/Theses%202080%20Computer%201/Samjhana/print%20garni%20wala.docx%23_Toc144814727) [6](file:///G:/Theses%202080%20Computer%201/Samjhana/print%20garni%20wala.docx%23_Toc144814727)

[FIGURE 2:ER-DIAGRAM](#page22) [12](#page22)

[FIGURE 3 CLASS-DIAGRAM](#page23) [13](#page23)

[FIGURE 4: DFD LEVEL-0](#page24) [14](#page24)

[FIGURE 5: DFD LEVEL 1](#page24) [14](#page24)

[FIGURE 6: DFD LEVEL 2](#page25) [15](#page25)

[FIGURE 7: DATABASE SCHEME](#page27) [17](#page27)

[FIGURE 8 GANTT CHART](#page42) [32](#page42)

viii

**List of Tables**

[Table 1: Requirement Matrix](#page21) [11](#page21)

[Table 2: Task Division](#page28) [18](#page28)

[Table 3: Test case for verification of user login](#page31) [21](#page31)

[Table 4 : Test case for verification of admin login](#page34) [24](#page34)

[Table 5: Test case for verification of venue owner login](#page36) [26](#page36)

[Table 6: Test case for verification of registration](#page39) [29](#page39)

[Table 7: Test case for booking venue](#page40) [30](#page40)

[Table 8: Test case for payment](#page41) [31](#page41)

ix

**List of abbreviation**

DFD- Data Flow Diagram

MySQL- My Structured Query Language

ER- Entity Relationship

x

# INTRODUCTION

Job Nexus, a mobile application, is set to revolutionize the job search process in Nepal. Job Nexus establishes partnerships with local organizations to facilitate the provision of vacancies for semi-skilled manpower, as well as opportunities for individuals seeking entry-level positions. This strategic collaboration ensures that Job Nexus offers opportunities to a diverse range of job seekers, including those with varying levels of skills and experience. Leveraging these partnerships, Job Nexus broadens job access and provides opportunities for individuals in rural regions and local communities

The job market in Nepal, much like in many developing countries, is characterized by a significant gap between job seekers and job providers. Many job seekers, particularly those from rural areas and those with minimal skills, face difficulties in accessing job opportunities. On the other hand, employers often struggle to find suitable candidates for their vacancies. The rise of technology and mobile applications presents an opportunity to bridge this gap effectively.

By offering a platform that connects job seekers with a wide array of job opportunities, including roles that require minimal skills, Job Nexus plays a crucial role in promoting inclusivity and economic empowerment. Job Nexus simplifies the job search process for both candidates and employers. Job seekers have the ability to easily upload their resumes and explore job openings matched to their profiles, while employers benefit from simplified job posting and application management. Furthermore, Job Nexus is committed to strengthening skill development and professional growth. By creating a unified platform for training programs and volunteer opportunities, Job Nexus empowers individuals to enhance their skills and access new career opportunities.

This application was developed using technologies such as React Native for frontend development and Laravel for backend, along with various tools and utilities to ensure easy and flawless navigation. This enabled Job Nexus to serve as a beacon for candidates searching for job opportunities and an asset for those who benefited from this application.

1

**2. Problem Statement**

Following were the problems we found in job portal applications:.

* Job seekers encounter difficulties finding appropriate roles because of insufficient skill-based matching on existing platforms.
* Semiskilled workers, like plumbers and construction workers, confront a scarcity of job listings as portals primarily prioritize skilled positions.
* In the absence of dedicated platforms, there's a notable gap for registering volunteers during events, leaving organizers without efficient means to manage manpower.
* There aren't many apps that bring together different training and workshop programs, making it hard for people to find skill-building opportunities.
* Current job portals in Nepal lack collaborations with local businesses, hindering job access, particularly in rural regions

2

# OBJECTIVES

* To establish collaborations with local businesses to broaden job access and ensure rural inclusion and semi-skilled job potentials.
* To implement advance filter to better match job seekers with suitable roles based on their skills.
* To create a unified platform for users to easily find and enroll in diverse training programs, volunteers, fostering skill development.

3

**4.Background Study**

Nepal’s job market is diverse, encompassing a range of sectors including agriculture, manufacturing, services, and more recently, technology and tourism. Despite this diversity, the market faces several challenges:

1. **High Unemployment Rate**: According to the Central Bureau of Statistics Nepal, the unemployment rate has been fluctuating, with youth unemployment particularly high. Many graduates are unable to find jobs that match their qualifications.
2. **Underemployment**: A significant portion of the workforce is underemployed, meaning they are working in jobs that do not fully utilize their skills or provide adequate income.
3. **Rural-Urban Divide:** There is a stark contrast between job opportunities in urban centers and rural areas. Many rural job seekers migrate to cities in search of better opportunities, leading to urban overcrowding and rural depopulation.
4. **Lack of Access to Information:** Job seekers often lack access to timely and relevant information about job openings, especially those in rural areas. Employers also face challenges in reaching a broad pool of candidates.

**Comparative Analysis of Existing Job Portals**

Several job portals are currently operating in Nepal, each with its own strengths and limitations.

1. **Merojob**

**Strengths:**

* One of the largest and most popular job portals in Nepal.
* Extensive database of job listings across various industries.
* User-friendly interface with advanced search options.

**Limitations:**

* Primarily focused on urban areas and skilled positions.
* Limited reach in rural areas and among semi-skilled workers.

1. **Kantipur Job**

**Strengths:**

* Wide range of job listings from various sectors.
* Strong brand recognition due to association with Kantipur Media Group.

**Limitations:**

* User interface can be cumbersome and outdated.
* Focuses more on white-collar jobs, with less emphasis on semi-skilled and entry-level positions.

1. **RamroJob**

**Strengths:**

* Extensive job listings and user-friendly interface.
* Good categorization of jobs by industry and job type.

**Limitations:**

* Limited outreach and engagement with rural job markets.
* Less emphasis on skill development and training opportunities.

1. **JobsNepal**

**Strengths:**

* Long-standing presence in the Nepali job market.
* Wide variety of job postings, including some from international companies.

**Limitations:**

* Interface and user experience are not as modern or intuitive.
* Primarily targets skilled and professional job seekers.

1. **Mero Rojgari**

**Strengths:**

* Focuses on job opportunities for semi-skilled and blue-collar workers.
* Accessible and straightforward for users with minimal technical skills.

**Limitations:**

* Smaller database of job listings compared to larger portals.
* Limited features for employers and job seekers..

**Need for Job Nexus**

Given the current landscape, there is a clear need for a platform like Job Nexus which:

1. **Focuses on Inclusivity:** By targeting semi-skilled and entry-level positions and ensuring accessibility for job seekers from rural areas.
2. **Facilitates Partnerships:** By collaborating with local organizations to source job openings and training opportunities.
3. **Utilizes Technology**: By employing modern technologies such as React Native and Laravel to create a seamless user experience for both job seekers and employers.
4. **Empowers Users:** By offering not just job listings but also training programs, volunteer opportunities, and career development resources to enhance skills and employability.

By conducting a comprehensive background study, we are prepared to develop a venue management app that meets the needs of your target audience and stands out in the market.

5

.





**6. Requirement document**

**6.1. Tools and software requirements**

Integrated Development Environment (IDE):

Frontend Development: Visual Studio Code (VS Code) - A lightweight and versatile code editor with robust support for JavaScript and React Native development.

Backend Development: Visual Studio Code (VS Code) - Equipped with extensions for PHP development, allowing seamless integration with Laravel framework.

Version Control System: GitHub - A widely-used platform for version control and collaboration, facilitating code sharing, reviewing, and tracking changes.

Package Manager: npm (Node Package Manager) - The default package manager for JavaScript runtime environment Node.js, essential for managing dependencies and packages in React Native projects.

Integrated Development Environment (IDE): Several IDE options are available for React native app development. Popular choices include: Visual Studio Code (VS Code): A lightweight and free IDE with excellent React native support through extensions. Android Studio: A full-featured IDE developed by Google, which includes built-in support for react native.

State management: Effective state management is vital for a smooth ecommerce experience. Flutter offers various options like Provider, River pod, Redux, or Mob for handling app state

Deployment and Hosting Tools:

Expo: A powerful toolchain and platform for React Native development, providing features such as rapid prototyping, debugging, and over-the-air updates. It simplifies the development process and offers easy deployment to mobile devices.

Composer: Dependency manager for PHP, primarily used for managing dependencies in Laravel projects. It simplifies package installation and version management in PHP applications..

**6.2. Functional Requirements**

* User Registration and Authentication: - Enable users to register and create accounts. - Implement secure authentication methods like email/password, social media login, or OAuth. - Provide password reset and account recovery option

6.2. Functional Requirements

* User Registration and Authentication:

Users should be able to sign up and log in using email, phone number, or social media accounts through Firebase Authentication.

Password reset functionality via email should be implemented using Firebase Authentication services.

* User Profiles:

Job seekers and employers should have profiles where they can manage personal information, preferences, and settings.

User profiles should be linked to Firebase Authentication for authentication and security purposes.

* Job Listings and Applications:

Employers should be able to post job listings with details such as job title, description, requirements, location, and salary range.

Job seekers should be able to browse, search, and filter job listings based on various criteria.

Job seekers should be able to submit job applications, including uploading resumes and cover letters.

* Notifications:

Push notifications should be implemented using Expo Notifications to notify users about new job listings, application status updates, and other relevant events.

* Deployment:

The application should be deployed and distributed using Expo, enabling easy deployment to both iOS and Android devices for testing and production purposes.

* Integration:

Integration with GitHub should be set up for version control, collaboration, and automated CI/CD workflows using GitHub Actions.

* Firebase services (Authentication, Notifications) should be integrated into the React Native application for user authentication and push notification functionality.

These tools and functionalities will provide a robust development environment and enable the implementation of essential features for the Job Nexus mobile application.s

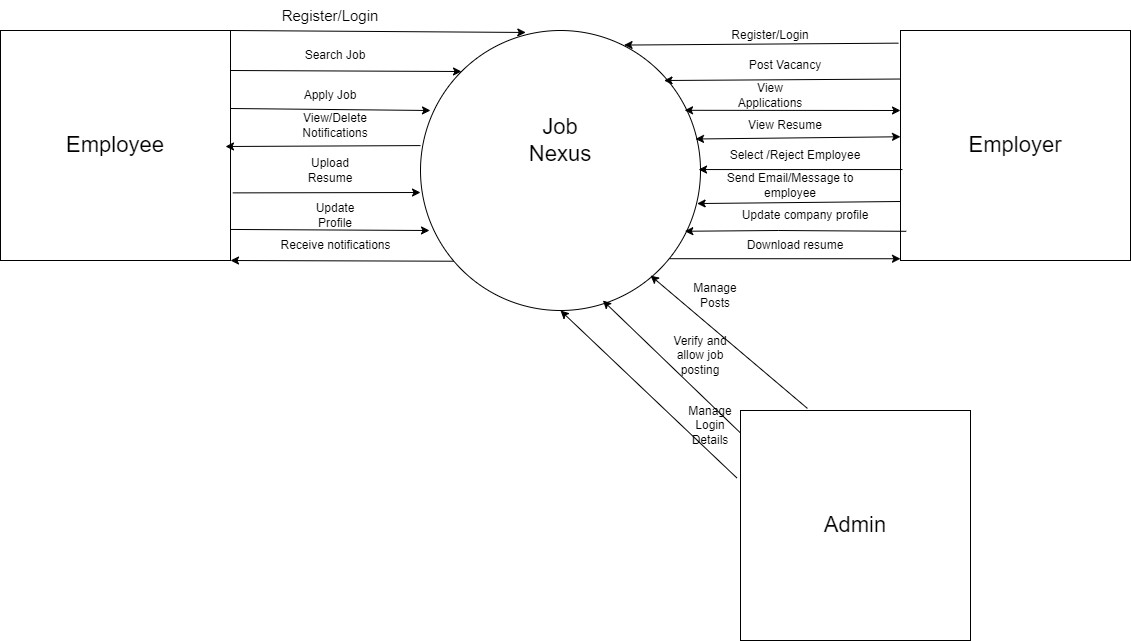
1. User Profile and Settings:- Enable users to view and update their profile information.
2. Backend Integration: - Connect the app with a backend system to handle data storage, product management, order processing, and other business logic.

**6.3 Requirement Matrix:**

|  |  |  |  |
| --- | --- | --- | --- |
| SN | requirement, modules, | Description for the | Priority (high, |
|  | system and features | modules | medium, low) |
|  |  |  |  |
| 1. | Login and security system | Login can be done on the | high |
|  |  | basis of their levels |  |
|  |  |  |  |
| 2. | Data entry | Data entry of the major | high |
|  |  | details of the users |  |
|  |  |  |  |
| 3. | Venue Detail | The venue owner and | high |
|  |  | admin can place the |  |
|  |  | venue details |  |
|  |  |  |  |
| 4. | Checking of Booked and un | User must procced | high |
|  | booked venues | during testing of system |  |
|  |  |  |  |
| 5. | Performance Checkout | System must be able to | high |
|  |  | satisfy the customer in |  |
|  |  | user friendly manner |  |
|  |  |  |  |
| 6. | Booked venue | User must be simple able | high |
|  |  | to retrieve the booked |  |
|  |  | venue |  |
|  |  |  |  |
|  | *Table 1: Requirement Matrix* | |  |

**7. System Analysis and Design:**

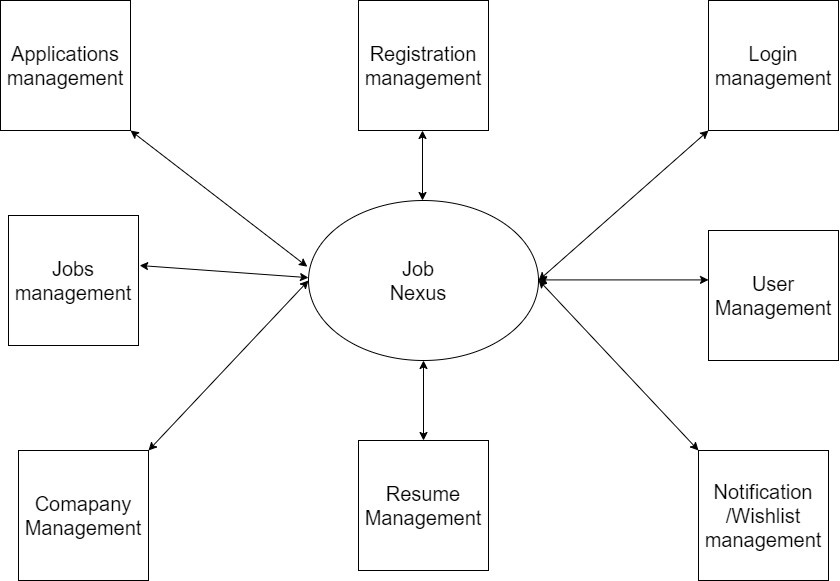
**7.1 Dfd Level 0**



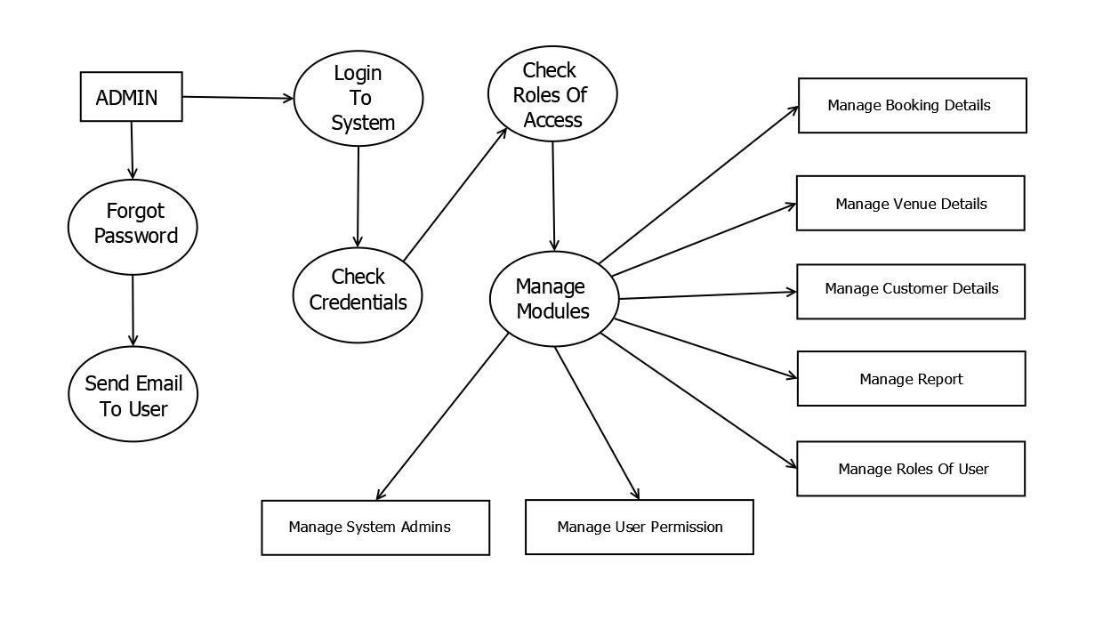
**Fig: Dfd Level 0**

7

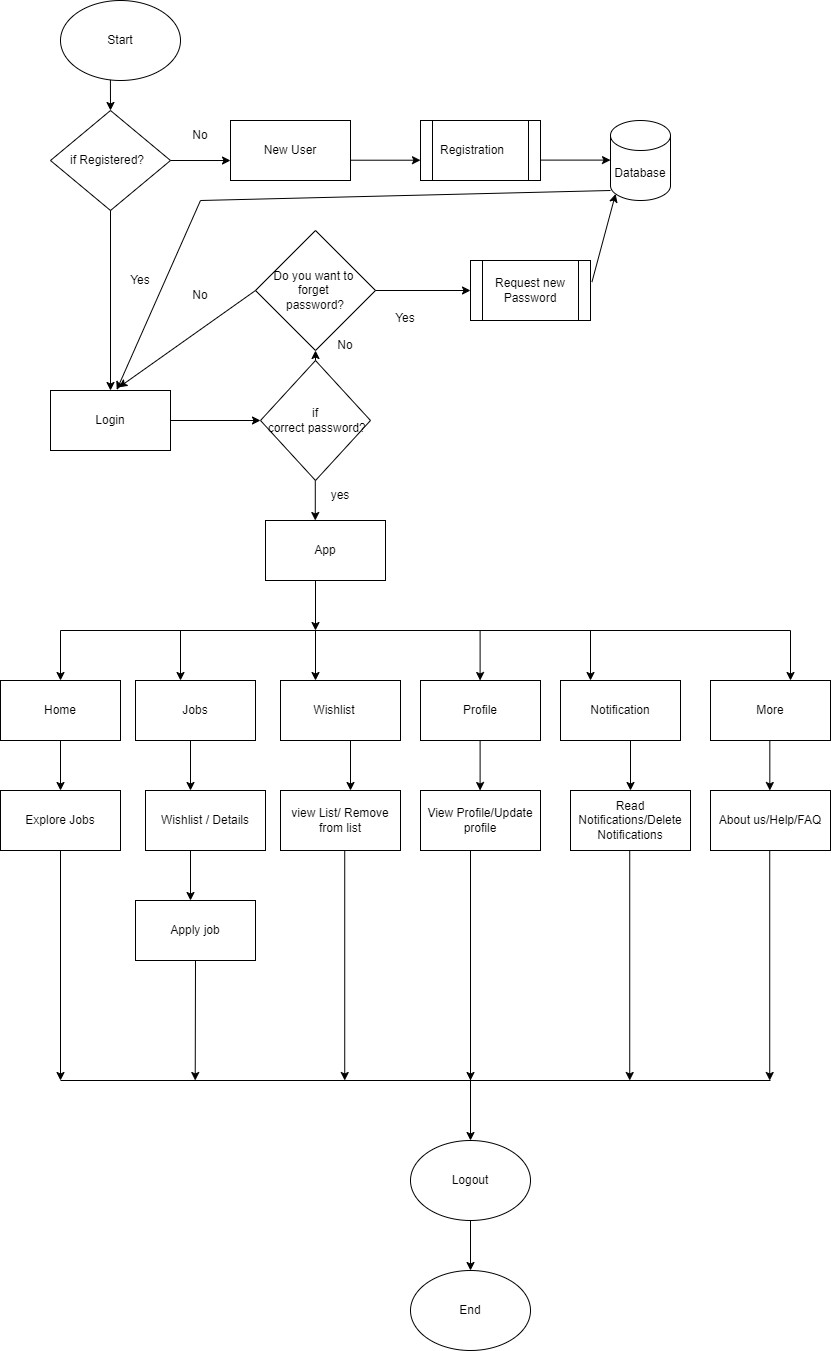
**7.2 Dfd Level 2**



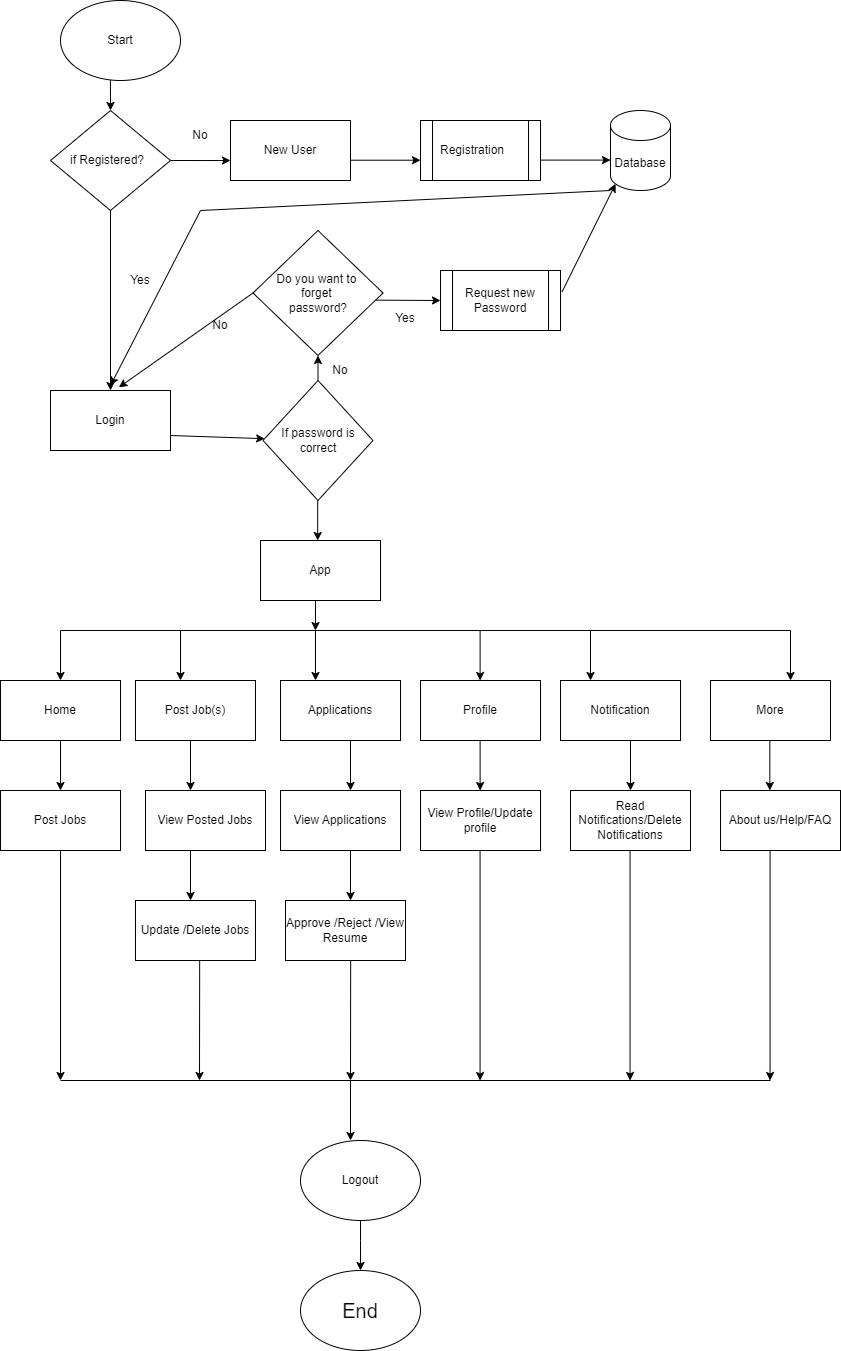
**7.2 Dfd Level 1**



**7.2 Flowchart for Employee**



**7.2 Flowchart for Employee**



**8. Development**

The Venue Match App development commenced with the adoption of the "Iterative Model". The Iterative Model is a software development approach that focuses on incremental and iterative progress throughout a project's lifecycle. In this model, the development process is divided into small cycles or iterations, each of which involves a subset of the project's requirements. The model begins with an initial planning phase where the project's objectives, requirements, and scope are defined. Following this, the first iteration is executed, typically involving design, development, and testing activities.

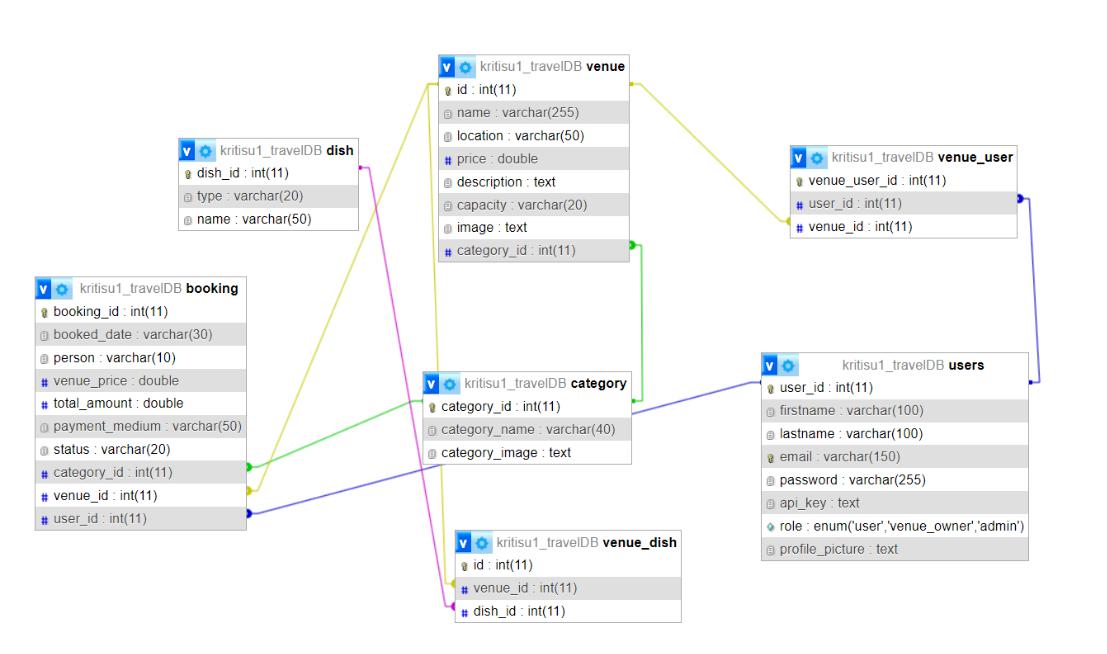
What sets the Iterative Model apart is that it doesn't attempt to define all project requirements upfront. Instead, it allows for flexibility, enabling developers and stakeholders to adapt and refine requirements as the project progresses. Each iteration results in a partially completed version of the software, and feedback from users and stakeholders is gathered and incorporated into the subsequent iterations.

This iterative process continues until the software reaches a state where it meets the desired level of functionality, quality, and user satisfaction. The advantage of the Iterative Model lies in its ability to accommodate changing requirements, respond to user feedback, and reduce the risk of late-stage project failures by addressing issues early in the development process.

As we know the process of planning, organizing, coordinating, and controlling resources to achieve specific goals is also referred to as development. Software engineers have created more sophisticated tools known as IDE (Integrated Development Environment) to accommodate the evolution of software development. We have used Visual studio IDE which has features that integrate flawlessly with flutter frameworks. Likewise, for effective and worthwhile documentation, many inhouse office products like Ms-excel, MS-word, and PowerPoint were used. For the charts and diagrams, DIA tool is used. PHP is a backend tools used in the system. We have used Discord, Git and Microsoft Teams for our collaborative platform to discuss with the problems and task division among the members with our project supervision among the team members in teamwork manner. Our system must store information about the

16

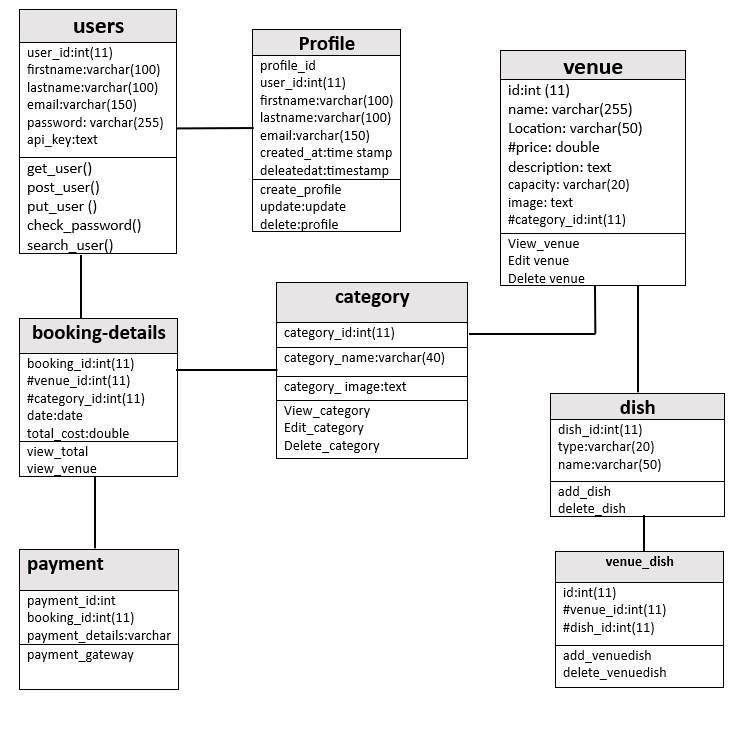
booking, venue details, Reservations, and so on. As a result, we've identified the major tables that will be implemented on the chosen RDBMS. So here is the database schema for our project "Venue Match".



11

12

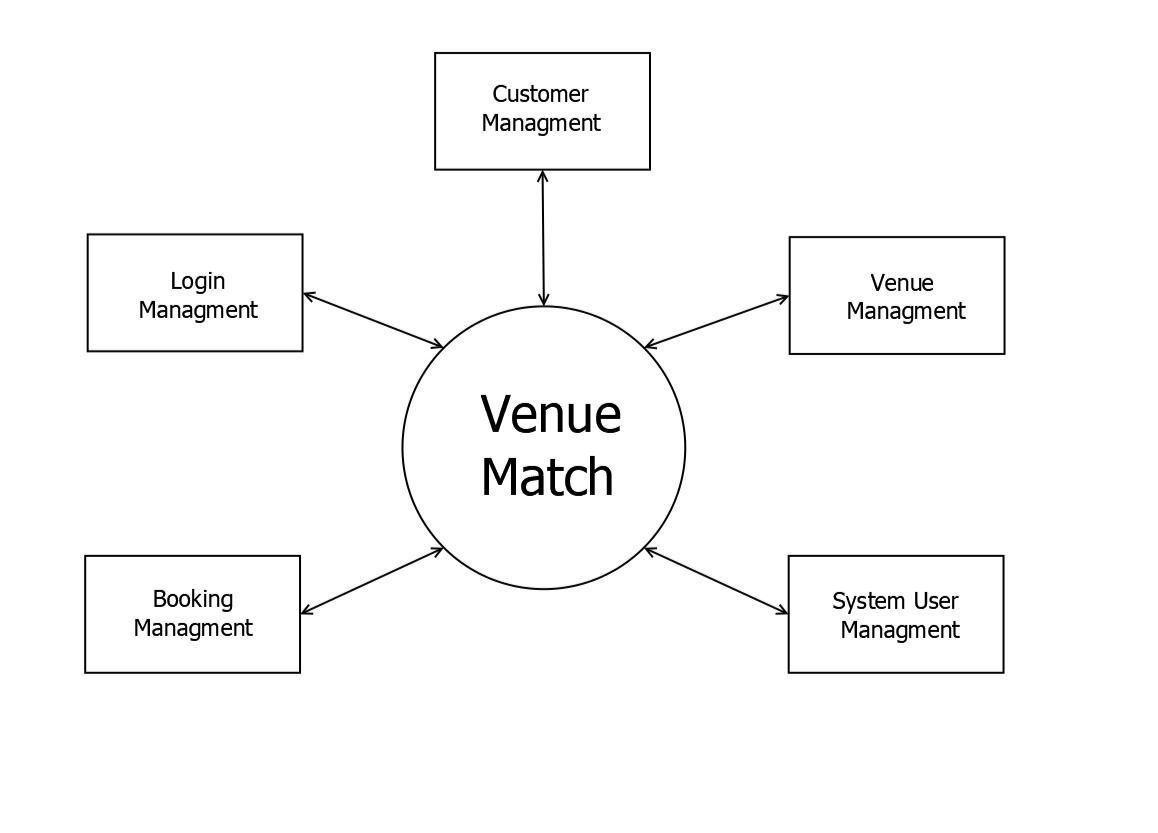
**7.2 Class-Diagram**



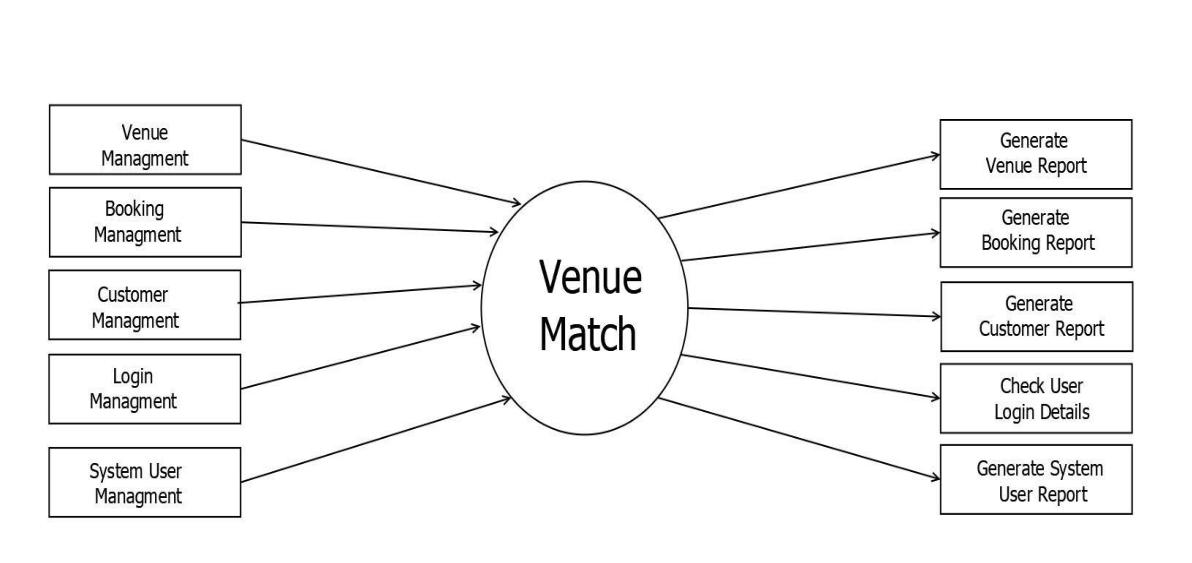
*FIGURE 3 CLASS-DIAGRAM*

13

**7.3 DFD (Data Flow Diagram)**



*FIGURE 4: DFD LEVEL-0*



*FIGURE 5: DFD LEVEL 1*

14

*FIGURE 6: DFD LEVEL 2*

15

*FIGURE 7: DATABASE SCHEME*

The illustrated database visually represents the organization and relationships among the tables within the Venue Match database. This database schema serves as the structured, language-supported description of the database's architecture within a Database Management System (DBMS). The figure above provides a comprehensive list of tables and their associated entities involved in the development of Venue Match.

Specifically, the depicted database schema outlines the organization and interconnections of tables within a database named "kritisu1\_travelDB." Notable tables within the Venue Match database include "kritisu1\_travelDB\_user," "kritisu1\_travelDB\_booking," "kritisu1\_travelDB\_venue\_user," among others. These tables are integral components of the system.

To establish relationships and maintain data integrity, numerous foreign keys are employed throughout the database. A foreign key is a pivotal database constraint that establishes a connection between two tables within a relational database. This constraint

17

actively enforces referential integrity by guaranteeing that the values within a column of one table correspond to the values within a column of another table.

In essence, foreign keys play a vital role in ensuring the coherence and reliability of data across different tables, thus facilitating robust relationships and enabling efficient data retrieval within the Venue Match database.

|  |  |  |  |
| --- | --- | --- | --- |
| S. N | Name of student |  | Work Assigned |
|  |  |  |  |
| 1. | Narvasha Adhikari | o | Documentation |
|  |  | o | Proposal Planning |
|  |  | o | Problem Identification |
|  |  | o | Design (system design) |
|  |  | o Coding (mostly website and some | |
|  |  |  | parts of mobile app) |
|  |  | o | Testing |
|  |  |  |  |
| 2. | Samjhana Poudel | o | Documentation |
|  |  | o | Requirement Analysis |
|  |  | o Design (support in system design) | |
|  |  | o Coding (mostly mobile app and some | |
|  |  |  | parts of website) |
|  |  | o | Backend |
|  |  | o | Testing |
|  |  |  |  |

*Table 2: Task Division*

18

**9. TESTING**

A test case comprises a set of conditions or variables employed by a tester to ascertain if a system being evaluated fulfills prescribed requirements or functions correctly. The process of creating test cases can also aid in revealing potential shortcomings within an application's requirements or design. The Software Testing Life Cycle (STLC) serves as a testing methodology that contributes to the attainment of software quality standards with greater efficiency.

Throughout the entirety of the design and development stages, we have executed unit tests on each individual component, confirming their proper functionality before their integration into the larger system.

Given that we adopted the iterative model for our project's development. The iterative model testing process is an integral component of the iterative development methodology, where software is built and refined incrementally through a series of repetitive cycles. These cycles involve designing, developing, and testing small portions of the software in each iteration. The testing process within the iterative model is dynamic and evolves over the course of multiple iterations.

**9.1. Test cases**

**Test Case -No.1**

**Test Case Id:** TC001, TC002, TC003, TC004, TC005, TC006

**Test priority (low/medium/high):** med

**Module name**: Login module of Venue Match

**Test title:** verification of user login

**Description**: test the login page for valid entry

**Test executed by**: Narvasha Adhikari

**Test reviewed by**: Kreeti Subedi

**Pre-conditions**: user has valid username and password

19

**Test Steps**: Navigate to the login panel in the project's executable file. Enter random characters, zero or null values, capitalized strings, functions inside of functions, and then check to see if the page redirects to the appropriate section.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ste | Test | Test | | Test | Expected |  | Test data | | | |  | Actual | | Status |  |
| p | case | cases | | steps | results |  |  |  |  |  |  | result |  | (Pass/F |  |
|  | id |  |  |  |  |  |  |  |  |  |  |  |  | ail) |  |
|  |  |  | |  |  |  |  |  | |  |  |  |  |  |  |
| 1 | TC0 | verificat | | Navigat | System |  | User= | | | |  | User | is | pass |  |
|  | 01 | ion | | e | displays |  | [narvasha@gmai](mailto:narvasha@gmail.com) | | | |  | navigat | |  |  |
|  |  |  |  | to login | homepag |  | [l.com](mailto:narvasha@gmail.com) | |  |  |  | ed | to |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  | page | e |  | Password= | | | |  | dashbo | |  |  |
|  |  |  |  |  |  |  | narvasha@123 | | | |  | ard |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | with |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | success | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | ful |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | login |  |  |  |
|  |  |  | |  |  |  |  | | | |  |  | |  |  |
| 2 | TC0 | Valid | | Click | Displays |  | User= | | | |  | Error in | | pass |  |
|  | 02 | usernam | | on login | error |  | [user@gmail.co](mailto:user@gmail.com) | | | |  | success | |  |  |
|  |  | e | wrong | button | message |  | [m](mailto:user@gmail.com) | | |  |  | ful |  |  |  |
|  |  | passwor | | enter | login |  |  |  | | |  | login |  |  |  |
|  |  |  | Password= | | | |  |  |  |  |
|  |  | d |  | valid | failed |  | user@123 | | | |  |  |  |  |  |
|  |  |  |  | userna |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | me and |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | passwor |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | d |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | |  |  |  |  | | | |  |  | |  |  |
| 3 | TC0 | Wrong | | Click | Displays |  | User= | | | |  | Error in | | pass |  |
|  | 03 | usernam | | on login | error |  | [user@gmail.co](mailto:user@gmail.com) | | | |  | success | |  |  |
|  |  | e | valid | button | message |  | [m](mailto:user@gmail.com) | | |  |  | ful |  |  |  |
|  |  | passwor | | enter | login |  |  |  | | |  | login |  |  |  |
|  |  |  | Password= | | | |  |  |  |  |
|  |  | d |  | valid | failed |  | user@123 | | | |  |  |  |  |  |
|  |  |  |  | userna |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

20

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | me | and |  |  |  |  |  |  |  |
|  |  |  | passwor | |  |  |  |  |  |  |  |
|  |  |  | d |  |  |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  |  | |  |
| 4 | TC0 | Button | Click | | Successf | | Navigated | to | success | | pass |
|  | 04 | click test | on login | | ul | click | welcome screen | |  |  |  |
|  |  |  | button | | and |  |  |  |  |  |  |
|  |  |  |  |  | further | |  |  |  |  |  |
|  |  |  |  |  | result | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | TC0 | Verify | Try | to | error |  | Click on login |  | error |  | pass |
|  | 05 | login | login | |  |  |  |  |  |  |  |
|  |  | without | without | |  |  |  |  |  |  |  |
|  |  | registrat | providi | |  |  |  |  |  |  |  |
|  |  | ion test | ng |  |  |  |  |  |  |  |  |
|  |  |  | necessa | |  |  |  |  |  |  |  |
|  |  |  | ry |  |  |  |  |  |  |  |  |
|  |  |  | credenti | |  |  |  |  |  |  |  |
|  |  |  | als |  |  |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  |  | |  |
| 4 | TC0 | Forgot | Provide | | Authenti | | User=empty |  | Retriev | | pass |
|  | 06 | passwor | valid | | cate | the | Password=empt | | e | for |  |
|  |  | d test | user |  | user | and | y |  | user |  |  |
|  |  |  | name | | change | |  |  | account | |  |
|  |  |  | and |  | the | new |  |  | s |  |  |
|  |  |  | passwor | | password | |  |  |  |  |  |
|  |  |  | d before | |  |  |  |  |  |  |  |
|  |  |  | clicking | |  |  |  |  |  |  |  |
|  |  |  | forgot | |  |  |  |  |  |  |  |
|  |  |  | passwor | |  |  |  |  |  |  |  |
|  |  |  | d |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  | |  | |  |
|  |  |  |  | *Table 3: Test case for verification of user login* | | | | | | | |

21

**Test Case -No.2**

**Test Case Id:** TC001, TC002, TC003, TC004, TC005, TC006

**Test priority (low/medium/high):** med

**Module name**: Login module of Venue Match

**Test title:** verification of admin login on website

**Description**: test the login page for valid entry

**Test executed by**: Narvasha Adhikari

**Test reviewed by**: Kreeti Subedi

**Pre-conditions**: user has valid username and password

**Test Steps**: Navigate to the login panel in the project's executable file. Enter random characters, zero or null values, capitalized strings, functions inside of functions, and then check to see if the page redirects to the appropriate section.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ste | Test | Test | Test | Expected |  | Test data | |  | Actual | | Status |  |
| p | case | cases | steps | results |  |  |  |  | result |  | (Pass/F |  |
|  | id |  |  |  |  |  |  |  |  |  | ail) |  |
|  |  |  |  |  |  |  | |  |  |  |  |  |
| 1 | TC0 | verificati | Navigat | System |  | User= | |  | User | is | pass |  |
|  | 01 | on | e | displays |  | [admin@gmail.](mailto:admin@gmail.com) | |  | navigat | |  |  |
|  |  |  | to login | homepag |  | [com](mailto:admin@gmail.com) |  |  | ed | to |  |  |
|  |  |  |  |  |  |
|  |  |  | page | e |  | Password= | |  | dashbo | |  |  |
|  |  |  |  |  |  | admin@123 | |  | ard with | |  |  |
|  |  |  |  |  |  |  |  |  | success | |  |  |
|  |  |  |  |  |  |  |  |  | ful |  |  |  |
|  |  |  |  |  |  |  |  |  | login |  |  |  |
|  |  |  |  |  |  |  | |  |  | |  |  |
| 2 | TC0 | Valid | Click on | Displays |  | User= | |  | Error in | | pass |  |
|  | 02 | usernam | login | error |  | [admin@gmail.](mailto:admin@gmail.com) | |  | success | |  |  |
|  |  | e wrong | button | message |  | [com](mailto:admin@gmail.com) |  |  | ful |  |  |  |
|  |  | passwor | enter | login |  | Password= | |  | login |  |  |  |
|  |  | d | valid | failed |  | admin@123 | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

22

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | usernam | |  |  |  |  |  |  |  |  |  |
|  |  |  | e | and |  |  |  |  |  |  |  |  |  |
|  |  |  | passwor | |  |  |  |  |  |  |  |  |  |
|  |  |  | d |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  | | |  | |  |
| 3 | TC0 | Wrong | Click on | | Displays | |  | User= | | | Error in | | pass |
|  | 03 | usernam | login | | error |  |  | [admin@gmail.](mailto:admin@gmail.com) | | | success | |  |
|  |  | e valid | button | | message | |  | [com](mailto:admin@gmail.com) |  |  | ful |  |  |
|  |  | passwor | enter | | login |  |  | Password= | | | login |  |  |
|  |  | d | valid | | failed | |  | admin@123 | | |  |  |  |
|  |  |  | usernam | |  |  |  |  |  |  |  |  |  |
|  |  |  | e | and |  |  |  |  |  |  |  |  |  |
|  |  |  | passwor | |  |  |  |  |  |  |  |  |  |
|  |  |  | d |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  | | |  | |  |
| 4 | TC0 | Button | Click on | | Successfu | |  | Navigated to | | | success | | pass |
|  | 04 | click test | login | | l | click |  | dashboard | | |  |  |  |
|  |  |  | button | | and |  |  |  |  |  |  |  |  |
|  |  |  |  |  | further | |  |  |  |  |  |  |  |
|  |  |  |  |  | result | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | | |  |  |  |
| 4 | TC0 | Verify | Try | to | error |  |  | Click on login | | | error |  | pass |
|  | 05 | login | login | |  |  |  |  |  |  |  |  |  |
|  |  | without | without | |  |  |  |  |  |  |  |  |  |
|  |  | registrati | providin | |  |  |  |  |  |  |  |  |  |
|  |  | on test | g |  |  |  |  |  |  |  |  |  |  |
|  |  |  | necessar | |  |  |  |  |  |  |  |  |  |
|  |  |  | y |  |  |  |  |  |  |  |  |  |  |
|  |  |  | credenti | |  |  |  |  |  |  |  |  |  |
|  |  |  | als |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  | | |  | |  |
| 4 | TC0 | Forgot | Provide | | Authentic | |  | User=empty | | | Retriev | | pass |
|  | 06 | passwor | valid | | ate | the |  | Password=em | | | e | for |  |
|  |  | d test | user |  | user | and |  | pty | | | user |  |  |
|  |  |  | name | | change | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

23

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | and | the new |  | account |  |
|  |  |  | passwor | password |  | s |  |
|  |  |  | d before |  |  |  |  |
|  |  |  | clicking |  |  |  |  |
|  |  |  | forgot |  |  |  |  |
|  |  |  | passwor |  |  |  |  |
|  |  |  | d |  |  |  |  |
|  |  |  |  |  |  |  |  |

*Table 4 : Test case for verification of admin login*

**Test Case -No.3**

**Test Case Id:** TC001, TC002, TC003, TC004, TC005, TC006

**Test priority (low/medium/high):** med

**Module name**: Login module of Venue Match

**Test title:** verification of venue owner login on website

**Description**: test the login page for valid entry

**Test executed by**: Narvasha Adhikari

**Test reviewed by**: Kreeti Subedi

**Pre-conditions**: user has valid username and password

**Test Steps**: Navigate to the login panel in the project's executable file. Enter random characters, zero or null values, capitalized strings, functions inside of functions, and then check to see if the page redirects to the appropriate section.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ste | Test | Test | Test | Expected | Test data | Actual | Status |
| p | case | cases | steps | results |  | result | (Pass/F |
|  | id |  |  |  |  |  | ail) |
|  |  |  |  |  |  |  |  |

24

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TC0 | verificati | | Navigat | | System | |  | User= | | | User | is | pass |  |
|  | 01 | on |  | e |  | displays | |  | [owner@gmail.](mailto:owner@gmail.com) | | | navigat | |  |  |
|  |  |  |  | to | login | dashboar | |  | [com](mailto:owner@gmail.com) |  |  | ed | to |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  | page | | d |  |  | Password= | | | dashbo | |  |  |
|  |  |  |  |  |  |  |  |  | owner@123 | | | ard with | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | success | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | ful |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | login |  |  |  |
|  |  |  | |  | |  | |  |  | | |  | |  |  |
| 2 | TC0 | Valid | | Click on | | Displays | |  | User= | | | Error in | | pass |  |
|  | 02 | usernam | | login | | error |  |  | [owner@gmail.](mailto:owner@gmail.com) | | | success | |  |  |
|  |  | e | wrong | button | | message | |  | [com](mailto:owner@gmail.com) |  |  | ful |  |  |  |
|  |  | passwor | | enter | | login |  |  | Password= | | | login |  |  |  |
|  |  | d |  | valid | | failed | |  | owner@123 | | |  |  |  |  |
|  |  |  |  | usernam | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | e | and |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | passwor | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | d |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | |  | |  | |  |  | | |  | |  |  |
| 3 | TC0 | Wrong | | Click on | | Displays | |  | User= | | | Error in | | pass |  |
|  | 03 | usernam | | login | | error |  |  | [owner@gmail.](mailto:owner@gmail.com) | | | success | |  |  |
|  |  | e | valid | button | | message | |  | [com](mailto:owner@gmail.com) |  |  | ful |  |  |  |
|  |  | passwor | | enter | | login |  |  | Password= | | | login |  |  |  |
|  |  | d |  | valid | | failed | |  | owner@123 | | |  |  |  |  |
|  |  |  |  | usernam | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | e | and |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | passwor | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | d |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | |  | |  | |  |  | | |  | |  |  |
| 4 | TC0 | Button | | Click on | | Successfu | |  | Navigated to | | | success | | pass |  |
|  | 04 | click test | | login | | l | click |  | dashboard | | |  |  |  |  |
|  |  |  |  | button | | and |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | further | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | result | |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

25

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | TC0 | Verify | Try to | error |  | Click on login | error |  | pass |
|  | 05 | login | login |  |  |  |  |  |  |
|  |  | without | without |  |  |  |  |  |  |
|  |  | registrati | providin |  |  |  |  |  |  |
|  |  | on test | g |  |  |  |  |  |  |
|  |  |  | necessar |  |  |  |  |  |  |
|  |  |  | y |  |  |  |  |  |  |
|  |  |  | credenti |  |  |  |  |  |  |
|  |  |  | als |  |  |  |  |  |  |
|  |  |  |  |  | |  |  | |  |
| 4 | TC0 | Forgot | Provide | Authentic | | User=empty | Retriev | | pass |
|  | 06 | passwor | valid | ate | the | Password=em | e | for |  |
|  |  | d test | user | user | and | pty | user |  |  |
|  |  |  | name | change | |  | account | |  |
|  |  |  | and | the | new |  | s |  |  |
|  |  |  | passwor | password | |  |  |  |  |
|  |  |  | d before |  |  |  |  |  |  |
|  |  |  | clicking |  |  |  |  |  |  |
|  |  |  | forgot |  |  |  |  |  |  |
|  |  |  | passwor |  |  |  |  |  |  |
|  |  |  | d |  |  |  |  |  |  |
|  |  |  |  |  | |  |  |  |  |
|  |  | *Table 5: Test case for verification of venue owner login* | | | | | |  |  |

**Test Case -No.4**

**Test Case Id:** TC001, TC002, TC003, TC004, TC005, TC006

**Test priority (low/medium/high):** med

**Module name**: Registration

**Test title:** test of registration

**Description**: registration of new user

**Test executed by**: Narvasha Adhikari

26

**Test reviewed by**: Kreeti Subedi

**Pre-conditions**: user has valid username and password; email id must not be used before registration to the system i.e., is unique email id for each user

**Test Steps**: click on register button and fill with correct data

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ste | Test | Test |  | Test | Expected | |  | Test data | |  | Actual | Status |  |
| p | case | cases |  | steps | results |  |  |  |  |  | result | (Pass/F |  |
|  | id |  |  |  |  |  |  |  |  |  |  | ail) |  |
|  |  |  | |  |  |  |  |  | |  |  |  |  |
| 1 | TC0 | Registrat | | Click | All | the |  | email= | |  | As | pass |  |
|  | 01 | ion | of | on | required | |  | [narvasha@gmai](mailto:narvasha@gmail.com) | |  | expecte |  |  |
|  |  | new user | | register | fields are | |  | [l.com](mailto:narvasha@gmail.com) | |  | d, |  |  |
|  |  |  |  | button | filled and | |  |  |  |  |  |  |  |
|  |  |  |  |  | Password= | |  |  |  |  |
|  |  |  |  | and fill | processe | |  | narvasha@123 | |  |  |  |  |
|  |  |  |  | fields of | d | to |  | Contact: | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | forms | verificati | |  | 9806669990 | |  |  |  |  |
|  |  |  |  | i.e. | on | of |  |  |  |  |  |
|  |  |  |  |  | Address: | |  |  |  |  |
|  |  |  |  | email id | details |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | contact |  |  |  | Lakeside, | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | number, |  |  |  | Pokhara | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | address |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | |  |  | |  |  |  |  |
| 2 | TC0 | Valid |  | Click | Displays | |  | email= | |  | Error in | pass |  |
|  | 02 | usernam | | on login | error |  |  | [narvasha@gmai](mailto:narvasha@gmail.com) | |  | success |  |  |
|  |  | e wrong | | button | message | |  | [l.com](mailto:narvasha@gmail.com) | |  | ful |  |  |
|  |  | passwor | | enter | login |  |  |  |  |  | login |  |  |
|  |  |  |  | Password= | |  |  |  |
|  |  | d |  | valid | failed |  |  | narvasha@123 | |  |  |  |  |
|  |  |  |  | userna |  |  |  | Contact: | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | me and |  |  |  | 9806669990 | |  |  |  |  |
|  |  |  |  | passwor |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | Address: | |  |  |  |  |
|  |  |  |  | d |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

27

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Lakeside, | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | Pokhara | |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  | |  |  |  | |  |  |
| 3 | TC0 | Wrong | Click | | Displays | |  | User= | |  |  | Error in | | pass |  |
|  | 03 | usernam | on login | | error |  |  | [owner@gmail.c](mailto:owner@gmail.com) | | | | success | |  |  |
|  |  | e valid | button | | message | |  | [om](mailto:owner@gmail.com) | |  |  | ful |  |  |  |
|  |  | passwor | enter | | login | |  |  |  |  |  | login | |  |  |
|  |  |  | Password= | |  |  |  |  |
|  |  | d | valid | | failed | |  | owner@123 | |  |  |  |  |  |  |
|  |  |  | userna | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | me | and |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | passwor | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | d |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  | |  | |  | |  |  |
| 4 | TC0 | Button | Click | | Successf | |  | Navigated | | to | | success | | pass |  |
|  | 04 | click test | on login | | ul | click |  | dashboard | |  |  |  |  |  |  |
|  |  |  | button | | and |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | further | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | result | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |
| 4 | TC0 | Verify | Try | to | error |  |  | Click on login | |  |  | error |  | pass |  |
|  | 05 | login | login | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | without | without | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | registrati | providi | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | on test | ng |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | necessa | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | ry |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | credenti | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | als |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | |  | |  |  | |  |  |  | |  |  |
| 4 | TC0 | Forgot | Provide | | Authenti | |  | User=empty | |  |  | Retriev | | pass |  |
|  | 06 | passwor | valid | | cate | the |  | Password=empt | | | | e | for |  |  |
|  |  | d test | user |  | user | and |  | y | |  |  | user |  |  |  |
|  |  |  | name | | change | |  |  |  |  |  |  |  |  |  |
|  |  |  | and |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

28

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | passwor | the new |  | account |  |
|  |  |  | d before | password |  | s |  |
|  |  |  | clicking |  |  |  |  |
|  |  |  | forgot |  |  |  |  |
|  |  |  | passwor |  |  |  |  |
|  |  |  | d |  |  |  |  |
|  |  |  |  |  |  |  |  |

*Table 6: Test case for verification of registration*

Test case No-5

Test case ID – TC001, TC002

Test Priority (Low/Medium/High): Medium

Module Name: Booking venue

Test Title: Booking testing

Description: User successfully books Venue A, is able to select their preferred date and time for the test, reviews the booking details, confirms the booking, and ensures that the confirmation is accurate.

Test Executed by: Samjhana Poudel

Test Reviewed by : Suman Poudel

Pre-condition: User must be logged in.

Test Steps : Select the desired venue and navigate to booking screen and select date, total guests

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step 1 | Test Case | Test Cases | Test | Expected | Test | Actual | Status |  |
|  | Id |  | Steps | Results | Data | Results | (Pass/Fail) |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 1 | TC001 | User | Select | Navigate | Venue id | As | Pass |  |
|  |  | Should be | the | to | Category | expected |  |  |
|  |  | logged in | desired | booking | id |  |  |  |
|  |  |  | venue | screen |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

29

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Click on |  |  |  |  |
|  |  |  | book |  |  |  |  |
|  |  |  | now |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 2 | TC002 | User | User | Display | Total | As | Pass |
|  |  | should | should | the total | Amount | expected |  |
|  |  | navigate | select | details |  |  |  |
|  |  | to the | the date | and total |  |  |  |
|  |  | booking | and total | amount |  |  |  |
|  |  | screen | guests | for |  |  |  |
|  |  |  |  | booking |  |  |  |
|  |  |  |  | venue |  |  |  |
|  |  |  |  |  |  |  |  |
| 3 | TC003 | User | Click on | Navigate |  | As | Pass |
|  |  | should | book | to |  | expected |  |
|  |  | navigate | now | payment |  |  |  |
|  |  | to |  | screen |  |  |  |
|  |  | payment |  |  |  |  |  |
|  |  | screen |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

*Table 7: Test case for booking venue*

Test case No-6

Test case ID – TC001

Test Priority (Low/Medium/High): Medium

Module Name: Payment integration

Test Title: Payment testing

Description: A user should have the capability to successfully initiate a payment and successfully finalize the booking procedure.

Test Executed by: Samjhana Poudel

Test Reviewed by: Suman Poudel

Pre-condition: The user needs to have an active login session, choose their preferred venue, and complete all the necessary information on the booking screen.

30

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Test | Test | Test | Expected | Test Data | Actual | Status |  |
| 1 | Case Id | Cases | Steps | Results |  | Results | (Pass/Fail) |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 1 | TC001 | Payment | Navigate to | Venue | Phone no | As | Pass |  |
|  |  | process | payment | booked | and | expected |  |  |
|  |  |  | page. Enter |  | password |  |  |  |
|  |  |  | user's |  |  |  |  |  |
|  |  |  | payment |  |  |  |  |  |
|  |  |  | information. |  |  |  |  |  |
|  |  |  | Click on |  |  |  |  |  |
|  |  |  | "Pay Now" |  |  |  |  |  |
|  |  |  | button. |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

*Table 8: Test case for payment*

31

*FIGURE 8 GANTT CHART*

32

**11. Project Results:**

Venue Match App the Venue Match App was developed with the objective of providing users with a convenient platform to find suitable venues for their events and gatherings. After thorough development and testing, the app achieved the following results:

User-Friendly Interface: The Venue Match App boasts an intuitive and user-friendly interface, allowing users to easily navigate through the app's features and functionalities. The design incorporates a clean layout, making it simple for users to search, filter, and view venue options.

Accurate Venue Recommendations: The venue matching algorithm implemented in the app delivers accurate and relevant recommendations based on user preferences, such as location, capacity, amenities, and event type. Users can quickly find venues that align with their specific requirements.

Efficient Search and Filtering: The search and filtering mechanisms are efficient and responsive, enabling users to explore various venues and narrow down options quickly. Filters for price range, event type, and available dates enhance the search experience.

Seamless Booking Process: The app provides a seamless booking process, allowing users to reserve their chosen venues directly within the app. This streamlined process eliminates the need for manual bookings and reduces potential user frustrations.

Real-time Availability Tracking: Venue owners can update venue availability in real-time, ensuring that users receive up-to-date information about available dates and times.

Reliable and Secure Authentication: Robust authentication and authorization mechanisms ensure the security of user accounts and personal information. Users can confidently create accounts and log in using secure methods like email/password, social media login, or OAuth.

33

Scalable Architecture: The app was built using a scalable architecture, allowing it to handle increasing user traffic and growing venue listings without compromising performance.

Responsive Design: The Venue Match App's responsive design guarantees a consistent user experience across various devices and screen sizes, including smartphones, tablets, and web browsers.

Error Handling and Feedback: The app incorporates comprehensive error handling and provides meaningful feedback to users in case of errors or unsuccessful operations, enhancing the overall user experience. Extensibility and Maintainability: The project was developed with a focus on extensibility and maintainability. The codebase follows best practices and design patterns, making it easier to add new features and conduct future updates.

34

**12.Future Enhancement**

As the Venue Match App evolves and aims to provide an even better experience for its users, several future enhancements can be considered:

* Offer real-time availability updates and instant booking confirmation for users to access the latest information and secure reservations quickly.
* Enhance the app with event planning features: guest lists, and customization for seamless event management.
* Add venue comparison with user reviews and ratings for easy evaluation of different venues based on specific criteria.
* Implement social media integration to enable users to easily share their booked venues and event details, fostering social engagement by spreading the word among friends and family.
* Email verification and validation to ensure that an email address is both deliverable and genuine.
* Venue owner verification through documentation review (Proof of Identity**)**, Legal Agreements, Background Checks, Site visit.
* Introduce virtual 360-degree venue tours for immersive exploration, enhancing user understanding of venue spaces.
* Creating a smooth integration between the two platforms ( app and website) so that users can transition between them effortlessly with just a single action, typically a button or link.

35

**13.Conclusion**

The Venue Management System serves as a comprehensive and efficient solution, streamlining various facets of venue management. Notably, it enhances venue operations by simplifying event scheduling, bookings, resource allocation, and communication between stakeholders, markedly improving operational efficiency. Additionally, it fosters an improved customer experience, offering user-friendly interfaces for seamless venue browsing, availability checks, and reservations. This optimizes resource allocation by managing event calendars, bookings, and staff/equipment assignments, providing real-time visibility into resource availability, boosting productivity, and reducing costs. Effective communication is facilitated through various channels, minimizing misunderstandings and enhancing overall efficiency. Moreover, the system generates comprehensive reports and analytics, offering valuable insights into venue utilization, revenue generation, and customer preferences, empowering informed decision-making. Designed for scalability and customization, it adapts to diverse venue requirements, ensuring adaptability and easy expansion. In summary, the Venue Management System is a robust tool that significantly enhances venue management, resulting in increased efficiency, customer satisfaction, and business success upon implementation.

36

1. **References**
   1. Adler, T. (2007, March ). *Scribd*. Retrieved from Event Management System: https://www.scribd.com
   2. Alejandro and Pablo Blanes, J. C. (2010). *FLATICON*. Retrieved from Icons: https://www.flaticon.com/
   3. *Code Projects*. (2017). Retrieved from Event management

Project : https://code-projects.org

1. Flutter. (n.d.). *flutter* . Retrieved from https://flutter.dev/
2. *FreeProjectz*. (2014, November 7). Retrieved from event managment: https://www.freeprojectz.com
3. Jain, S. (n.d.). *GeeksforGeeks*. Retrieved from Flutter – Row and Column Widgets: https://www.geeksforgeeks.org
4. Maxim Melamedov, L. G. (n.d.). *Techopedia*. Retrieved from Tech: https://www.techopedia.com/

8. OpenAI. (2022, November

30).

*ChatGPT*.

Retrieved

from https://chat.openai.com/

1. Prosus. (2008, September 15 ). *StackOverflow*. Retrieved from SafeArea in Flutter: https://stackoverflow.com

10. shoutem. (n.d.). *shoutem*. Retrieved https://shoutem.com/blog/app-ideas-for-beginners/?

from app-ideas:

11. wiki. (n.d.). *wikipedia*.

Retrieved

from wikipedia.org:

https://en.wikipedia.org/wiki/

37

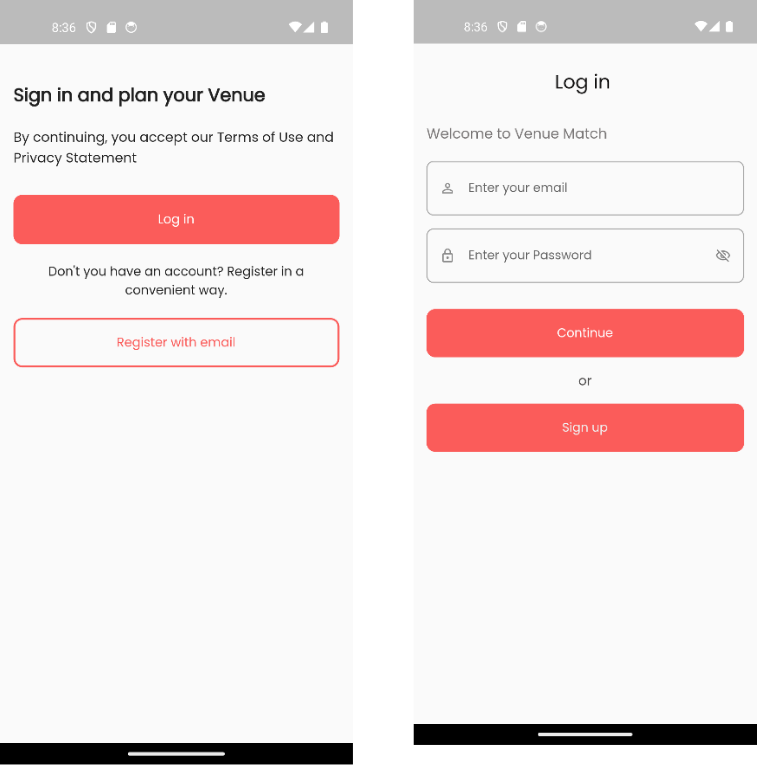
1. wikijsegfug. (n.d.). *flutterhuiewgf*. Retrieved from wiki fulutter: www.wiki.com
2. Zack Onisko, D. C. (2009, July 9). *Dribbble*. Retrieved from design: https://dribbble.com/

38

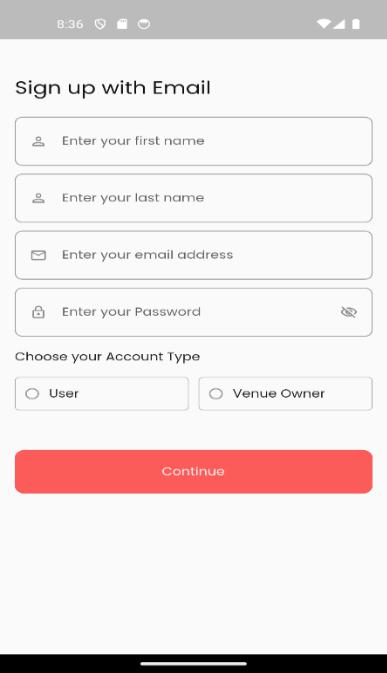
**Annexure**

During the development of our project we have faced unfamiliar errors while developing various services and we have come across to solve the problem within the given time period of time. So we are finally able to deliver the website and mobile application that will satisfy the customers. Here are some screenshots of our system

39

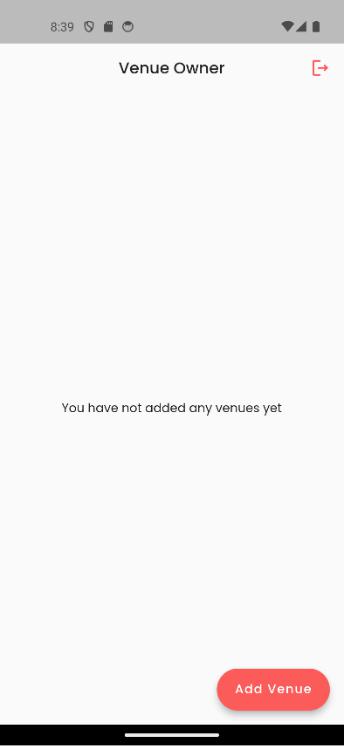
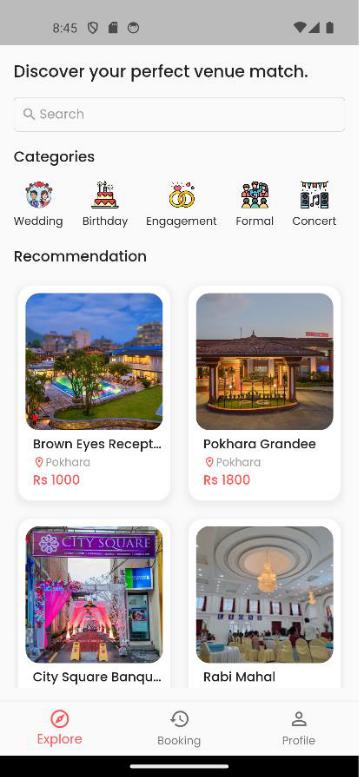


|  |  |  |
| --- | --- | --- |
|  | **WELCOME SCREEN** |  |
| **FIG: WELCOME SCREEN** | **FIG: LOGIN SCREEN** |  |
|  |  |



**FIG: SIGNUP SCREEN**

40

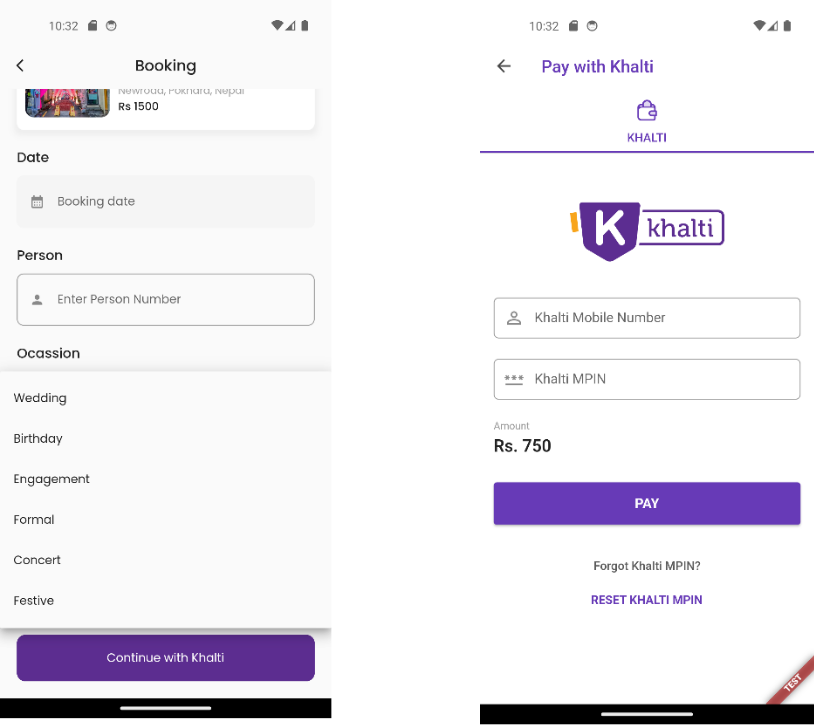


**FIG: VENUE OWNER PAGE** **FIG: EXPLORE SCREEN**



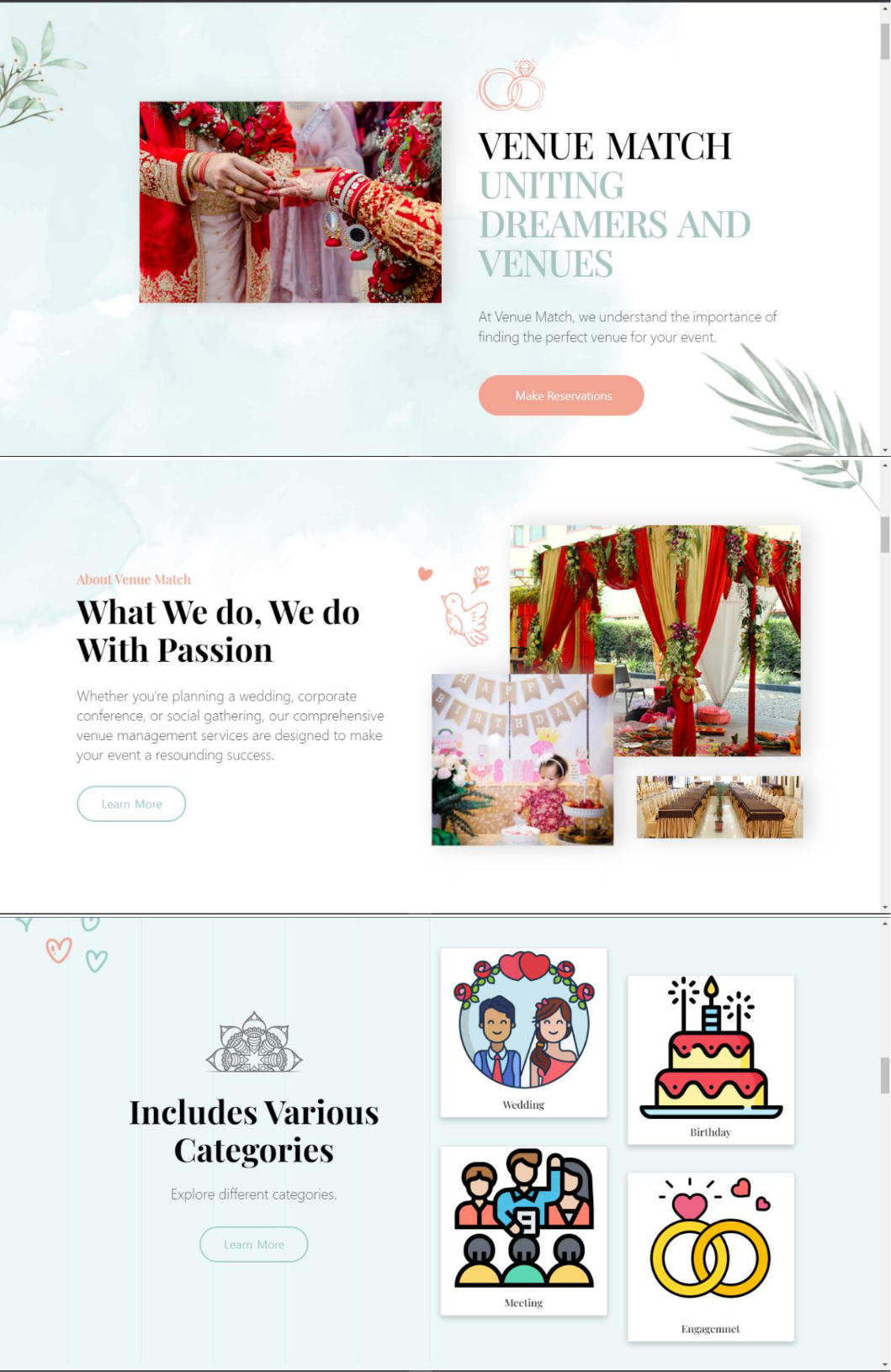
**FIG: DETAIL SCREEN**

41



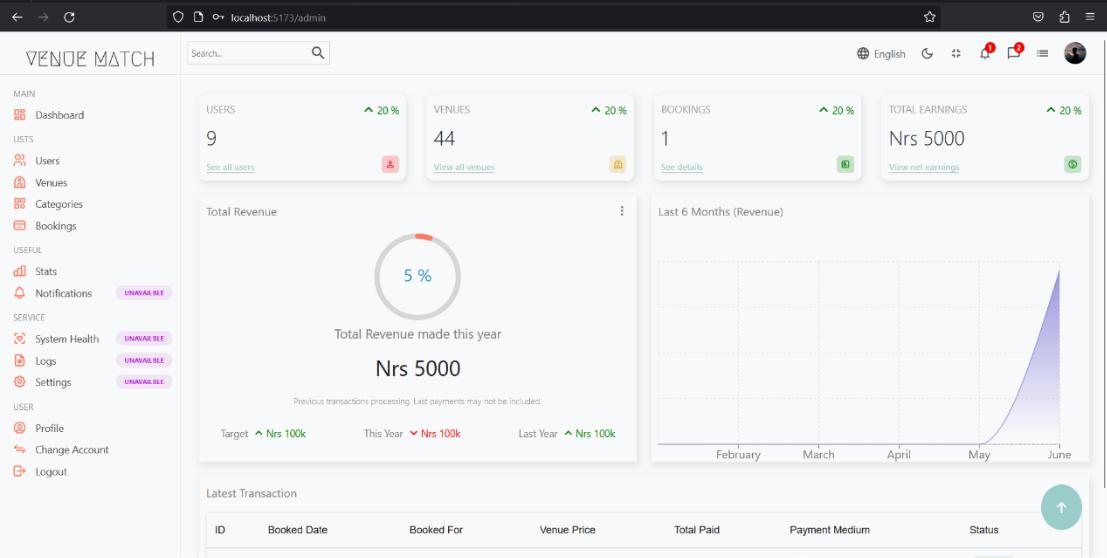
**FIG: BOOKING AND PAYMENT SCREEN**

42



**FIG: WEBSITE LANDING PAGE**

43



**FIG: ADMIN DASHBOARD**

44