Automatic Portfolio Website Generator

(Create your personalized portfolio in minutes!)

Supervised by

Nusrat Jahan
Assistant Professor and Head, Dept. of ITM
Department of Software Engineering
Daffodil International University

Submitted By

Md. Samiul Alam Siam
ID: 212-35-3179
Department of Software Engineering
Daffodil International University

APPROVAL

This thesis titled on "**Automatic Portfolio Website Builder**", submitted by **Md. Samiul Alam Siam (ID: 212-35-3179)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Fazla Elahe

Chairman

Dr. Md. Fazla Elahe
Assistant Professor & Associate Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Marzia 13.9.25

Internal Examiner 1

Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

SV 13.09.2025

Internal Examiner 2

Dr. Shabnom Mustary
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

13.09.25

External Examiner

Mohammod Abul Kashem
Professor
Department of Computer Science and Engineering
Dhaka University of Engineering & Technology, Gazipur.

DECLARATION

I hereby declare that; this project has been done by me under the supervision of **Ms. NUSRAT JAHAN, Assistant Professor, Department of Software Engineering, Faculty of Science and Information Technology, Daffodil International University.** I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Md. Samiul Alam Siam
ID: 212-35-3179
Department of Software Engineering
Daffodil International University

Certified by

Ms. Nusrat Jahan
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

ACKNOWLEDGEMENT

I want to start by expressing my deep gratitude to Almighty Allah. I am grateful for His blessings, which gave me the strength, focus, and determination to complete this project.

I am truly thankful to my parents for their constant encouragement and support. Their faith in my abilities has motivated me throughout my academic journey and this achievement.

I want to extend my heartfelt appreciation to **Prof. Dr. Imran Mahmud, Head of the Department of Software Engineering at Daffodil International University**. His guidance and ongoing motivation were essential during this project.

I also want to thank **Ms. Nusrat Jahan.** Her valuable insights, consistent support, and careful supervision were crucial in shaping the final result.

I appreciate all the faculty members and lab assistants in the Department of Software Engineering for their help and technical support throughout the development process.

Finally, I am grateful to my classmates and friends for their collaboration, suggestions, and encouragement during this journey. Their support, whether big or small, has meant a lot to me.

ABSTRACT

The platform provides job alerts and a search tool that links users to external job sites. Its clean, responsive interface helps showcase academic work, skills, and experience more effectively. This platform allows users to create personalized portfolios accessible from any device. It makes career opportunities more targeted.

The platform builds a connection between academic knowledge and professional application. This helpsto build individual visibility and a networked community of learners and professionals. The platform focuses on that, so users with different technical backgrounds can easily create, customize and update their portfolios.

Beyond improving visibility, it promotes learning through shared projects and teamwork. In doing so, it connects education with professional goals.

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Project Overview

The Automatic Portfolio Website Generator is an online system designed for users, especially students and recent graduates, to create professional portfolios quickly. By entering key personal and career information, users can generate a complete website that showcases their profile. In addition to portfolio creation, the system offers two special features: a Job Notification Bar that displays updated Job posts and a Job search function that links to major Job sites. These features help users build their careers and online presence.

1.2 Project Purpose

This project helps users create a professional online identity without needing coding skills. Users can enter their basic information, and the system builds their website with an interactive design. In addition to creating a portfolio, the system supports career growth by sending daily Job updates and providing a Job search page. These features make it easier for users to stay informed and apply for opportunities quickly.

1.2.1 Background

Creating a professional portfolio is crucial for students and Job seekers to showcase their skills, projects, and experiences. However, many struggle to build an online presence due to limited technical skills or time. Designing and hosting a personal portfolio website can be challenging and time-consuming, especially without web development experience. This system fills that gap by offering an automated solution that generates customized portfolios. By integrating features like a Job Notification Bar and a Search Jobs tool, the platform not only streamlines portfolio creation but also helps users stay updated with relevant career opportunities.

1.2.2 Benefits and Beneficiaries

This project supports students, recent graduates, and Job seekers. Its key benefits include:

- A fully functional, responsive, and easy-to-use web platform.
- Users can build professional portfolios with minimal technical effort.
- Users receive daily Job alerts through the Job Notification Bar.
- The Search Jobs feature helps users quickly find Job listings from popular portals.
- An updated online portfolio increases user visibility to potential employers.

1.2.3 Goal

- This system helps users, especially students and Job seekers, build a strong personal brand and online presence.
- It encourages originality and creativity in displaying personal projects and skills, reducing reliance on generic or copied templates.
- Job-related features keep users informed and assist them in actively seeking career opportunities.

1.3 Stakeholders

There are two main types of stakeholders in this project:

1. **Job Seeker**
   - They are the primary users who create and manage their portfolio websites.
   - They utilize features like the Job Notification Bar and Search Jobs tool to find relevant opportunities.
2. **System Administrator**
   - They manage the platform, ensure system integrity, and maintain data security.
   - They handle user management, content updates, and monitor platform performance.

1.4 Proposed System Model

The proposed system model outlines the development approach used in building the **Automatic Portfolio Website Generator**. It includes the methodology, project structure, and the integration of new features that distinguish this system from existing platforms.

This project follows the **Waterfall Model**, as shown in **Figure 1**, due to its structured and sequential development process.

1.4.1 Waterfall Model

The **Waterfall Model** was chosen for this project based on the following reasons:

- The project requirements are well-defined, clear, and fixed.
- Tasks can be easily organized and tracked.
- Each development stage is clearly defined and structured.
- The model is simple, easy to understand, and straightforward to implement.
- It enables smooth communication with users regarding system flow.
- Provides a strong, disciplined approach to project development.

1.4.2 How We Used the Waterfall Model

1. **Requirements**: Gather system requirements including portfolio builder, Job notification, and Job search modules.
2. **Analysis**: Analyze technical feasibility and system structure.
3. **Design**: Create UI/UX wireframes and database schema.
4. **Implementation**: Develop the system based on design and requirements.
5. **Testing**: Unit and system testing of all modules.
6. **Deployment**: Deploy the application to a live environment.
7. **Maintenance**: Fix bugs and update features post-launch.

Project proposed model

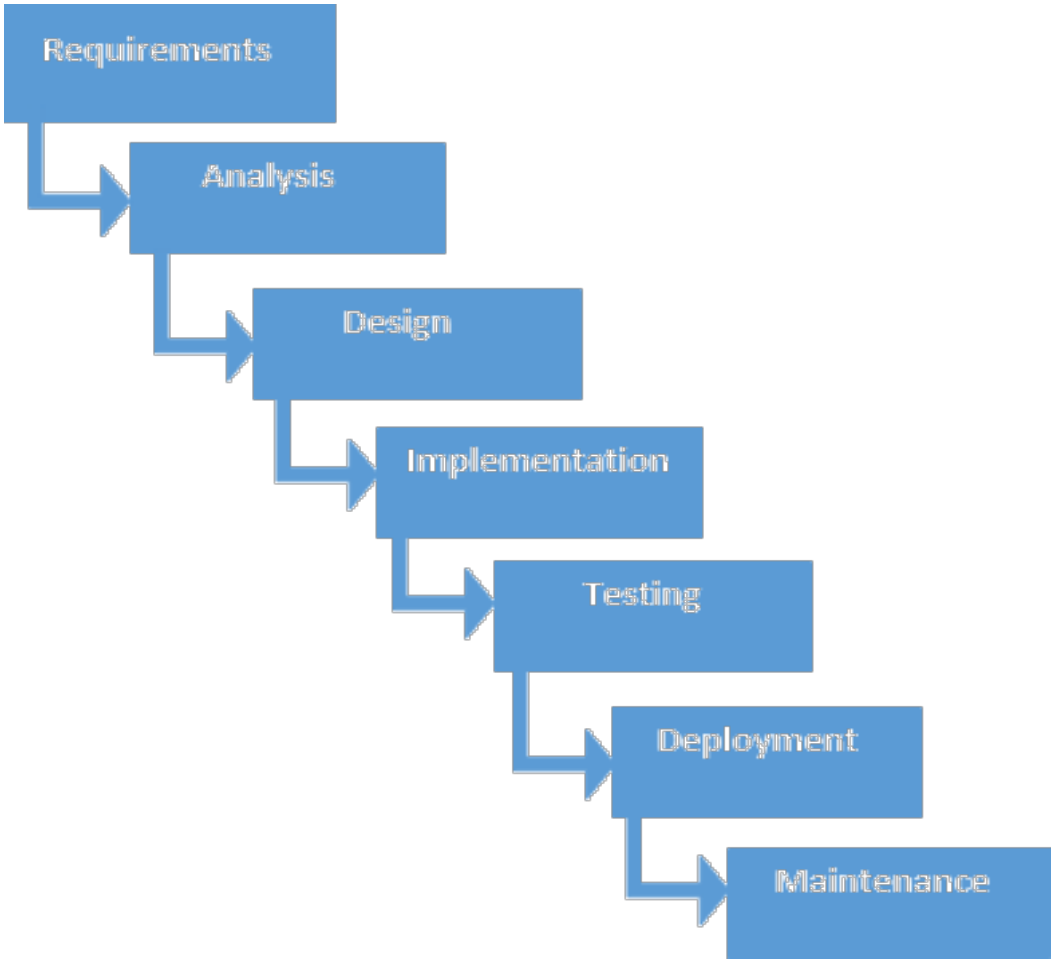


Figure 1: Project proposed model

1.5 Project Schedule

To ensure all tasks are Done on time, I created a project timetable.

1.5.1 Gantt Chart

To complete the project on time and according to plan, a structured schedule was maintained throughout the development cycle. A **Gantt chart** was used to visually track each phase, showing task durations and progress through horizontal bars. This helped ensure timely delivery and efficient workflow management., **Figure 2**

Gantt chart

Task	Start Date	End Date	Duration	Status	Feb	March	April	May	June	July
Requirement Analysis	01-02-25	10-02-25	10 days	Done						
System Design (UI/UX + ERD)	11-02-25	25-02-25	15 days	Done						
Front-End Development	26-02-25	20-03-25	23 days	Done						
Back-End Development (Firebase)	21-03-25	20-04-25	31 days	Done						
Firebase Integration	21-04-25	30-04-25	10 days	Done						
Search and Notification Features	01-05-25	20-05-25	20 days	Done						
Testing and Debugging	21-05-25	10-06-25	21 days	Done						
Final Deployment	11-06-25	15-06-25	5 days	Done						
Documentation and Presentation	16-06-25	25-07-25	40 days	Done						

Figure 2: Gantt chart

1.5.2 WBS Planning for Development Phase

- **Project Plan**

  [01 February 2025 to 25 July 2025]

- **Requirement Gathering**

  [01 February 2025 to 10 February 2025]

1. Brainstorming
2. Interviews
3. Research and Observation
4. Feasibility Analysis

- **Analysis**

  [11 February 2025 to 15 February 2025]

1. Use Case Study
2. Functional and Non-functional Requirements
3. Initial Database Planning

- **Design Phase**

  [16 February 2025 to 05 March 2025]

1. System Architecture
2. UI/UX Wireframes
3. ERD and UML Diagrams
4. Firebase/Backend Planning

- **Development Phase**

  [06 March 2025 to 30 May 2025]

1. Front-End (HTML, CSS, JS, Bootstrap)
2. Back-End (Firebase)
3. Firebase Integration
4. Job Notification and Search Features

- **Testing Phase**

  [01 June 2025 to 15 June 2025]

1. Test Plan Creation
2. Unit and Functional Testing
3. Bug Fixes and Debugging

- **Deployment and Finalization**

  [16 June 2025 to 25 July 2025]

1. Final Report and Documentation
2. Final Presentation Preparation

CHAPTER 2: SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

The process of figuring out user needs and ensuring the system meets them is called requirement analysis. It is one of the most important stages of system development and project management.

2.1 Functional Requirements

Functional requirements outline the specific behaviors and functions that the system must provide. The functional requirement table below highlights the major features of the Portfolio Website Builder System, laying the foundation for successful deployment.

Admin Registration

FR-01	Admin Registration
Description	Admin will monitor the entire system after registration.
Stakeholder	Admin

User Registration

FR-02	User Registration
Description	Allows users to register an account in the system.
Stakeholder	User

User Login

FR-03	User Login
Description	Enables users to Login to the platform to manage their portfolios.
Stakeholder	User

Admin Login

FR-04	Admin Login
Description	Admin can Login to access management functionalities.
Stakeholder	Admin

View Available Designs

FR-05	View Available Designs
Description	Users can explore available portfolio templates before selection.
Stakeholder	User

Add New Design

FR-06	Add New Design
Description	Admin can add new portfolio templates to the system.
Stakeholder	Admin

Fill Portfolio Form

FR-07	Fill Portfolio Form
Description	Users can enter their personal/professional data for portfolio creation.
Stakeholder	User

Preview Form

FR-08	Preview Form
Description	Allows users to preview their portfolio content before generation.
Stakeholder	User

Generate Portfolio

FR-09	Generate Portfolio
Description	Based on the filled form, a complete portfolio site is generated.
Stakeholder	User

Host Portfolio

FR-10	Host Portfolio
Description	Users can host their generated portfolio website.
Stakeholder	User

Job Notification Bar

FR-11	Job Notification Bar
Description	Displays relevant Job opportunities linked with the portfolio.
Stakeholder	User

Search Jobs

FR-12	Search Jobs
Description	Users can Search Jobs directly from the dashboard using keywords and filters.
Stakeholder	User

Logout

FR-13	Logout
Description	Enables both Admin and users to securely exit the system.
Stakeholder	Admin,User

2.2 Non-Functional Requirements

Non-functional requirements describe how the system operates instead of what it does. These include aspects like performance, security, usability, availability, and maintainability.

ID	Requirement	Description	Example
NFR-01	Usability	The system should be user-friendly and easy to navigate for all types of users.	Users can easily generate a portfolio in under 10 minutes.
NFR-02	Performance	The system should respond within 3 seconds for all user operations.	Clicking "Generate Portfolio" loads the preview in less than 3 seconds.
NFR-03	Scalability	The system should support a growing number of users and portfolios efficiently.	Can handle 1000+ simultaneous portfolio generations without crashing.
NFR-04	Reliability	The system should be available 99% of the time without unexpected crashes.	The hosting server remains online at all times, except during planned maintenance.
NFR-05	Security	Users' personal and portfolio data must be securely stored and transmitted.	Passwords are encrypted; Firebase restricts unauthorized data access.
NFR-06	Compatibility	The system should work across modern browsers (Chrome, Firefox, Edge).	Portfolio renders identically on Chrome and Firefox.
NFR-07	Maintainability	The codebase should be structured and documented to support future updates.	Future developers can easily modify the Job notification feature.
NFR-08	Portability	The application should be accessible on desktop and mobile browsers.	The portfolio form and preview adapt to both laptop and smartphone screens.

2.3 Software Requirements

The software stack required for developing and running the Portfolio Website Builder is as follows:

Component	Details
Operating System	Windows 8 or later (Windows 10/11 recommended)
Frontend	HTML, CSS, Bootstrap, JavaScript
Backend Framework	Firebase (JAVASCRIPT)
Database	Firebase Realtime Database (for user/system data), Firebase Firestore (for dynamic metadata)
Code Editor / IDE	Visual Studio Code (VS Code)

2.4 Hardware Requirements

The hardware specifications needed to run and test the system are:

Component	Minimum Requirement	Recommended
Processor	Intel Pentium 4 or later	Dual-core processor or higher
RAM	2 GB	4 GB or more
Storage	100 GB available hard disk space	SSD preferred for better performance
Display	13” Monitor with 1366×768 resolution	Full HD Monitor
Internet	Required (for Firebase, GitHub, deployment)	High-speed internet connection
Smartphone	Android 8.0+ / iOS 12+ (for responsive UI testing)	Latest Android/iOS version with Chrome/Safari

2.5 Dependability Requirements

This section outlines the key dependability attributes necessary for the Portfolio Website Builder project.

Reliability Requirements

ID	Title	Description	Stakeholder
RR-01	System Reliability	Data like user details and Job preferences must be stored securely and persistently.	Admin,User

Availability Requirements

ID	Title	Description	Stakeholder
AR-02	24/7 Availability	System should be online and operational at all times with minimal downtime.	Admin,User

Maintenance Requirements

ID	Title	Description	Stakeholder
MR-01	System Maintenance	Developers should easily identify and fix bugs or issues when features fail (e.g., search, alerts).	Developer, Admin

Supportability Requirements

ID	Title	Description	Stakeholder
SR-01	User Documentation	Users should have manuals, FAQs, and guides for tasks like portfolio creation and Job search.	Admin,User

Access Requirements

ID	Title	Description	Stakeholder
ACR-01	Authenticated Access	Only registered users can access and manage their portfolio; Admins manage content access.	Admin,User

Ease of Use Requirements

ID	Title	Description	Stakeholder
EUR-01	User-Friendly UI	Interface should be intuitive, with easy navigation and minimal learning curve.	User

Understandability and Politeness Requirements

ID	Title	Description	Stakeholder
UPR-01	Clear Instructions	Platform should offer clear prompts and feedback to guide users.	User

Accessibility Requirements

ID	Title	Description	Stakeholder
ACY-01	Device Compatibility	System must work smoothly on desktop, tablet, and smartphone.	User

Look and Feel Requirements (Style Requirements)

ID	Title	Description	Stakeholder
STR-01	Frontend Technologies	Use HTML, CSS, JavaScript, and ReactJS for interactive and consistent styling.	Developer

Legal Requirements (Standard Requirements)

ID	Title	Description	Stakeholder
LR-01	Compliance with Standards	The system should comply with applicable web development and data protection standards.	Developer, Admin

CHAPTER 3: SYSTEM ANALYSIS

3.1 USE CASE DIAGRAM

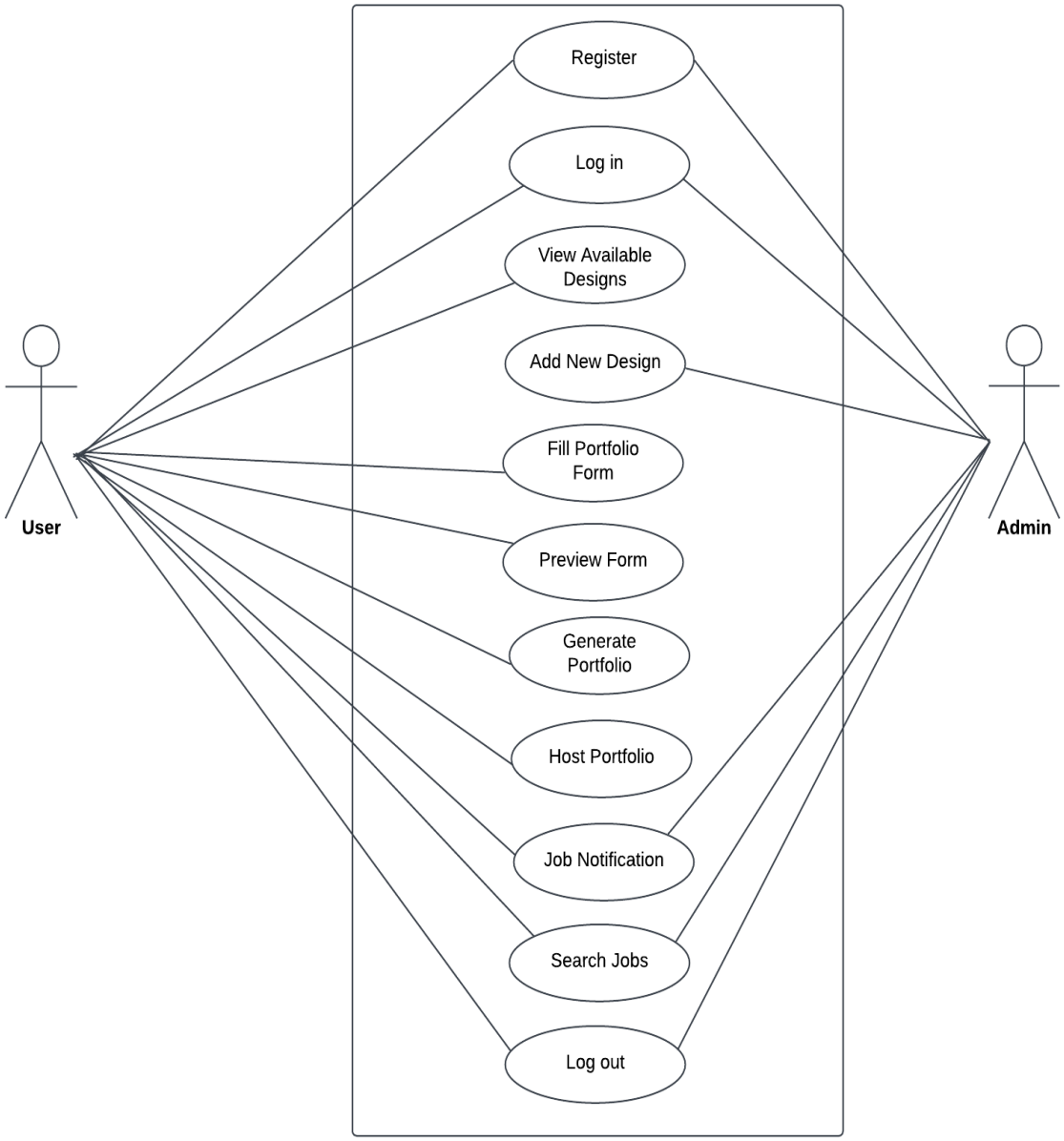


Figure 3: Use Case Diagram

3.2 USE CASE DESCRIPTION

3.2.1 Registration

Both users and Admins must register to use the system. After signing up, they can Login to access their respective dashboards and features.

Use Case Name	Register
Goal	Allow users and Admins to create an account in the system
Preconditions	The user/Admin is not logged in and has not registered yet
Primary Actor	User
Secondary Actor	Admin
Trigger	User/Admin clicks on "Register" button
Description / Main Success Scenario	1. User/Admin navigates to the registration page 2. User/Admin fills out the registration form with valid information 3. System validates the data 4. System stores the user/Admin credentials 5. System confirms successful registration
Post Condition	User/Admin account is created and ready for login
Alternative Flow	If any input is invalid, the system shows an error message and prompts to re-enter data

3.2.2 Login

Registered users and Admins can Login using their email and password. Upon successful login, they are redirected to their dashboard.

Use Case Name	Login
Goal	Allow registered users or Admins to access the system
Preconditions	User/Admin must be registered in the system
Primary Actor	User
Secondary Actor	Admin
Trigger	User/Admin clicks on the "Login" button
Description / Main Success Scenario	1. User/Admin navigates to the login page 2. Enters valid credentials (email and password) 3. System authenticates the credentials 4. If valid, user/Admin is logged in successfully
Post Condition	User/Admin is redirected to the dashboard
Alternative Flow	If credentials are invalid, an error message is shown and retry is allowed

3.2.3 View Available Designs

After logging in, users can browse a collection of available portfolio design templates. These designs help users choose a layout before filling out their portfolio information.

Use Case Name	View Available Designs
Goal	Display all available portfolio design templates
Preconditions	User/Admin must be logged in
Primary Actor	User
Secondary Actor	Admin
Trigger	Clicks on the "View Designs" or similar option
Description / Main Success Scenario	1. User/Admin selects “View Available Designs” 2. System retrieves list of design templates from database 3. Displays the templates in a user-friendly layout
Post Condition	User/Admin sees available portfolio templates
Alternative Flow	If no templates are found, system shows “No designs available”

3.2.4 Add New Design

Admins can upload new portfolio templates to the system, making them available for all users.

Use Case Name	Add New Design
Goal	Allow Admin to upload or create a new portfolio template/design
Preconditions	Admin must be logged in
Primary Actor	Admin
Secondary Actor	—
Trigger	Admin clicks on “Add New Design”
Description / Main Success Scenario	1. Admin selects the option to add a new design 2. Fills out design details (name, preview, description) 3. Uploads required assets (if any) 4. System saves the design template to database 5. Confirmation message is displayed
Post Condition	New design becomes available to users
Alternative Flow	If fields are missing or invalid, error is shown; Admin must fix and resubmit

3.2.5 Fill Portfolio Form

After logging in, users can fill out a portfolio form with their personal, educational, and project details.

Use Case Name	Fill Portfolio Form
Goal	Let user input personal and professional details for portfolio
Preconditions	User must be logged in
Primary Actor	User
Secondary Actor	—
Trigger	User selects a design and chooses to fill the form
Description / Main Success Scenario	1. User selects “Fill Portfolio Form” 2. Form with relevant fields is displayed 3. User enters data (name, skills, education, projects, etc.) 4. System validates and saves the input 5. System confirms data is saved
Post Condition	Portfolio form is saved for that user
Alternative Flow	If mandatory fields are missing or input is invalid, sh and prompt fix

3.2.6 Preview Form

Users can preview their portfolio using the selected template before finalizing it.

Use Case Name	Preview Form
Goal	Allow user to preview their portfolio before generating it
Preconditions	User must be logged in and must have filled the portfolio form
Primary Actor	User
Secondary Actor	—
Trigger	User clicks on “Preview Form” button
Description / Main Success Scenario	1. User selects the “Preview Form” option 2. System fetches the filled data 3. System displays a preview using selected design 4. User reviews the layout and content
Post Condition	User sees an accurate preview of the portfolio
Alternative Flow	If data is incomplete, system shows a warning message and asks for completion

3.2.7 Generate Portfolio

Once the form is complete, users can generate a final version of their portfolio based on the chosen design.

Use Case Name	Generate Portfolio
Goal	Generate a final version of the portfolio based on filled data and design
Preconditions	User must be logged in and previewed the portfolio
Primary Actor	User
Secondary Actor	—
Trigger	User clicks on “Generate Portfolio” button
Description / Main Success Scenario	1. User confirms preview is correct 2. Clicks on “Generate Portfolio” 3. System processes the data and selected design 4. Final HTML/CSS-based portfolio is created 5. System confirms generation successful
Post Condition	User’s portfolio is ready for hosting
Alternative Flow	If system fails to generate due to technical error, a retry option is provided

3.2.8 Host Portfolio

After generating the portfolio, users can host it online with a public link .

Use Case Name	Host Portfolio
Goal	Let users publish their generated portfolio to a public URL
Preconditions	Portfolio must be generated successfully
Primary Actor	User
Secondary Actor	—
Trigger	User clicks on “Host Portfolio” button
Description / Main Success Scenario	1. User clicks “Host Portfolio” 2. System generates a unique public link 3. Portfolio is uploaded to hosting server (Firebase or similar) 4. User is shown the hosted portfolio link
Post Condition	Portfolio is live and publicly accessible
Alternative Flow	If hosting fails, system shows error and provides retry option

3.2.9 Job Notification

The system sends Job notifications to users based on their portfolio skills or preferences.

Use Case Name	Job Notification
Goal	Notify users of Job openings relevant to their profile
Preconditions	User must have an active portfolio and be logged in
Primary Actor	System
Secondary Actor	User
Trigger	New Job is added by Admin or fetched via API
Description / Main Success Scenario	1. System checks for new Job opportunities 2. Matches Job with user profile/skills 3. Sends notification to relevant users 4. User views Job notification in dashboard
Post Condition	User gets notified of matching Job posts
Alternative Flow	If no relevant Job found, no notification is sent

3.2.10 Search Jobs

Users can search for Jobs using filters like keyword or location to find relevant opportunities.

Use Case Name	Search Jobs
Goal	Allow users to search for Job postings relevant to their portfolio or interests
Preconditions	User must be logged in
Primary Actor	User
Secondary Actor	—
Trigger	User navigates to the “Job Search” section and enters search criteria
Description / Main Success Scenario	1. User accesses the Job Search section 2. Inputs keywords, location, or filters 3. System searches Job postings based on input 4. Displays relevant Job listings 5. User views and applies to desired Jobs
Post Condition	Matching Jobs are displayed to the user
Alternative Flow	If no Jobs match, system shows “No results found”

3.2.11 Logout

Users and Admins can securely Logout of the system to end their session.

Use Case Name	Logout
Goal	Securely Logout the user from the system
Preconditions	User must be logged in
Primary Actor	User
Secondary Actor	—
Trigger	User clicks on the “Logout” button
Description / Main Success Scenario	1. User clicks “Logout” 2. System clears session/token 3. Redirects user to login or landing page
Post Condition	User is logged out and session is ended
Alternative Flow	If logout fails, show an error and offer retry option

3.3 ACTIVITY DIAGRAM

I created activity diagrams using the system's use cases as a guide. Each of the system's essential functions are shown in these diagrams.

3.3.1 Registration

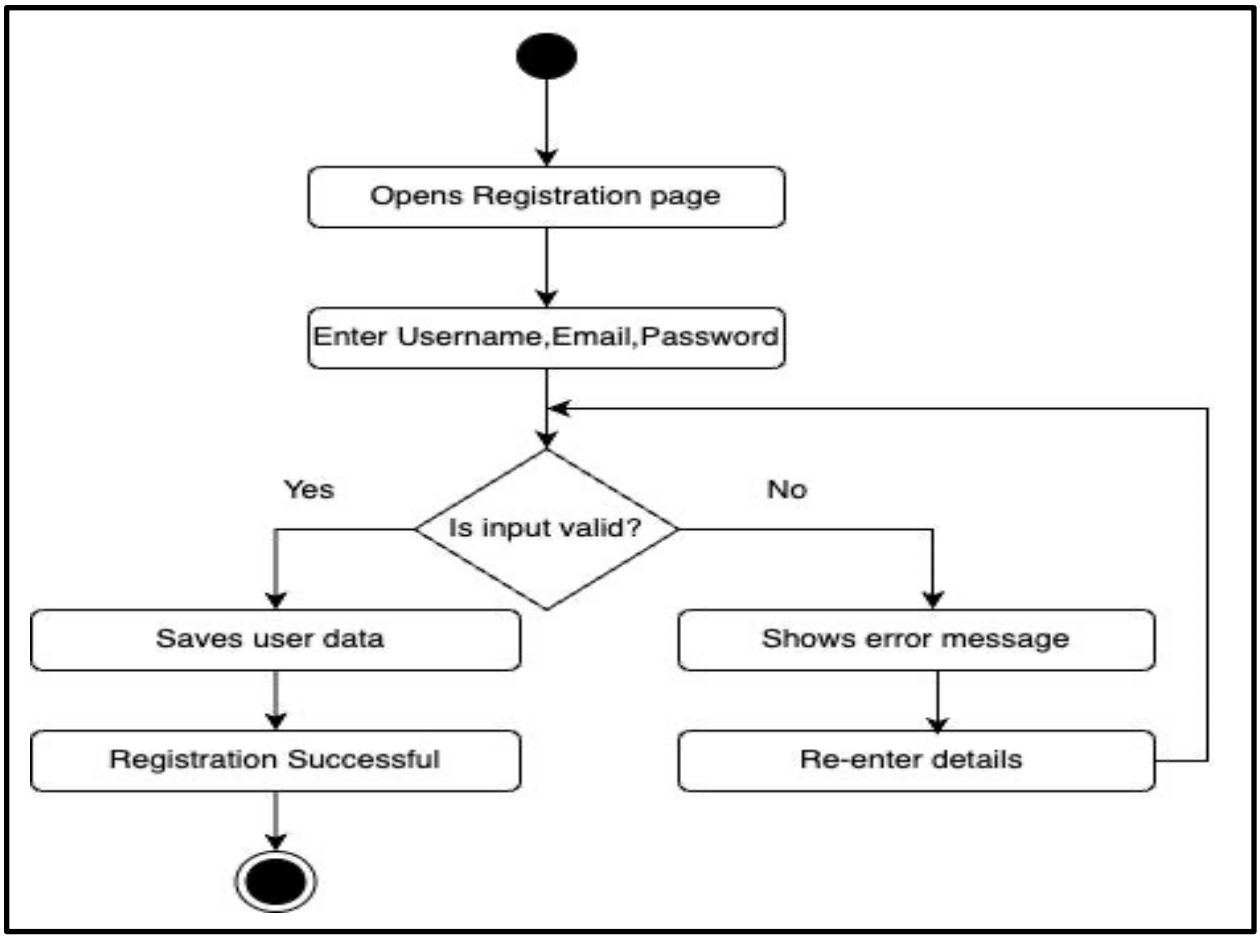


Figure 4: Activity diagram of Registration

3.3.2 Login

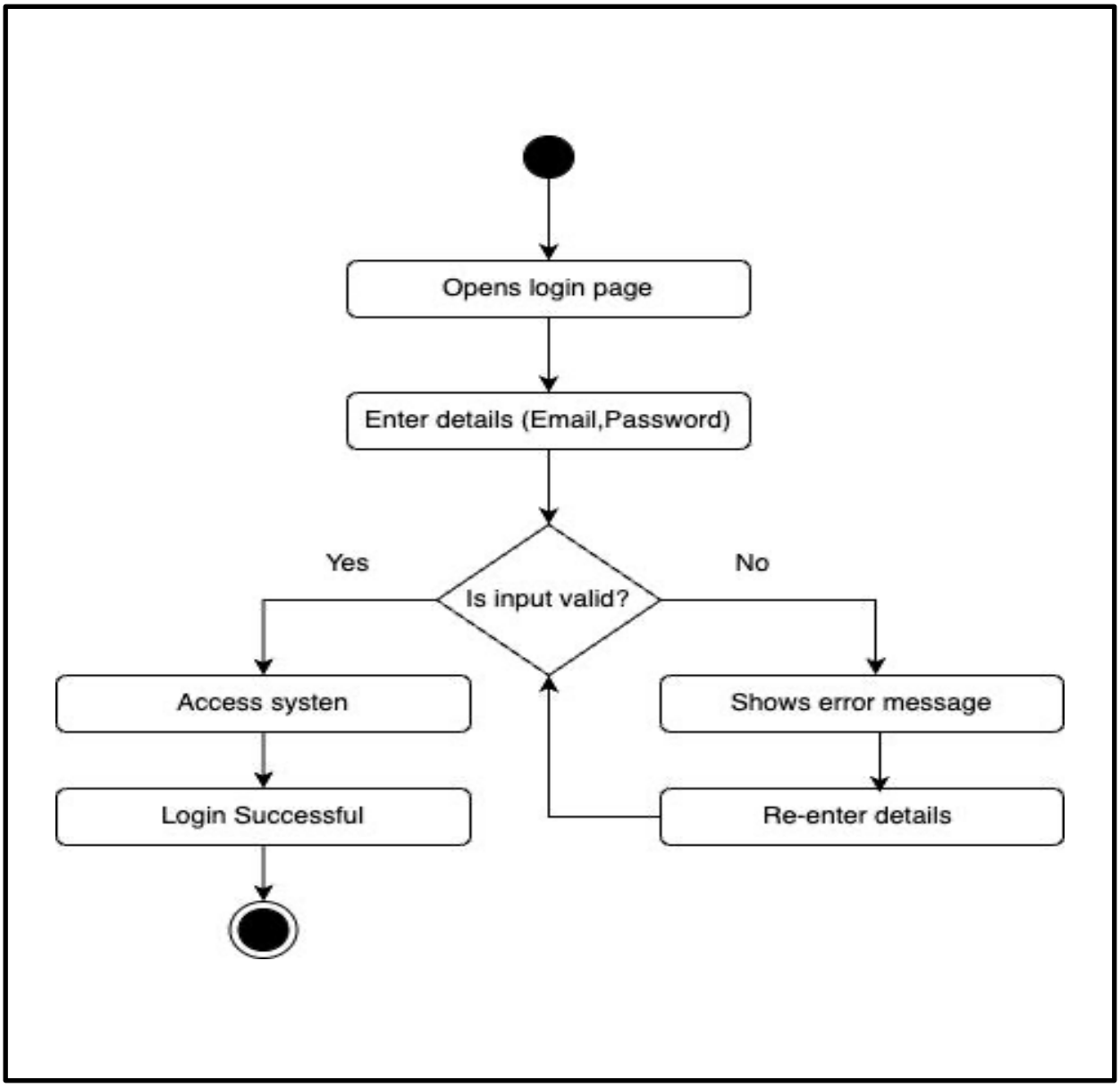


Figure 5: Activity diagram of Login

3.3.3 Select Available design

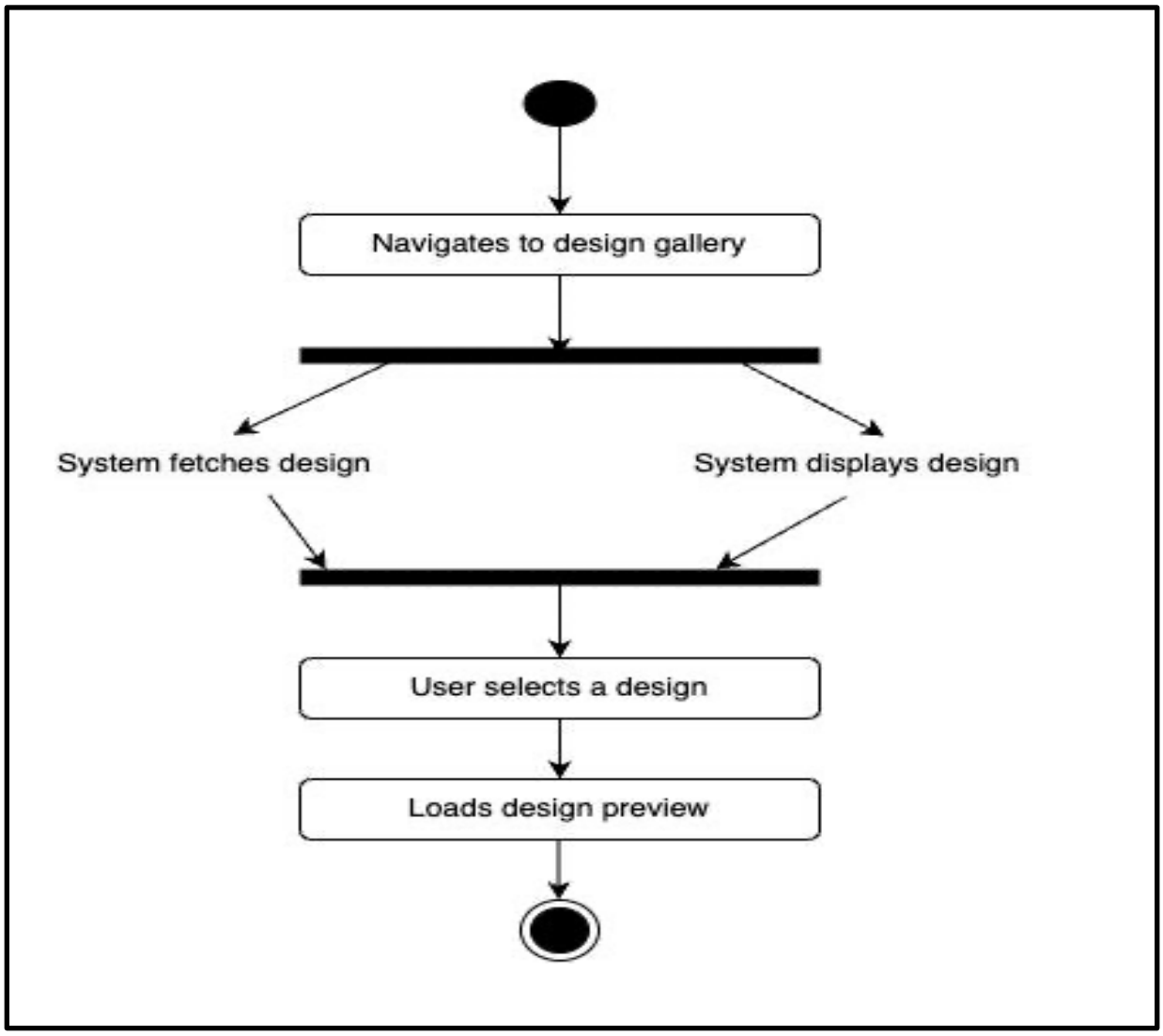


Figure 6: Activity diagram of Select Available design

3.3.4 Fill Portfolio form

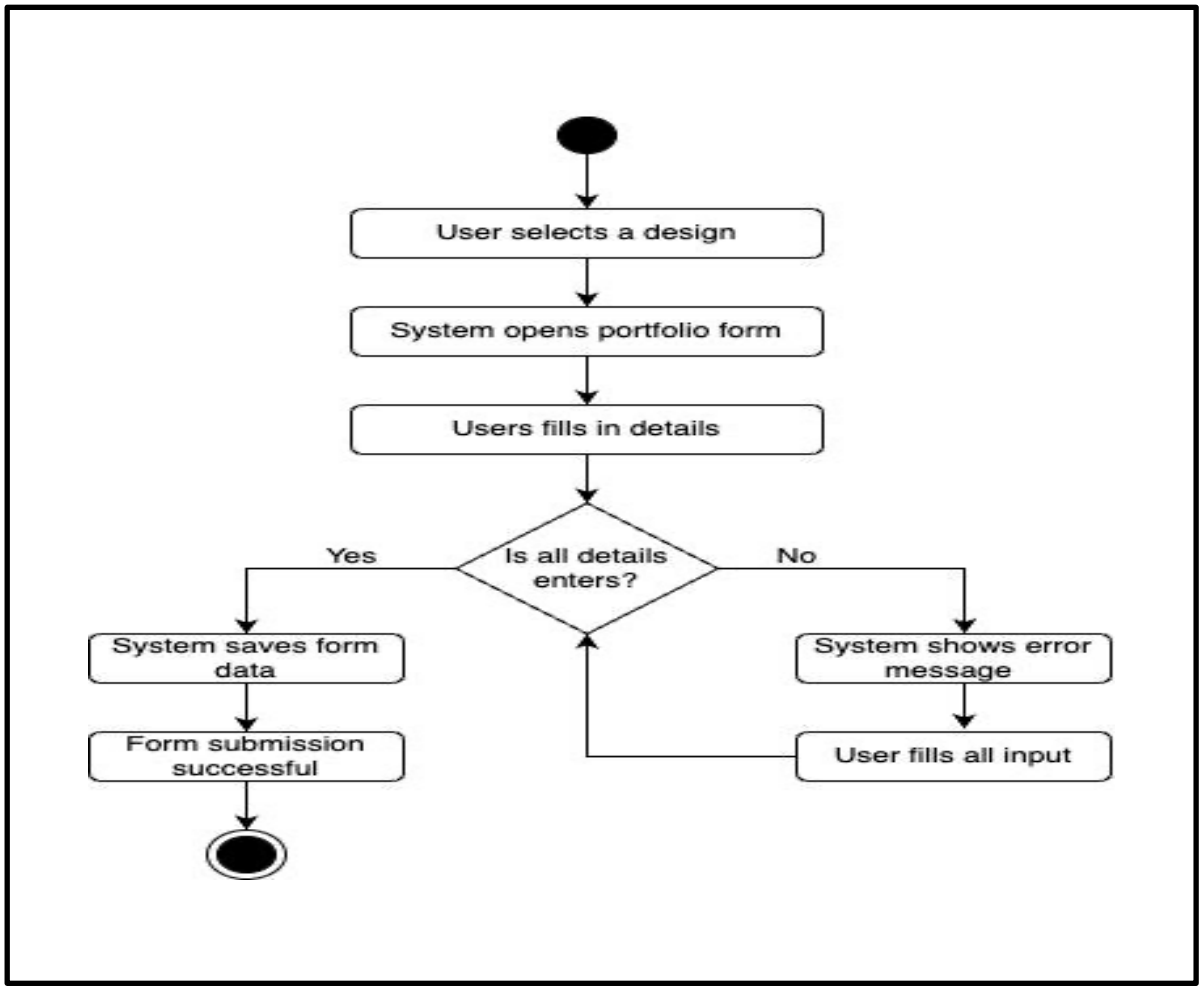


Figure 7: Activity diagram of Fill Portfolio form

3.3.5 Preview form

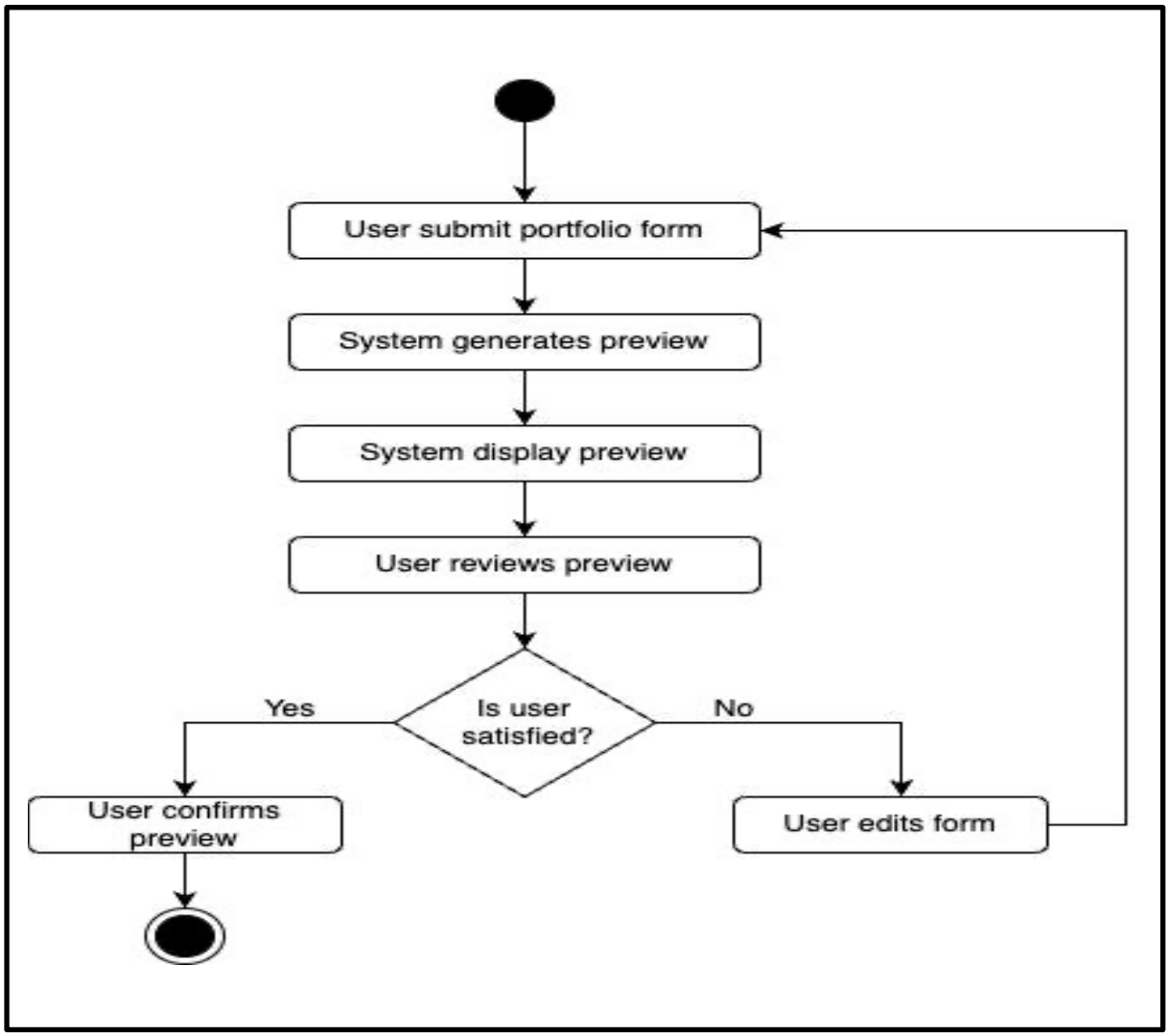


Figure 8: Activity diagram of Preview form

3.3.6 Generate Portfolio

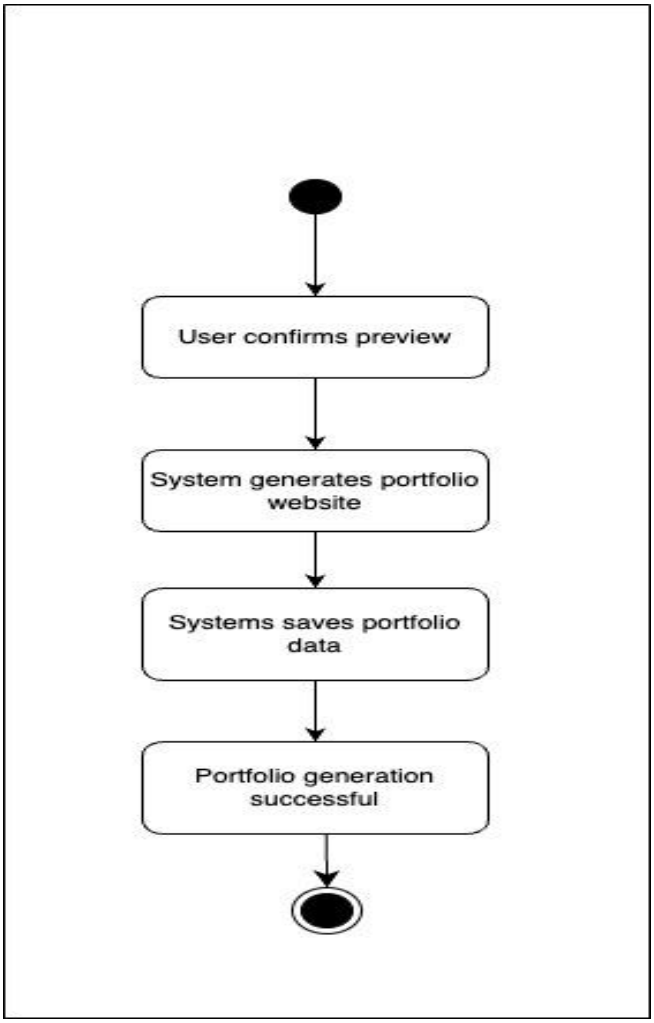


Figure 9: Activity diagram of Generate Portfolio

3.3.7 Host Portfolio

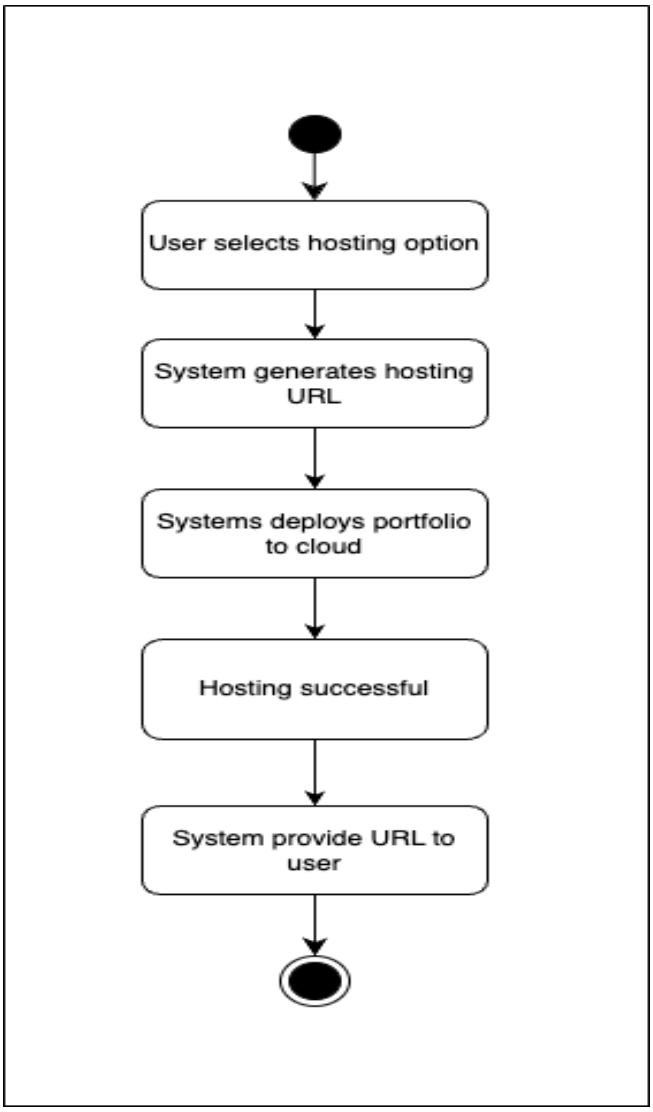


Figure 10: Activity diagram of Host Portfolio

3.3.8 Job Notification

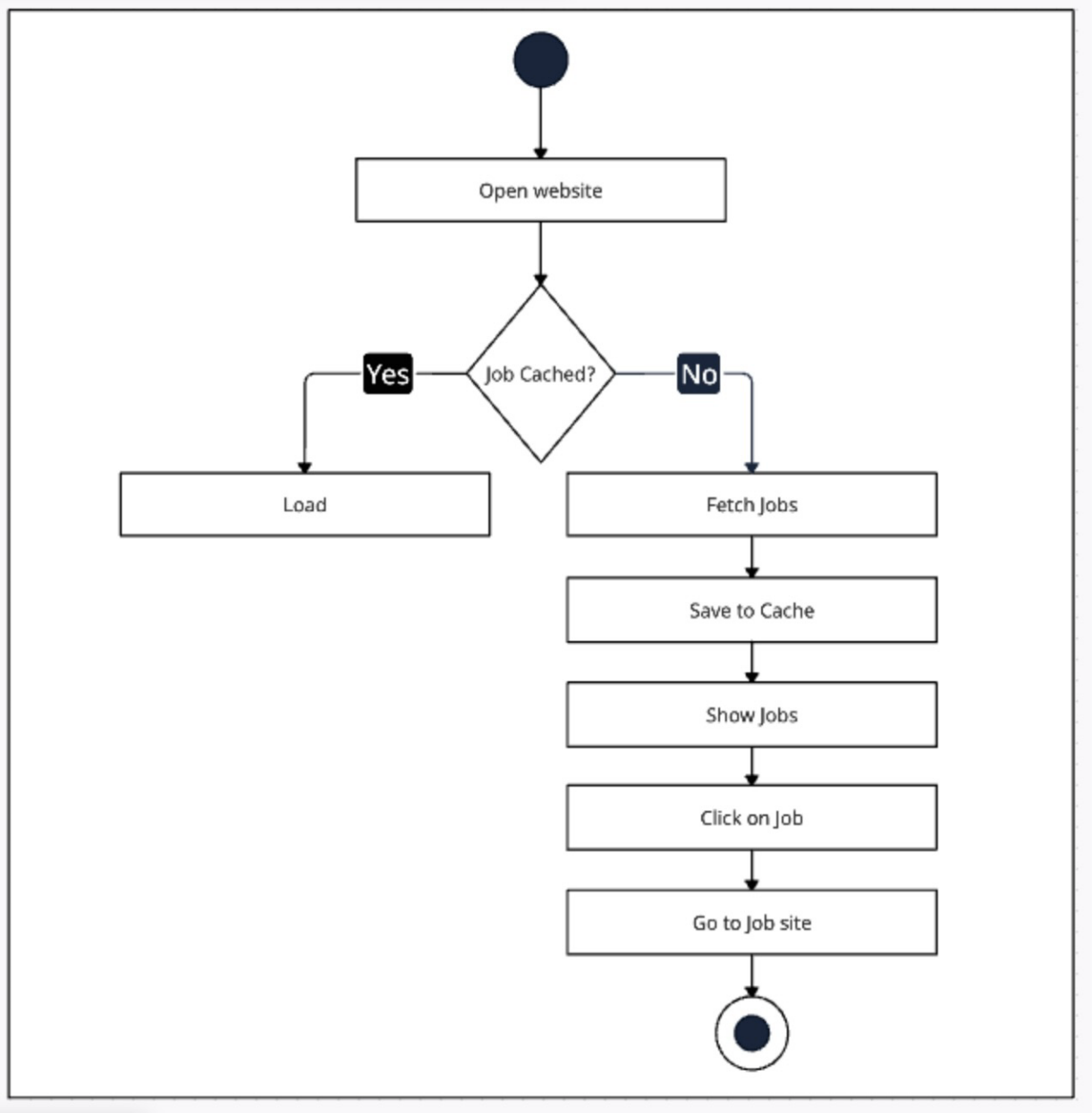


Figure 11: Activity diagram of Job Notification

3.3.9 Search Jobs

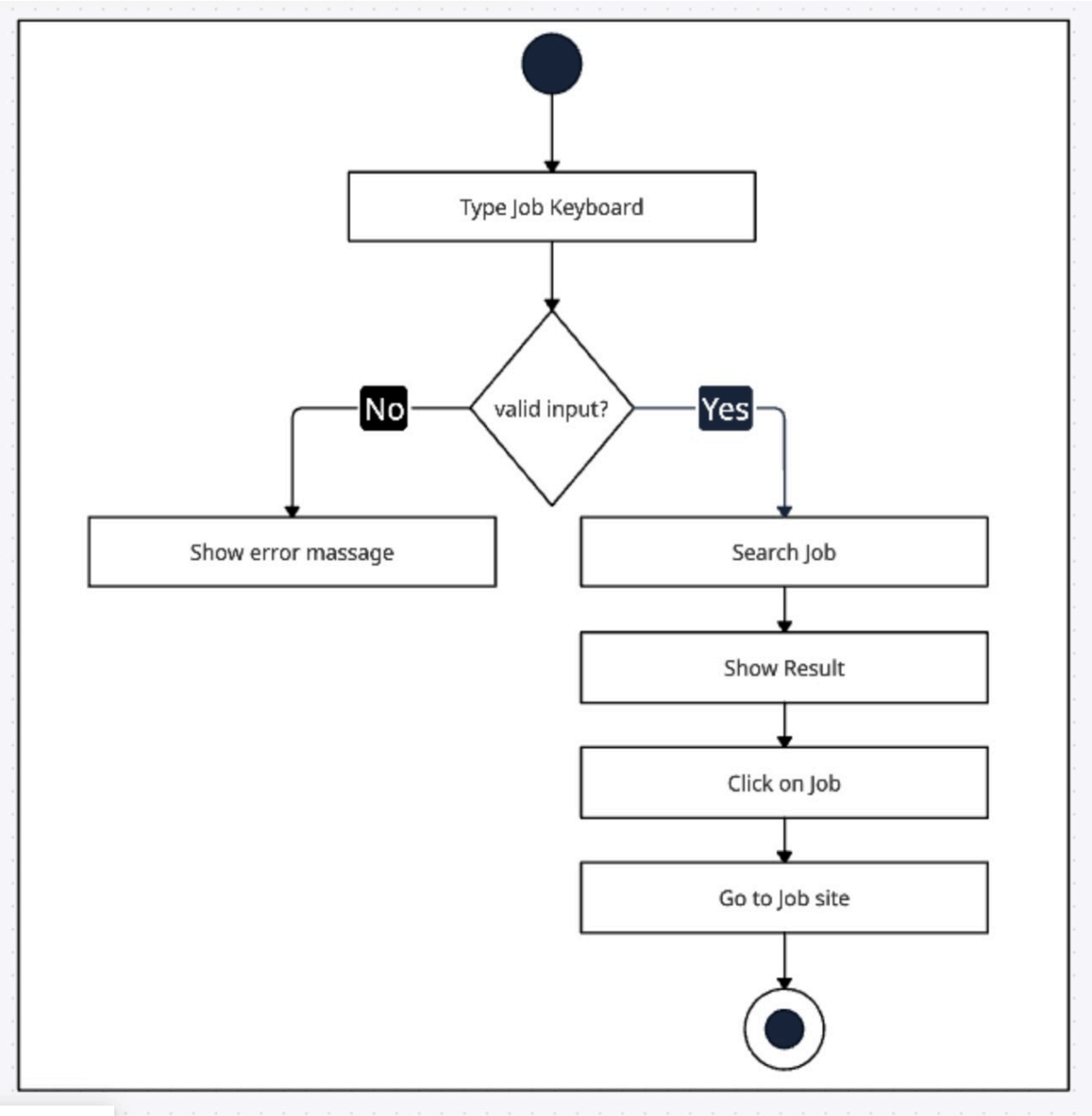


Figure 12: Activity diagram of Search Jobs

3.3.10 Logout

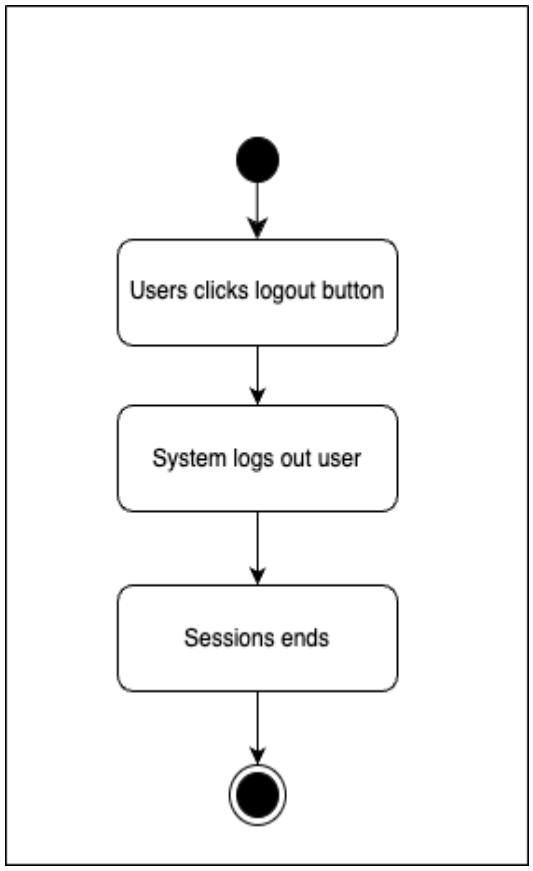


Figure 13: Activity diagram of Logout

3.4 SEQUENCE DIAGRAM

The sequence diagram shows how the system components interact step-by-step to complete each process.

3.4.1 Registration

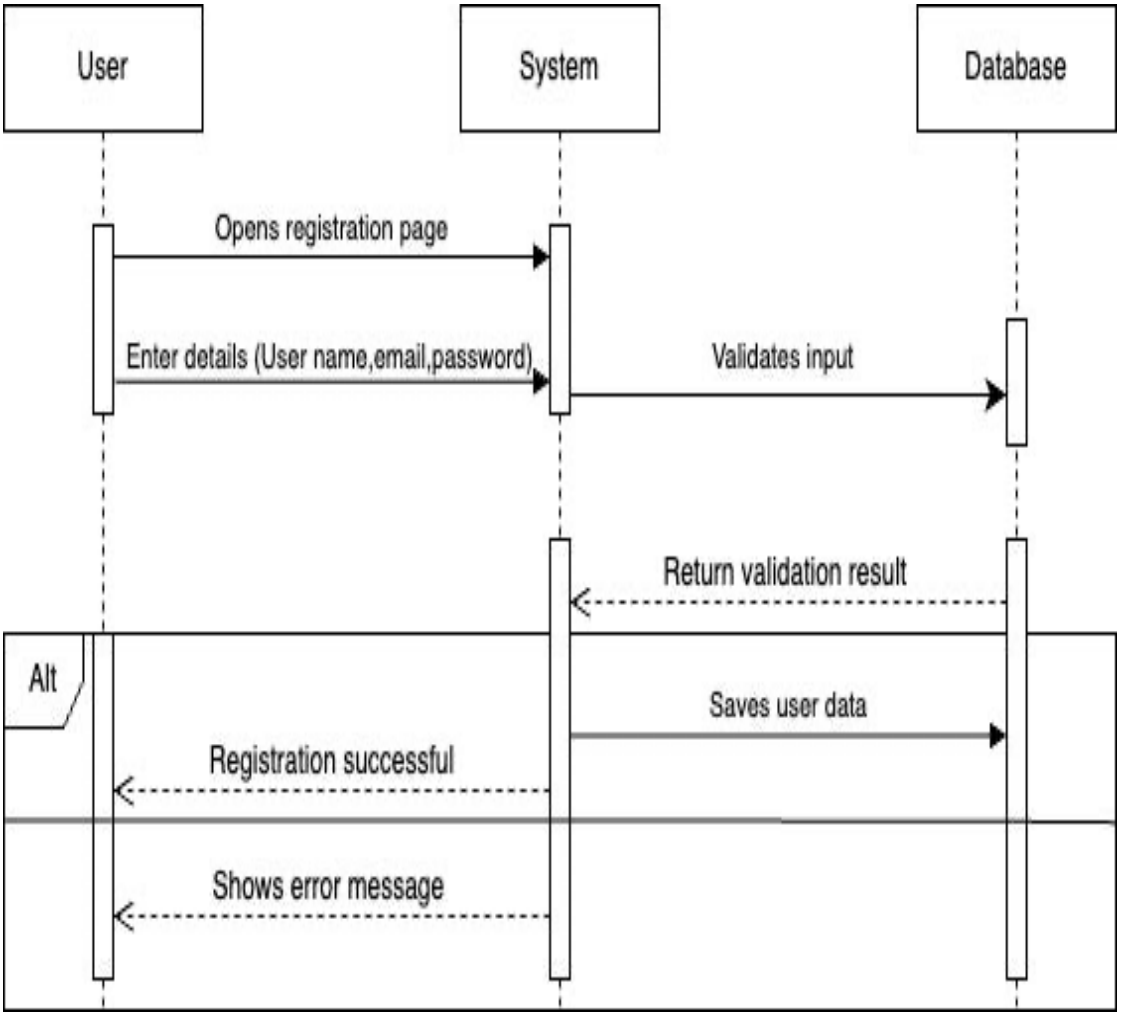


Figure 14: Sequence diagram for Registration

3.4.2 Login

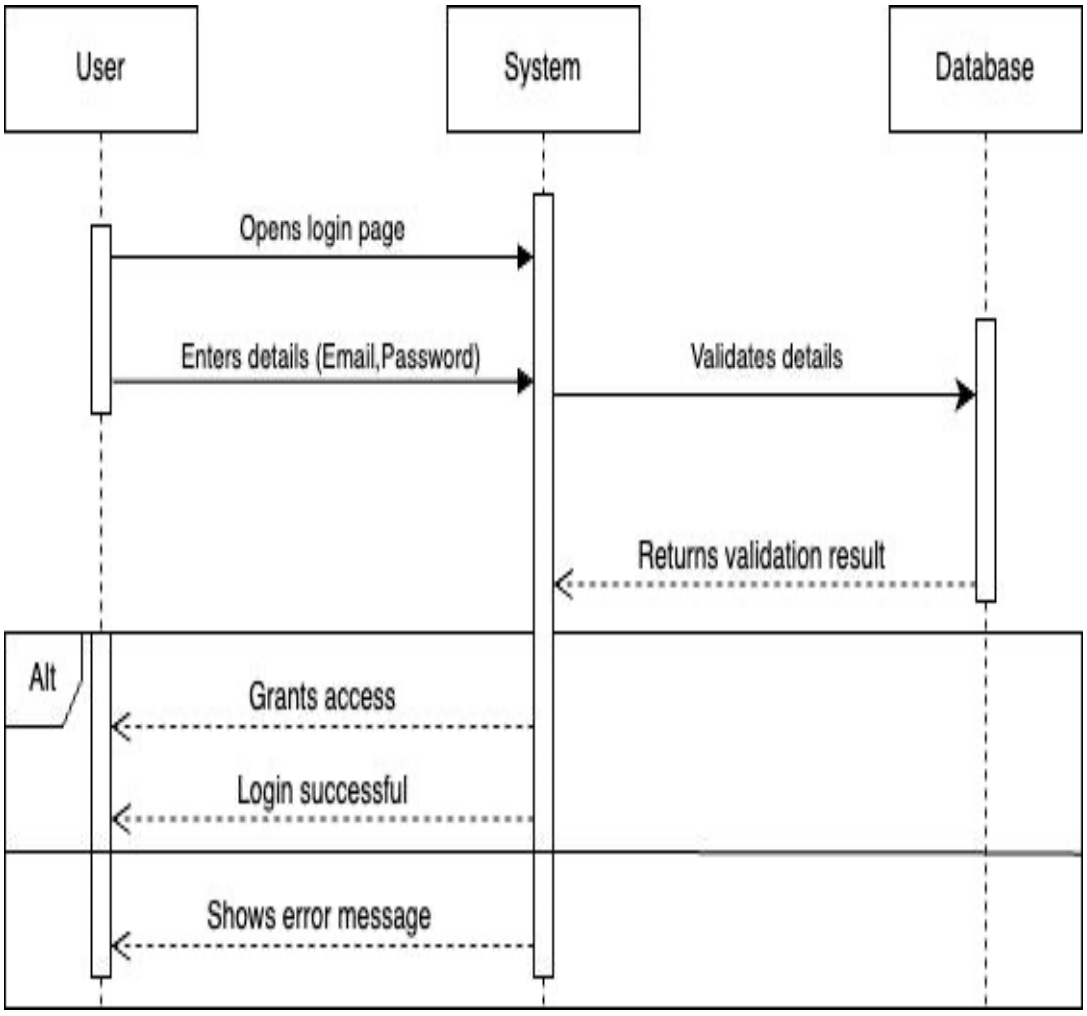


Figure 15: Sequence diagram for Login

3.4.3 Select Available design

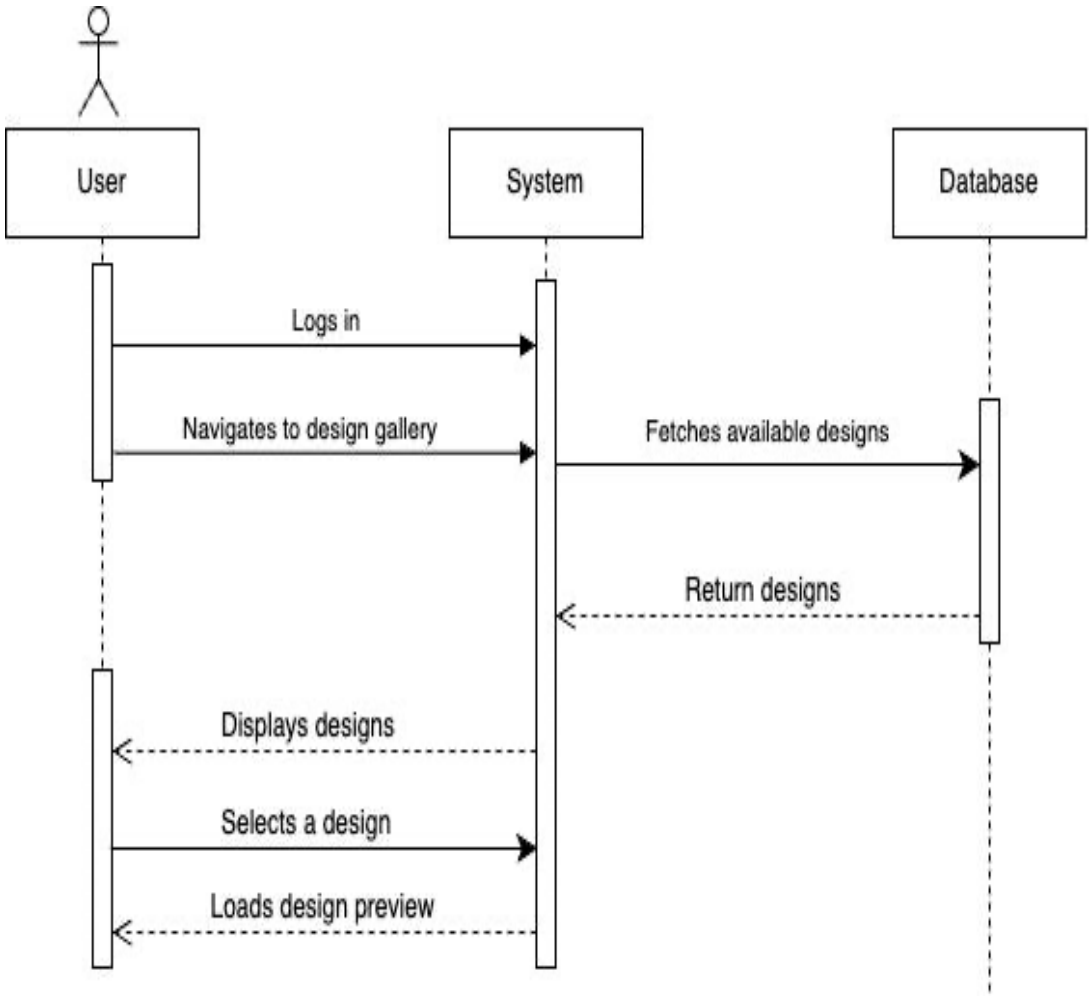


Figure 16: Sequence diagram for Select Available design

3.4.4 Fill Portfolio form

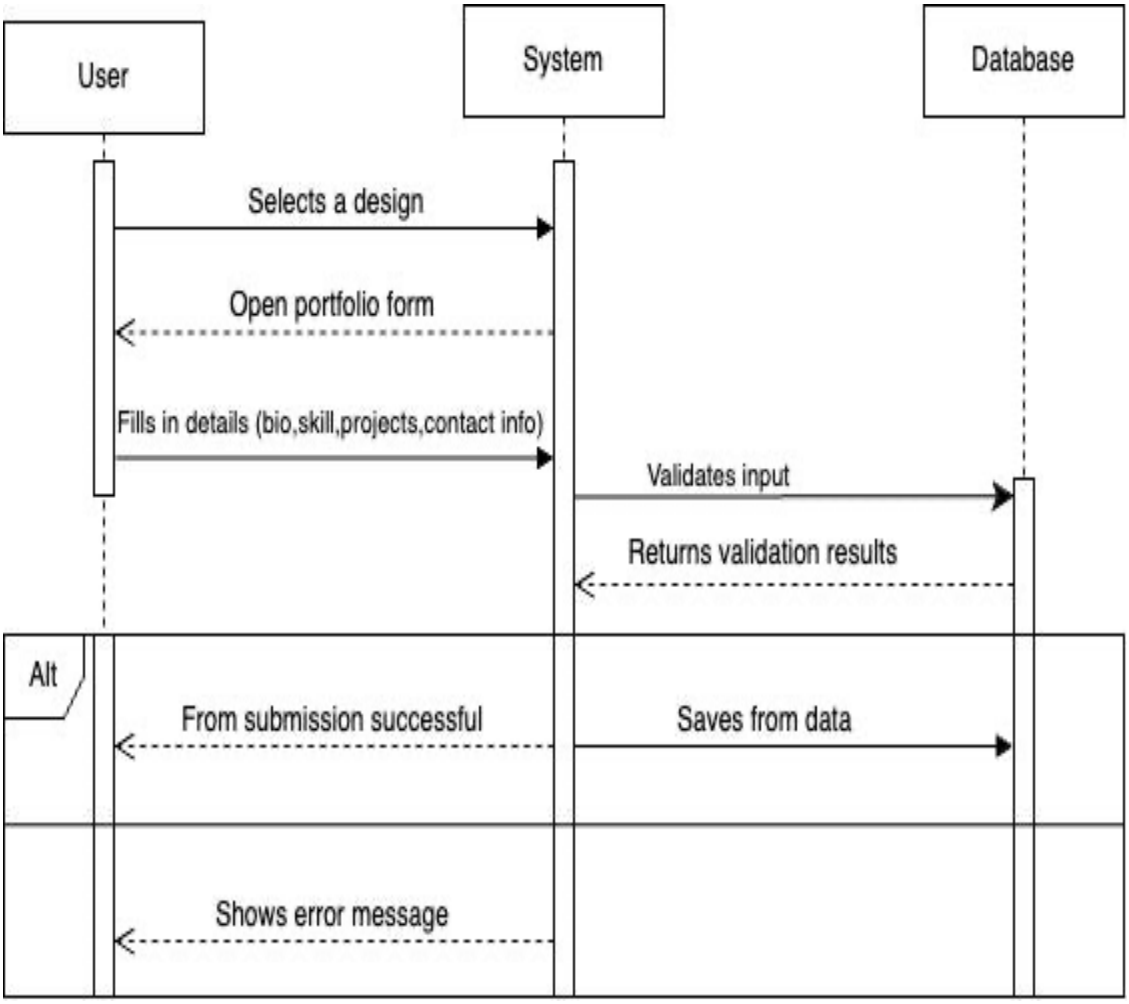


Figure 17: Sequence diagram for Fill Portfolio form

3.4.5 Preview form

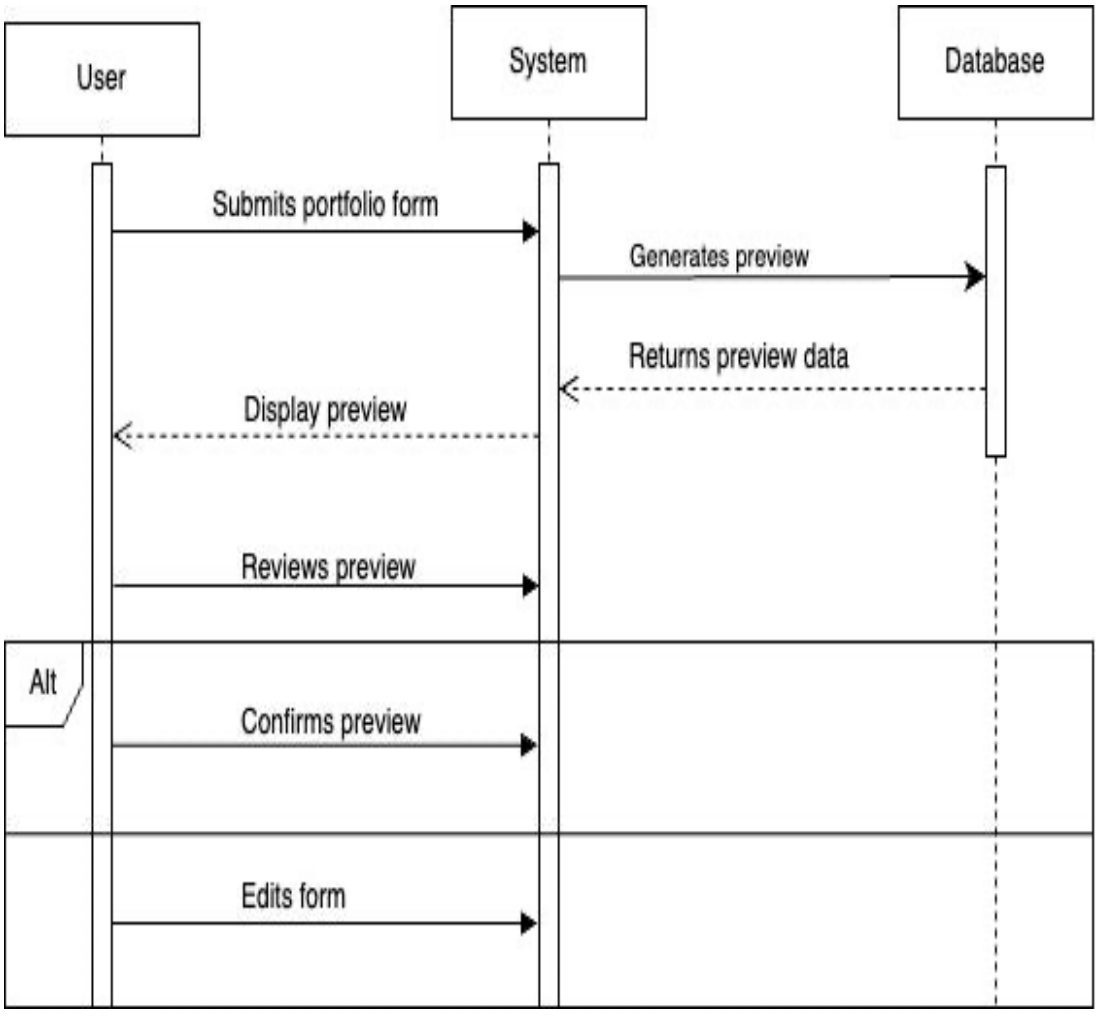


Figure 18: Sequence diagram for Preview form

3.4.6 Generate Portfolio

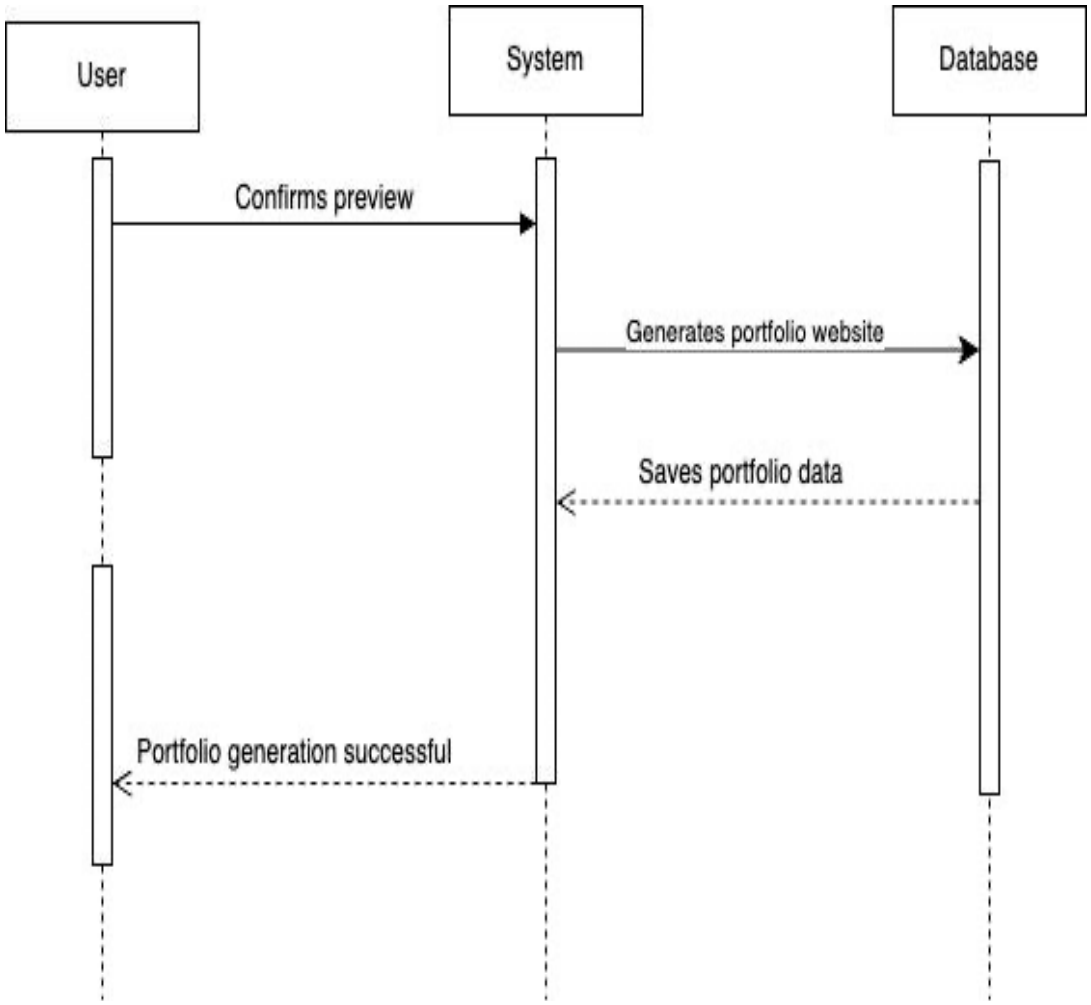


Figure 19: Sequence diagram for Generate Portfolio

3.4.7 Host Portfolio

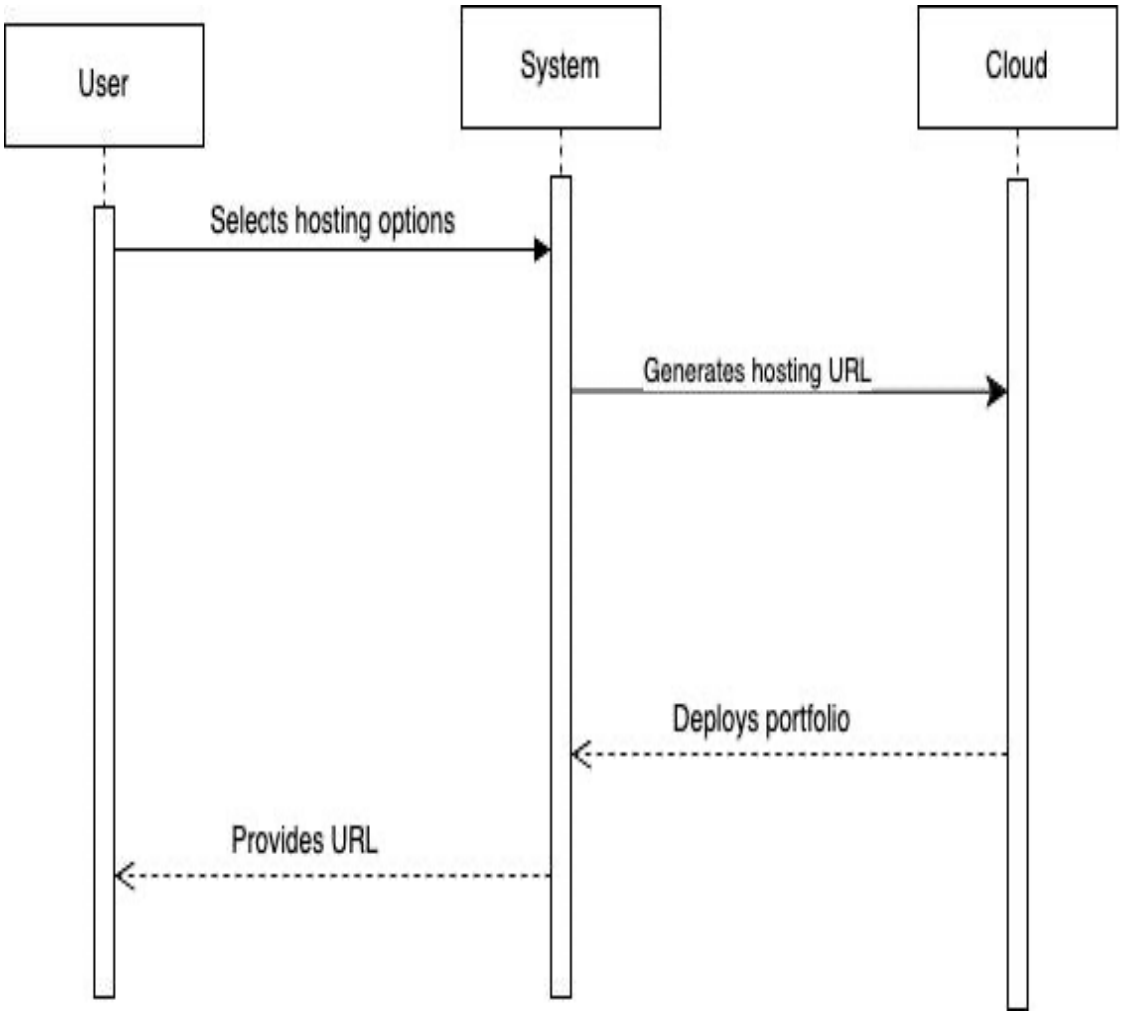


Figure 20: Sequence diagram for Host Portfolio

3.4.8 Host Job Notification

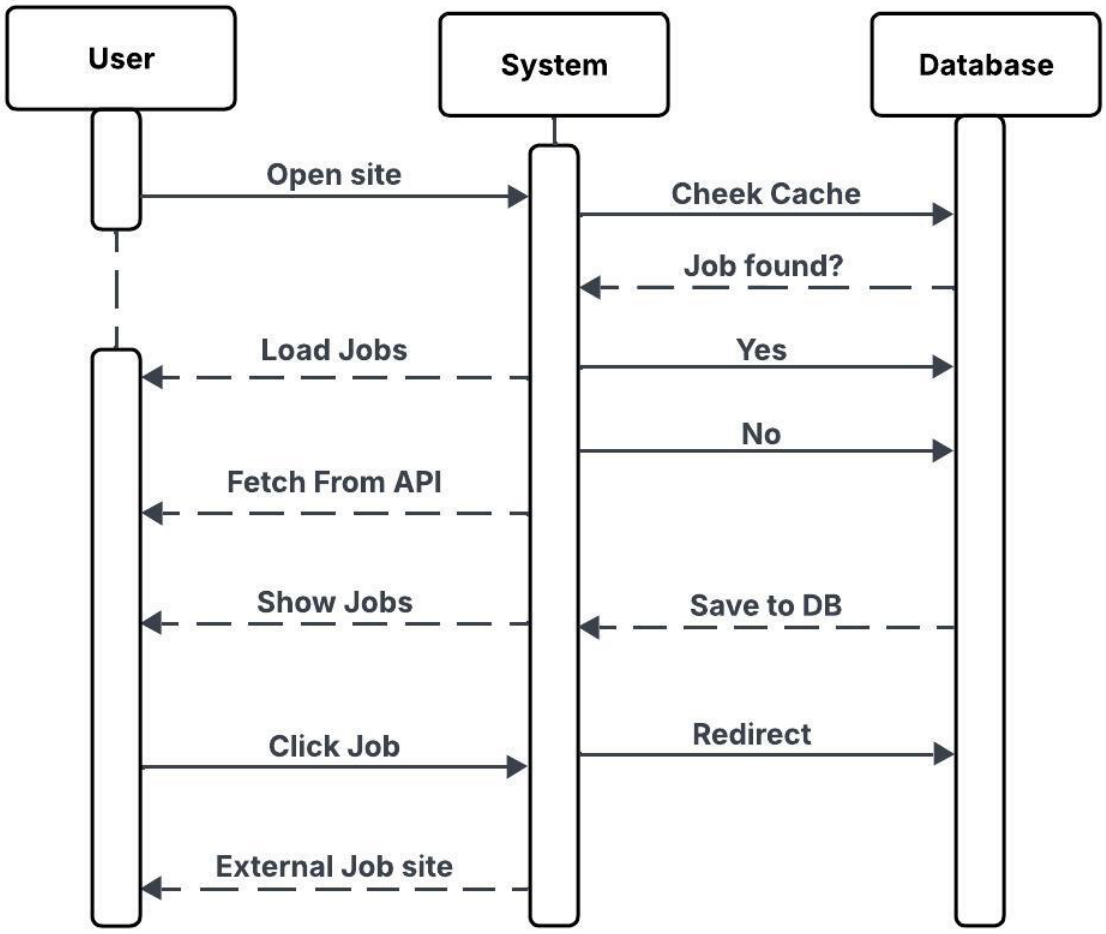


Figure 21: Sequence diagram for Job Notification

3.4.9 Search Jobs

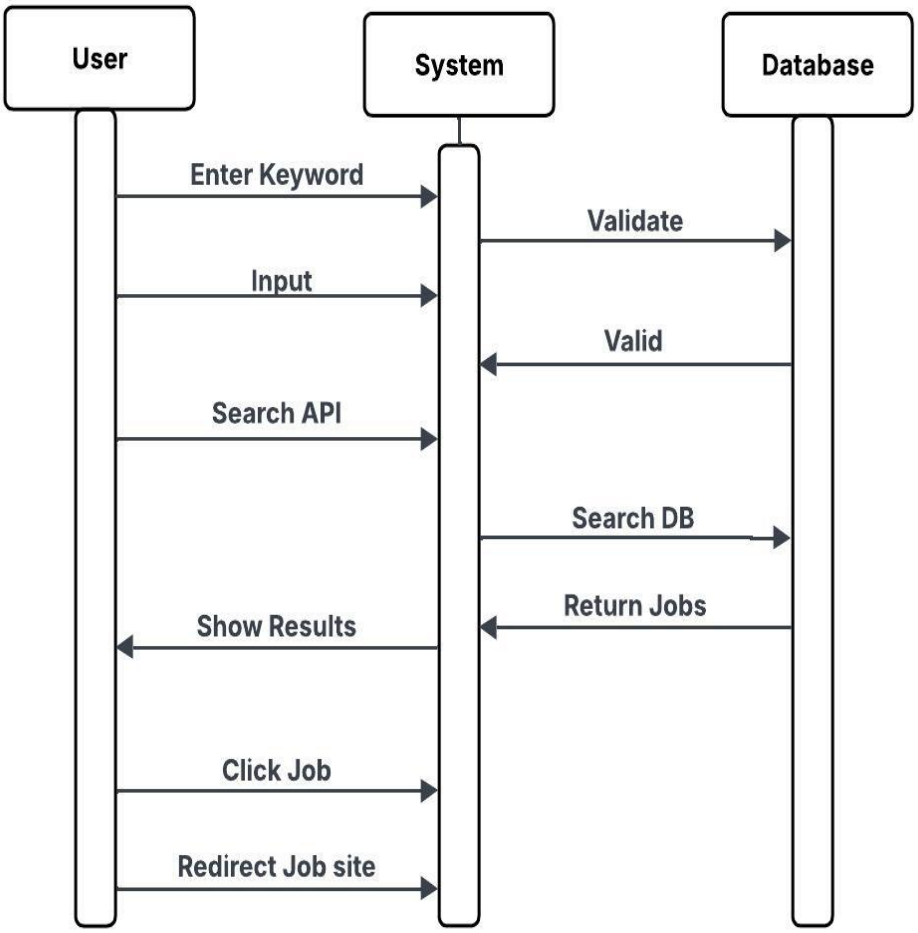


Figure 22: Sequence diagram for Search Jobs

3.4.10 Logout

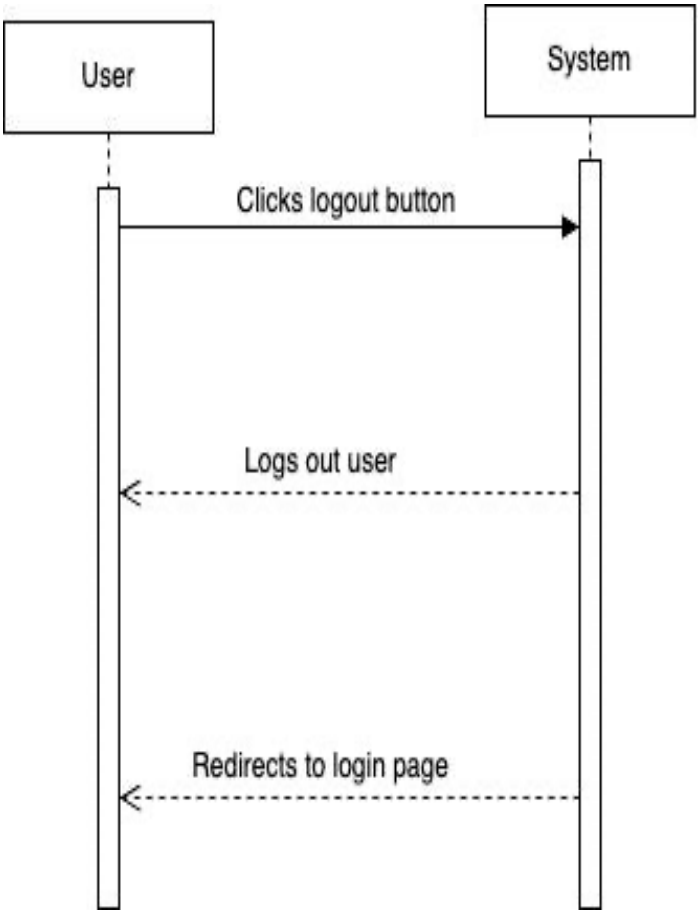


Figure 23: Sequence diagram for Logout

3.5 Data Flow Diagram

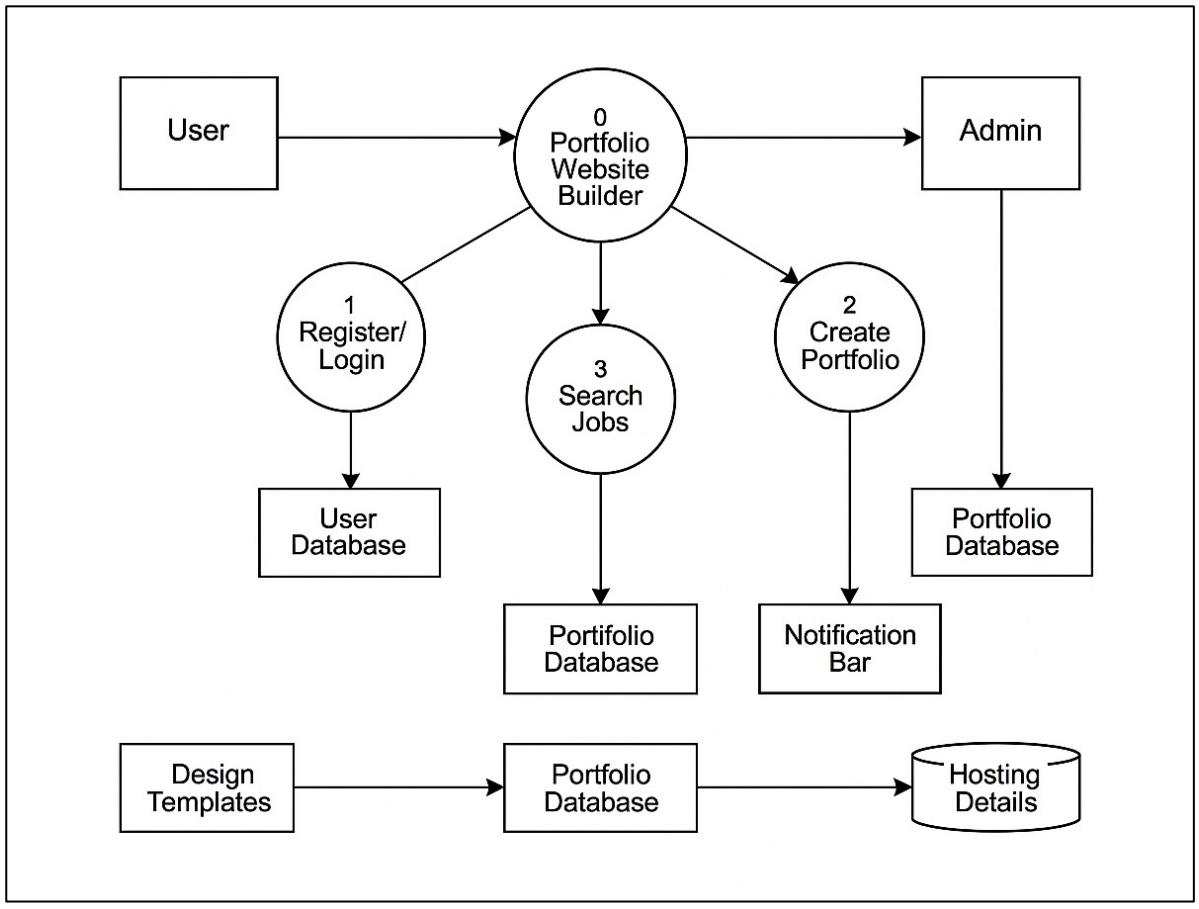


Figure 24: DFD Diagram

3.6 ENTITY RELATIONSHIP DIAGRAM

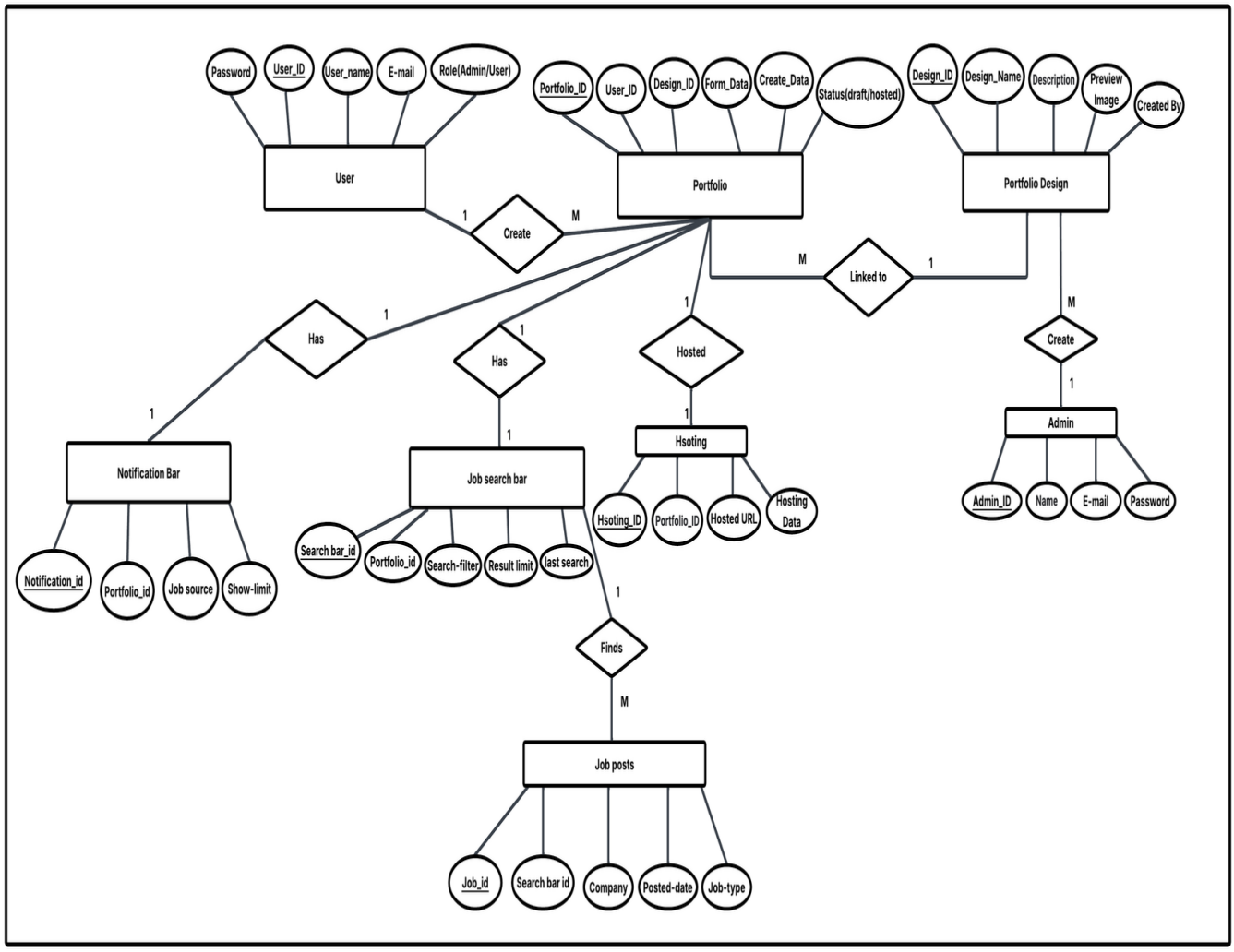


Figure 25: Entity Relationship Diagram

CHAPTER 4: SYSTEM TESTING

4.1 INTRODUCTION TO SYSTEM TESTING:

System Testing is a critical phase that involves validating the entire and fully integrated **Portfolio Website Builder** platform. The main objective is to ensure the system meets all specified requirements and functions correctly in a real-world environment.

Since the **Portfolio Website Builder** includes various interconnected modules—such as user registration, portfolio creation, Job notification, and Job search—system testing evaluates how these components work together as a unified platform. This testing phase simulates real user scenarios to verify both functionality and performance across different devices and browsers, ensuring the end-to-end system behaves as expected.

4.2 Testing Strategies

4.2.1 Test Approach

- The system will undergo **manual testing** to verify the functionality of all core features, including user registration, portfolio creation, Job notifications, and Job search.
- Testing will be guided by **user acceptance criteria**, ensuring that the system meets real-world expectations and provides a smooth, intuitive experience for end users.

4.2.2 Pass /Fail Criteria

- **Component Pass/Fail Criteria** – A component-level test will be marked as *pass* if it meets its defined design and functional requirements; otherwise, it will be marked as fail.
- **Integration Pass/Fail Criteria** – Integration tests will pass if interconnected modules (e.g., Job Alert System, search functionality, portfolio manager) work together according to the system's architectural design; they fail if inconsistencies or communication issues occur.
- **System Pass/Fail Criteria** – The entire system will pass final testing if it meets all **functional** and **non-functional** requirements, including usability, performance, security, and reliability. A failure in any key requirement will result in a test fail.

4.2.3 White Box Testing

Item	Details
Designed By	Samiul Siam
Design Date	21 June 2025
Executed By	Samiul Siam
Executed Date	28 June 2025
Pass/Fail	Pass
Comment	Internal logic, database calls, and conditional flows were tested via unit testing. No critical bugs were found.

Explanation:
White Box Testing focused on the internal structure of the application. Developers performed unit tests on:

- User registration logic
- Firebase Fire store write/read operations
- Job matching algorithm in the notification module
- Hosting URL generation logic
- Form data validation and submission

These tests ensured the internal code paths executed as intended and all logical conditions were covered.

4.2.4 Black Box Testing

Item	Details
Designed By	Samiul Siam
Design Date	21 June 2025
Executed By	Samiul Siam
Executed Date	29 June 2025
Pass/Fail	Pass
Comment	Functional testing through UI; all user-facing features (login, form, Job search) passed under real-world scenarios.

Explanation:
Black Box Testing involved interacting with the system solely through its user interface, simulating the experience of real users. This included:

- Registering/logging in as a user and Admin
- Creating and previewing a portfolio
- Hosting and accessing the portfolio live
- Receiving Job notifications
- Searching Jobs with filters and keywords

No internal code was reviewed, and all functional modules responded as expected under test scenarios.

4.3 Testing schedule

Testing Phase	Time	Owner
Test Plan Creation	1 week	Samiul Siam
Test Specification Creation	1 week	Samiul Siam
Test Specification Team Review	2 weeks	Samiul Siam
Component Testing	2 weeks	Samiul Siam
Integration Testing	1 week	Samiul Siam
System Testing	3 weeks	Samiul Siam

4.4 System Test Cases

Test Case	Designed By	Design Date	Executed By	Executed Date	Pass/Fail	Comment
1. User Registration	Samiul Siam	15 June 2025	Samiul Siam	22 June 2025	Pass	Registration form works correctly with valid input.
2. User Login	Samiul Siam	15 June 2025	Samiul Siam	22 June 2025	Pass	Login redirects to dashboard; error shown for wrong credentials.
3. View Available Designs	Samiul Siam	16 June 2025	Samiul Siam	23 June 2025	Pass	Designs load correctly from database.
4. Fill Portfolio Form	Samiul Siam	16 June 2025	Samiul Siam	23 June 2025	Pass	Form saves data and allows later access.
5. Preview Portfolio	Samiul Siam	17 June 2025	Samiul Siam	24 June 2025	Pass	Preview shows correctly with all form data.
6. Generate Portfolio	Samiul Siam	17 June 2025	Samiul Siam	25 June 2025	Pass	Portfolio generated using selected template and form data.
7. Host Portfolio	Samiul Siam	18 June 2025	Samiul Siam	26 June 2025	Pass	URL assigned, hosted successfully on Firebase.
8. Job Notification	Samiul Siam	18 June 2025	Samiul Siam	27 June 2025	Pass	Notification bar displays correct Job alerts.
9. Job Search	Samiul Siam	19 June 2025	Samiul Siam	27 June 2025	Pass	Filters and keywords show relevant Job results.
10. Admin Login	Samiul Siam	19 June 2025	Samiul Siam	28 June 2025	Pass	Admin dashboard accessible with correct login.
11. Add New Design	Samiul Siam	20 June 2025	Samiul Siam	28 June 2025	Pass	Admin can successfully add and publish designs.
12. Admin Logout	Samiul Siam	20 June 2025	Samiul Siam	28 June 2025	Pass	Session ends and redirects to login page.
13. Admin Registration	Samiul Siam	20 June 2025	Samiul Siam	28 June 2025	Pass	Admin account creation verified with email uniqueness check.

CHAPTER 5: DEVELOPMENT TOOL AND TECHNOLOGY

5.1 Development Technology

- **HTML:** Structured the content and layout of web pages.
- **CSS:** Styled the interface with colors, fonts, and responsive design.
- **Bootstrap:** Provided ready-made responsive UI components.
- **JavaScript:** Added interactivity and dynamic features to the frontend.
- **Firebase (JAVASCRIPT):** Handled backend logic, routing, and authentication.
- **Firebase Fire store:** Used as a real-time Firebase DATABASE to store portfolio data and Job-related info.

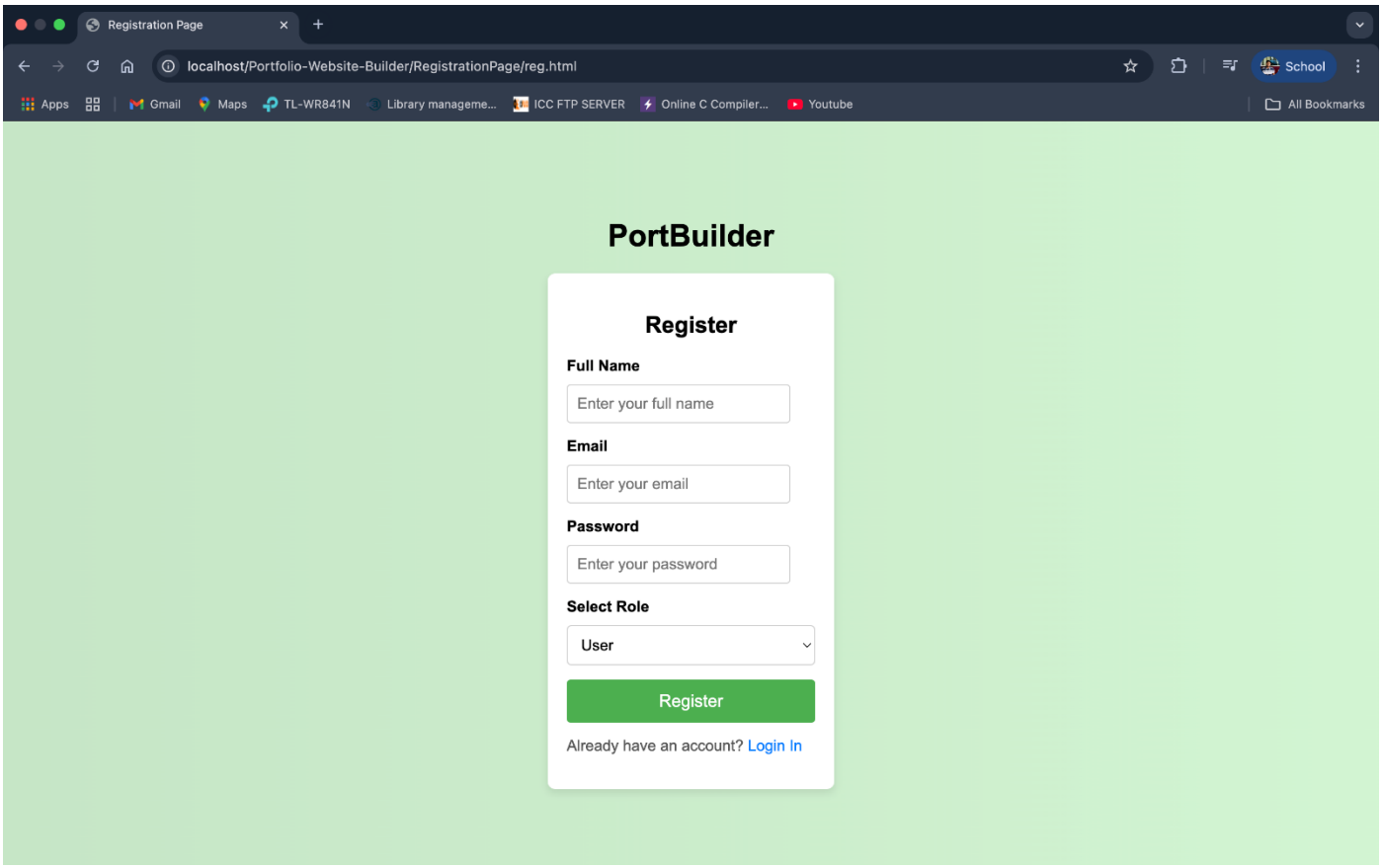
5.2 Development Tools and Platforms

- **VS Code:** Main code editor with extensions and terminal support.
- **Google Chrome:** Used for testing and debugging with dev tools.
- **Command Line Terminal:** Ran Firebase commands and Git operations.
- **GitHub:** Managed version control and project backups.
- **XAMPP:** Provided a local testing server.

CHAPTER 6: USER INTERFACE

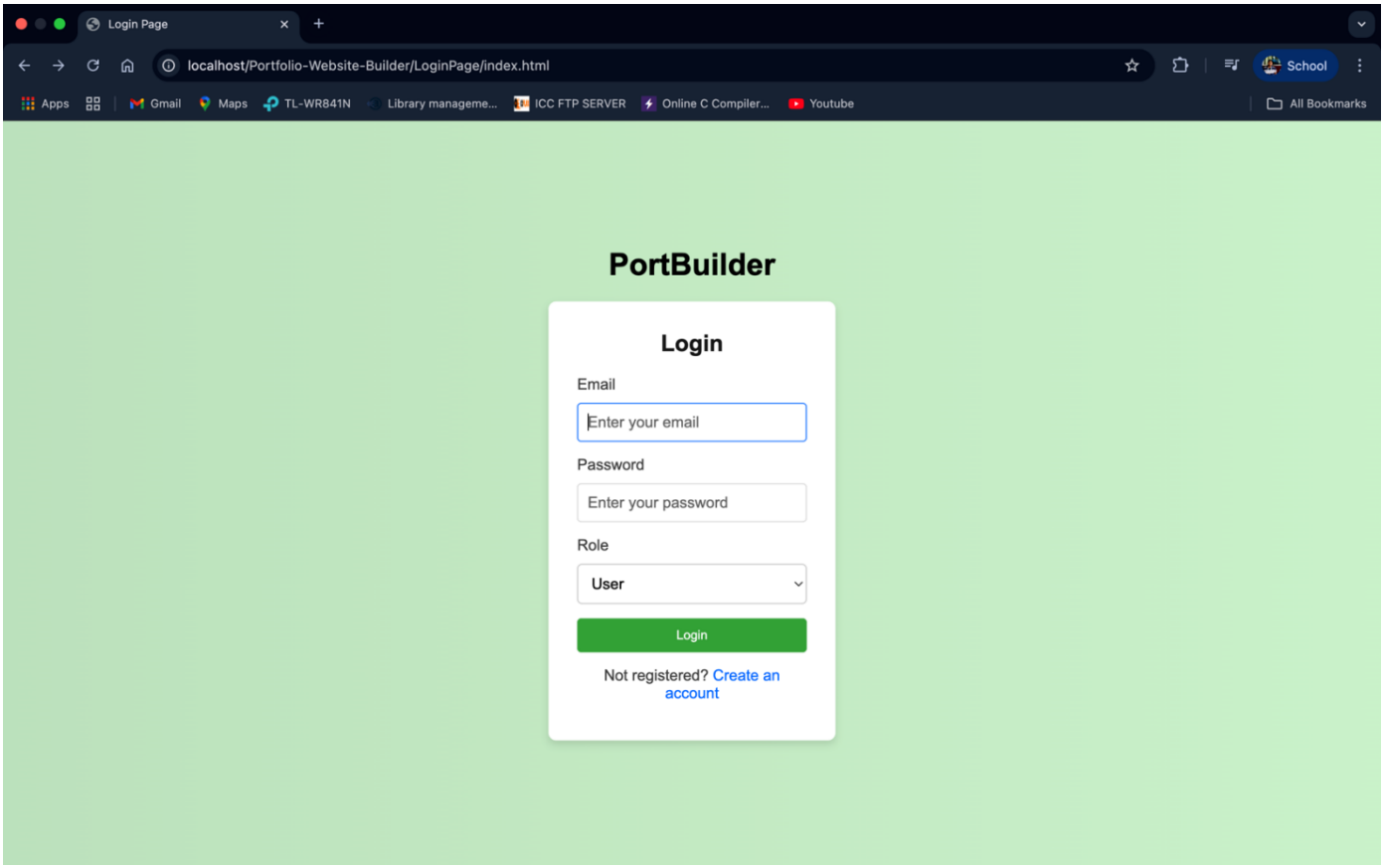
6.1 Registration Page

From this page, users can register to this system using necessary credentials. They can also have a login option in case of having an account.



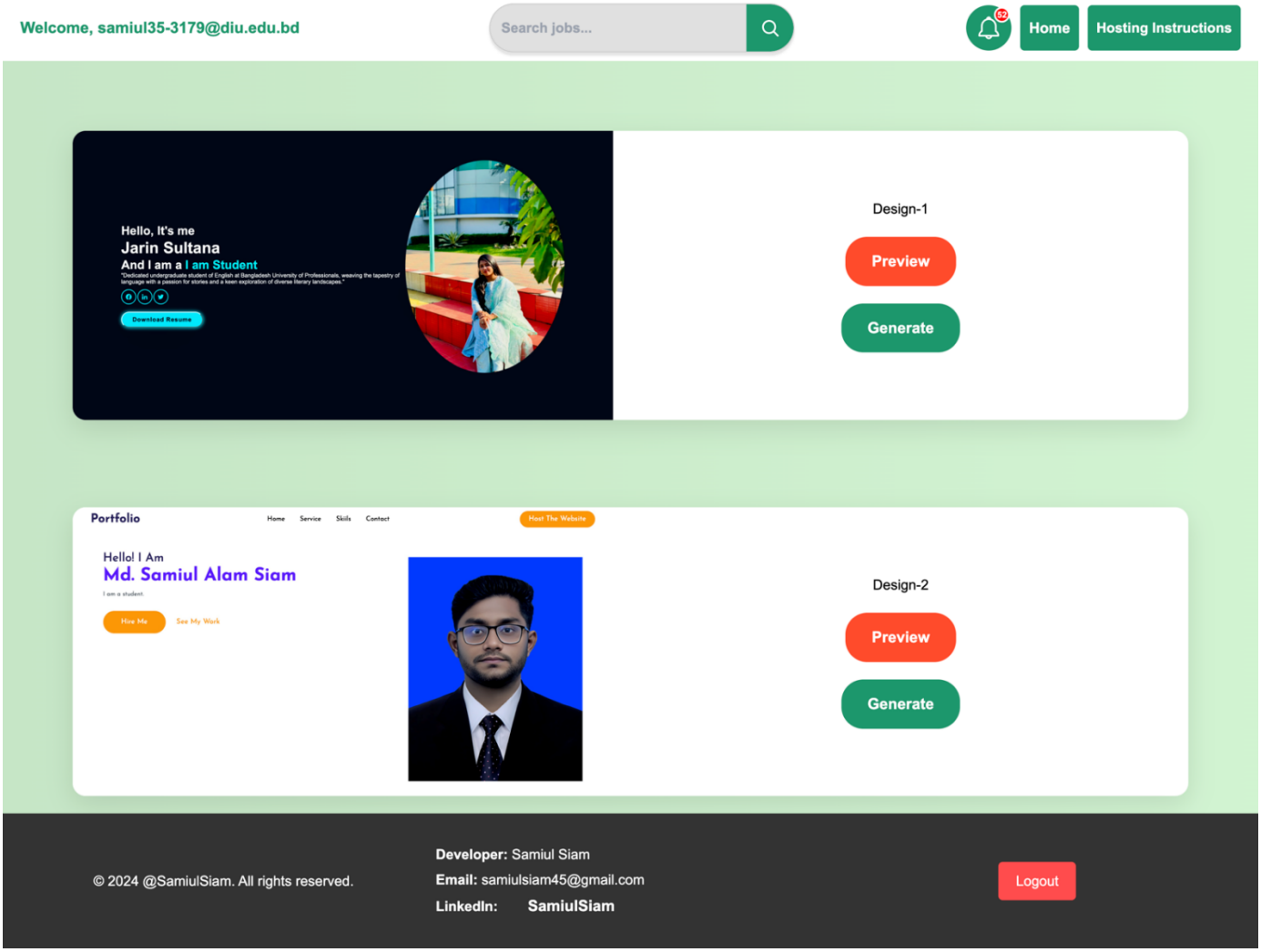
6.2 User Login page

Users can login to the system using login credentials**.**



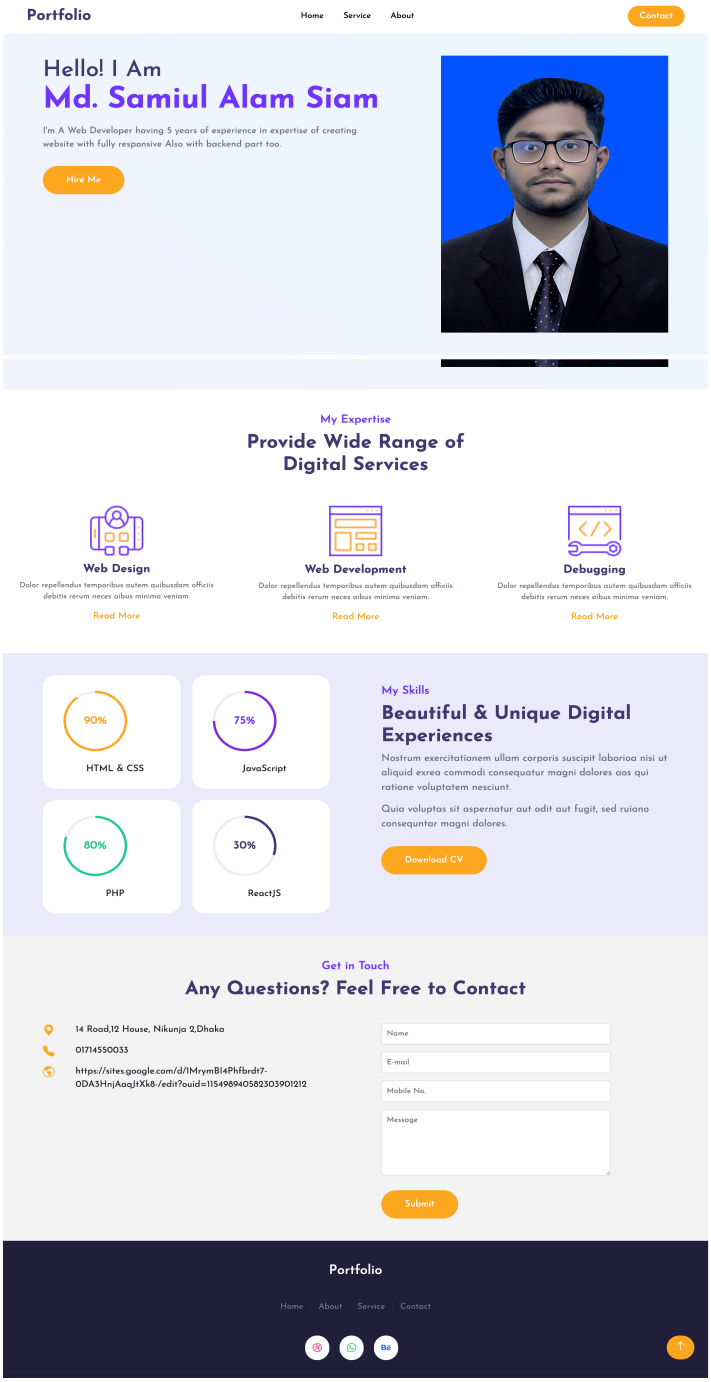
6.3 Home page

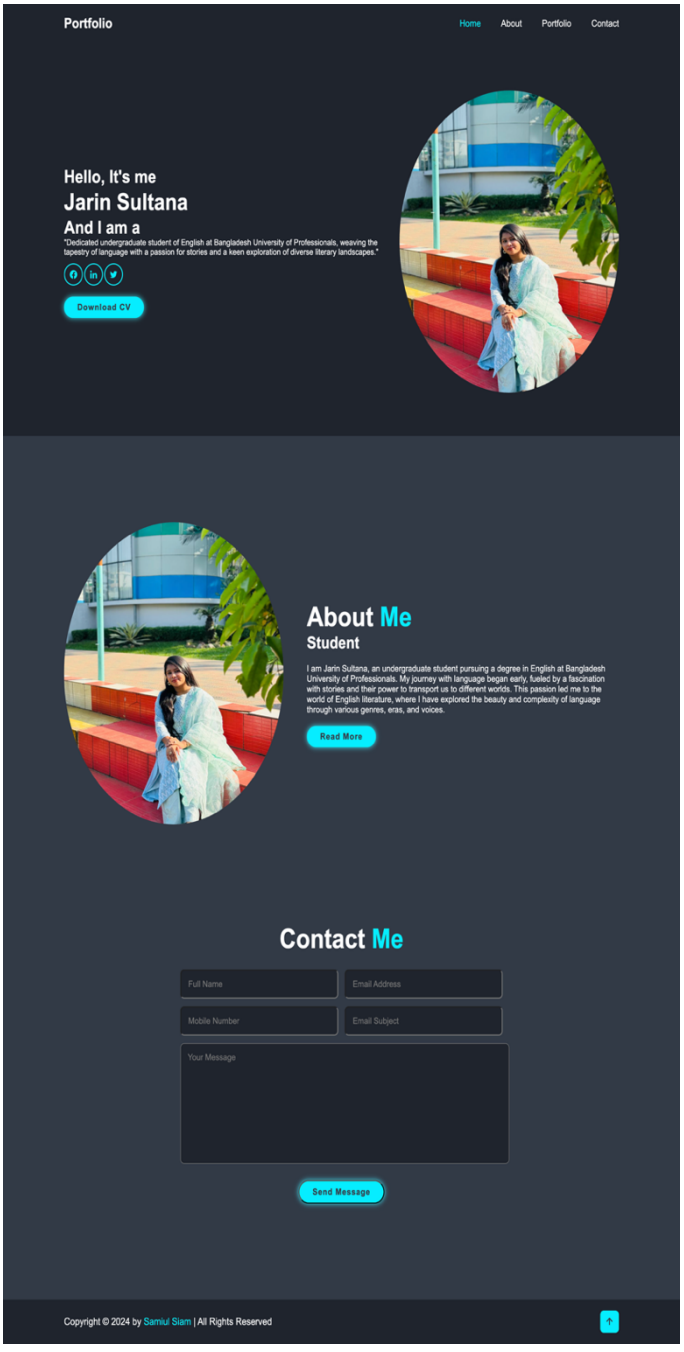
Users can preview and generate different portfolio designs, Search Jobs, and access hosting instructions.



6.4 Preview page

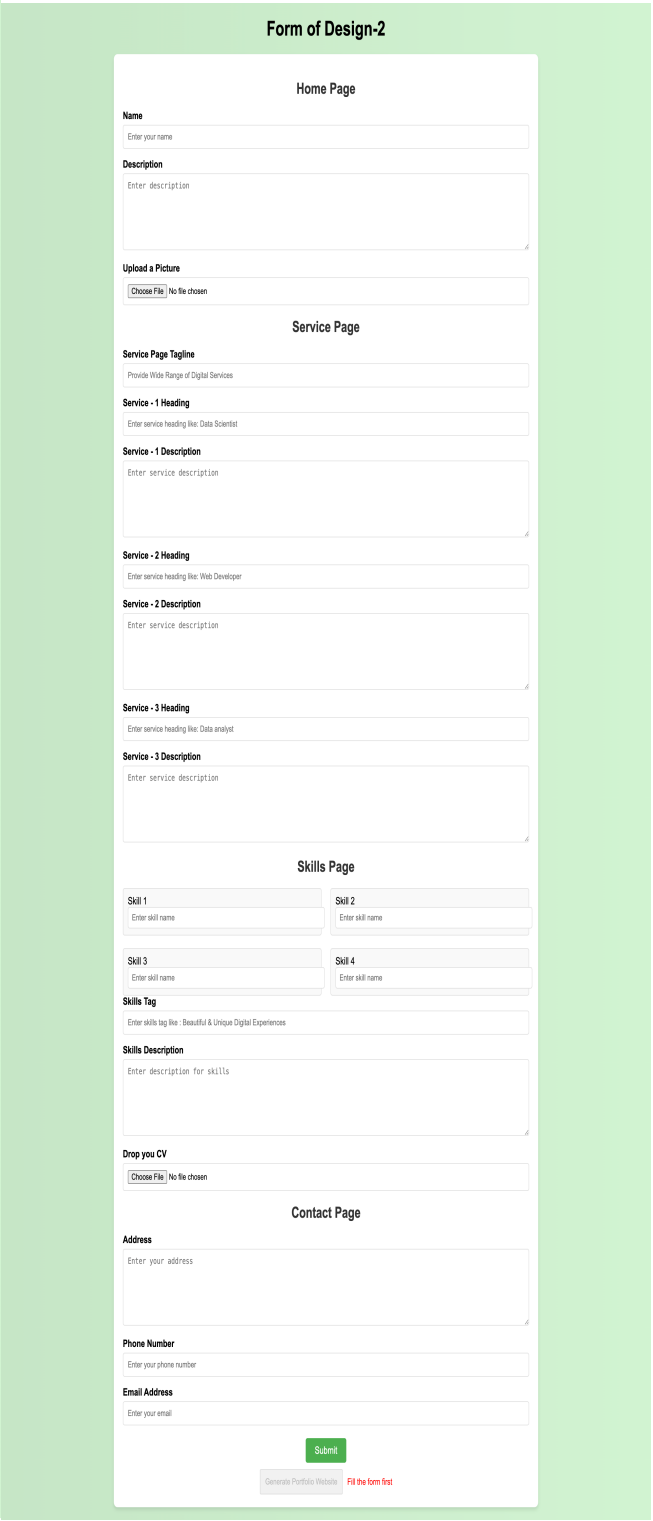
Users can preview different portfolio templates before generating or publishing them.





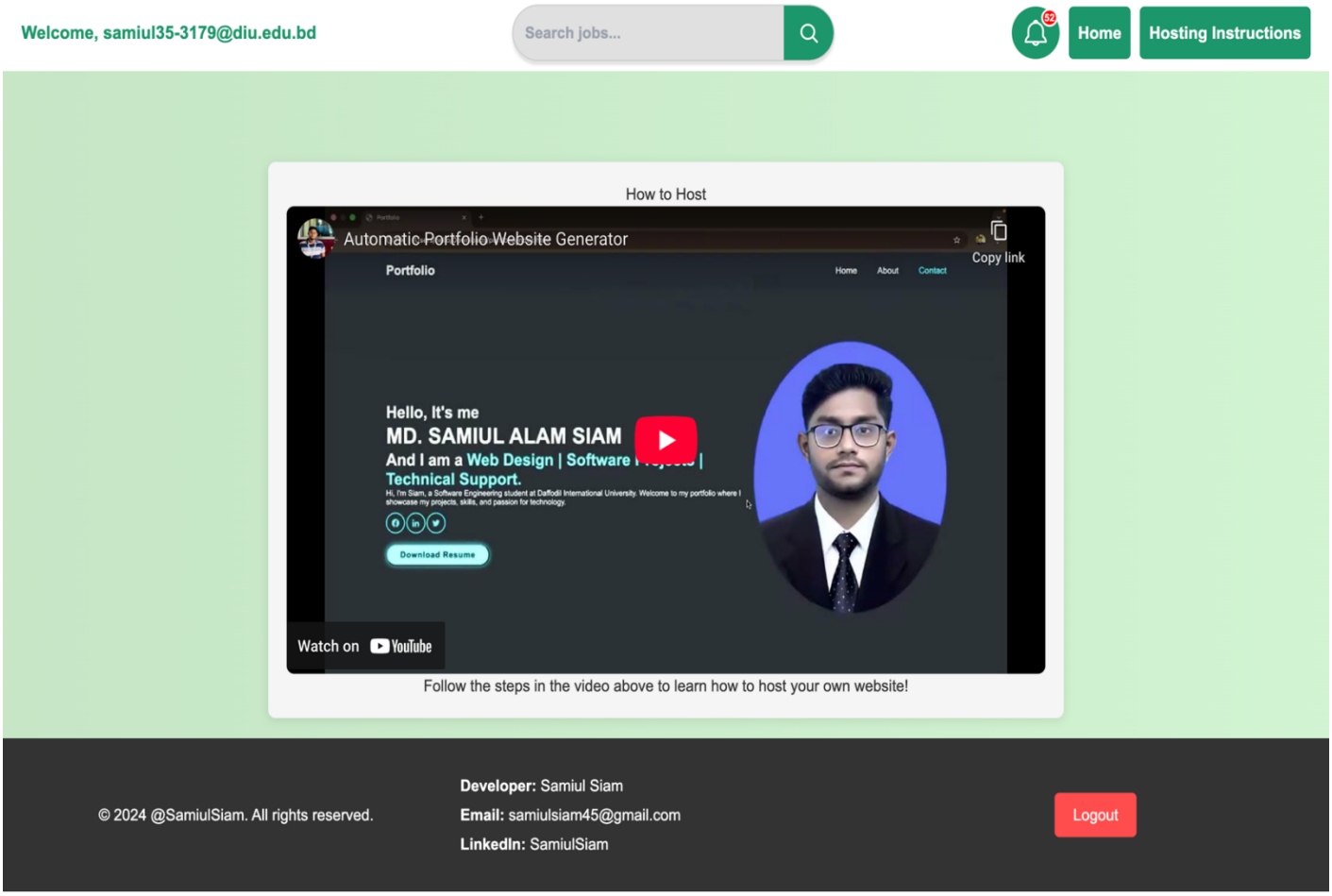
6.5 Generate Design Page

Users fill out forms with personal, service, and contact information to generate customized portfolio websites.



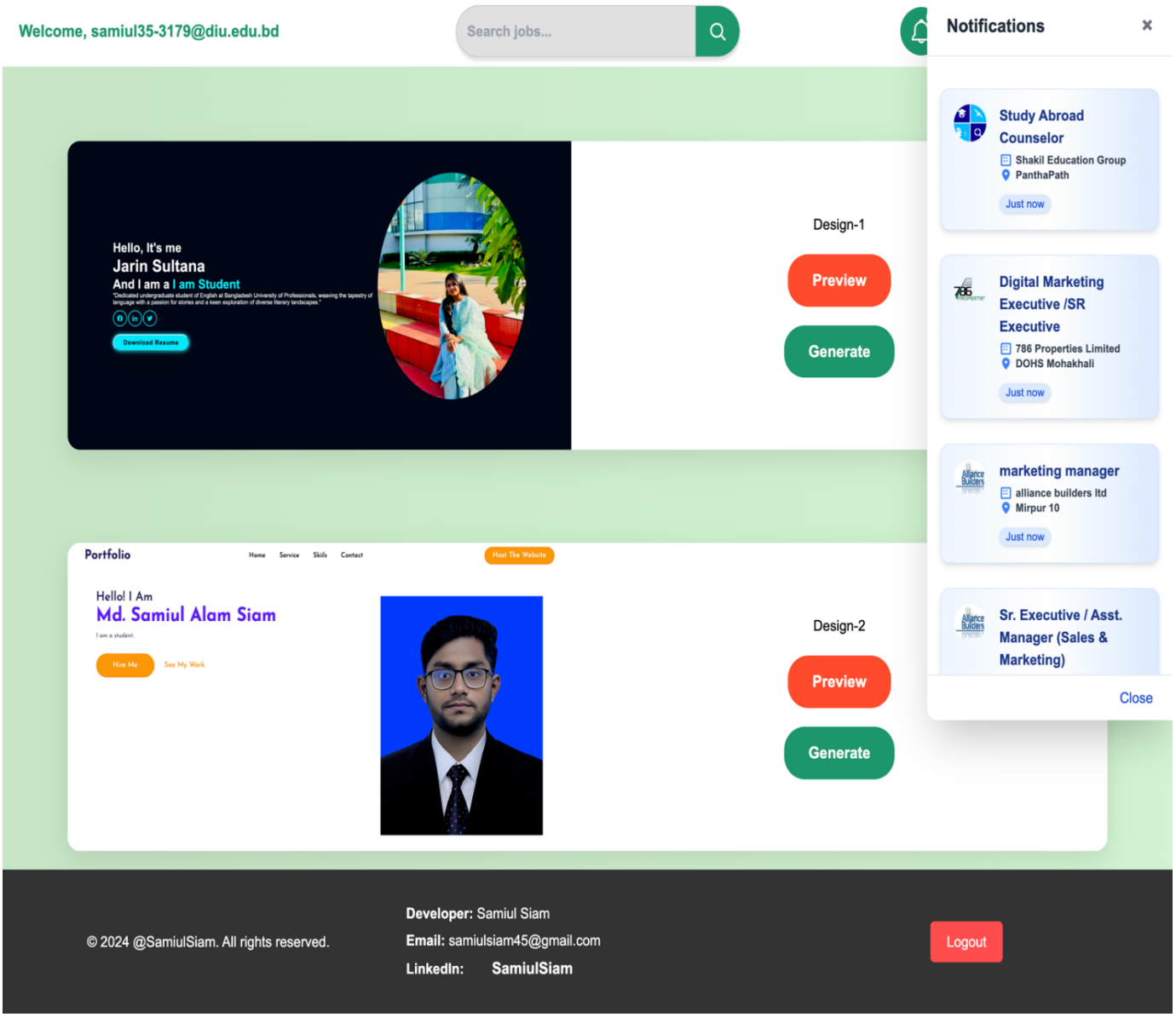
6.6 Hosting Instructions Page

Provides a step-by-step video tutorial to help users host their generated portfolio website.



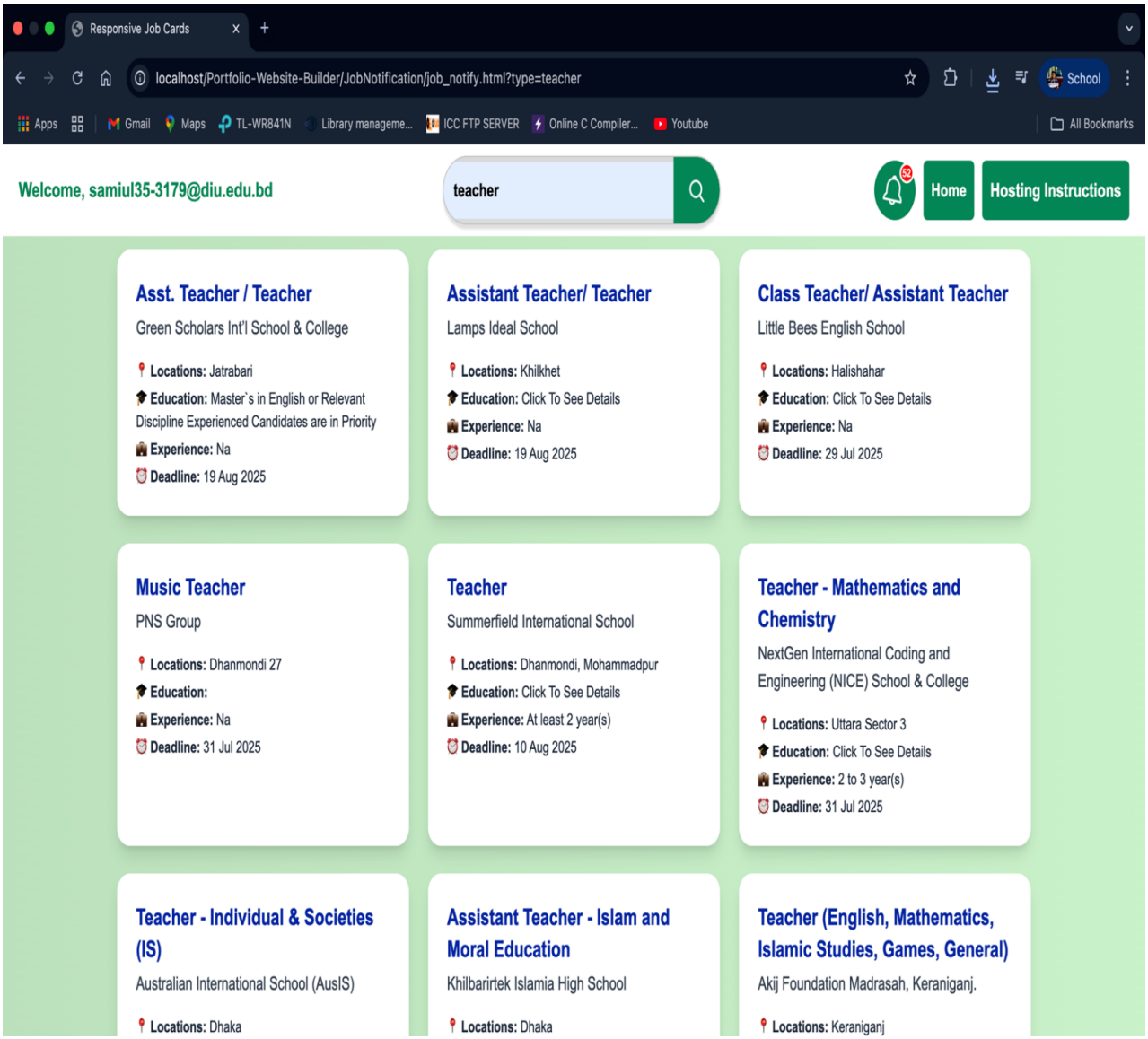
6.7 Notifications

Displays real-time Job alerts to users through a side panel, enhancing Job discovery directly from the platform.



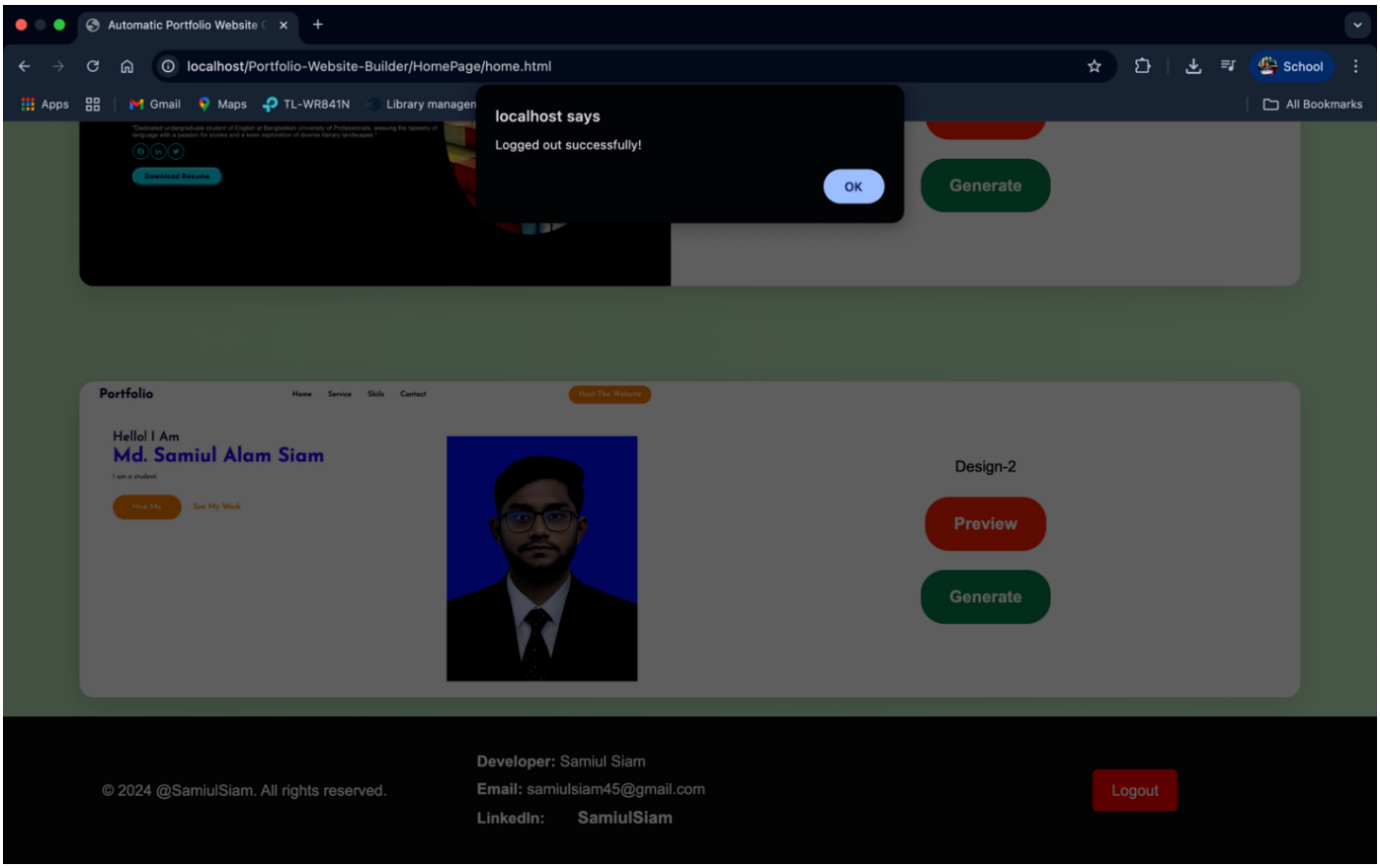
6.8Search Jobs

The Search Jobs feature enables users to find relevant Job postings by entering keywords and viewing results dynamically.



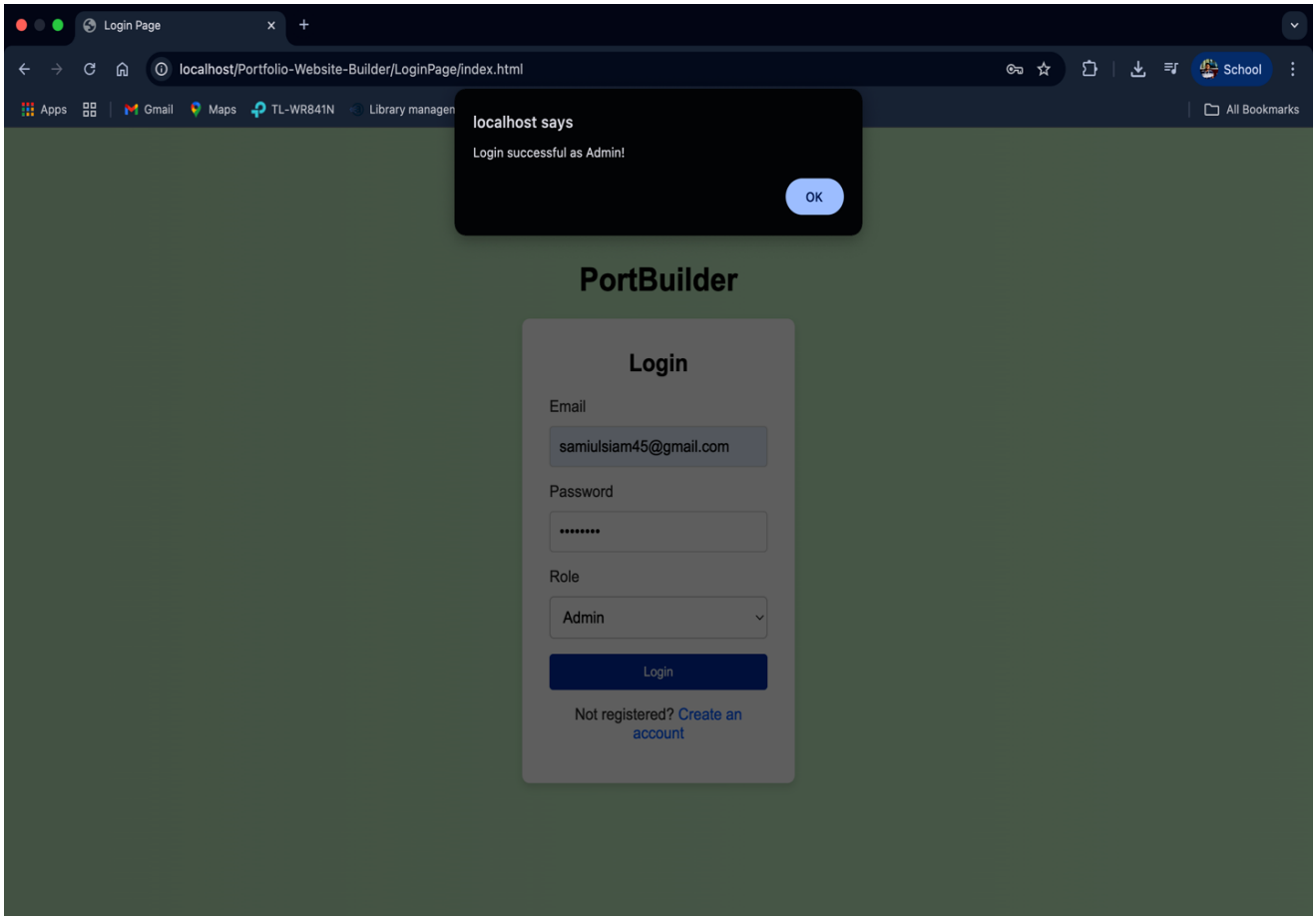
6.9 User Logout

Displays a confirmation popup upon successful logout and redirects the user to the login page.



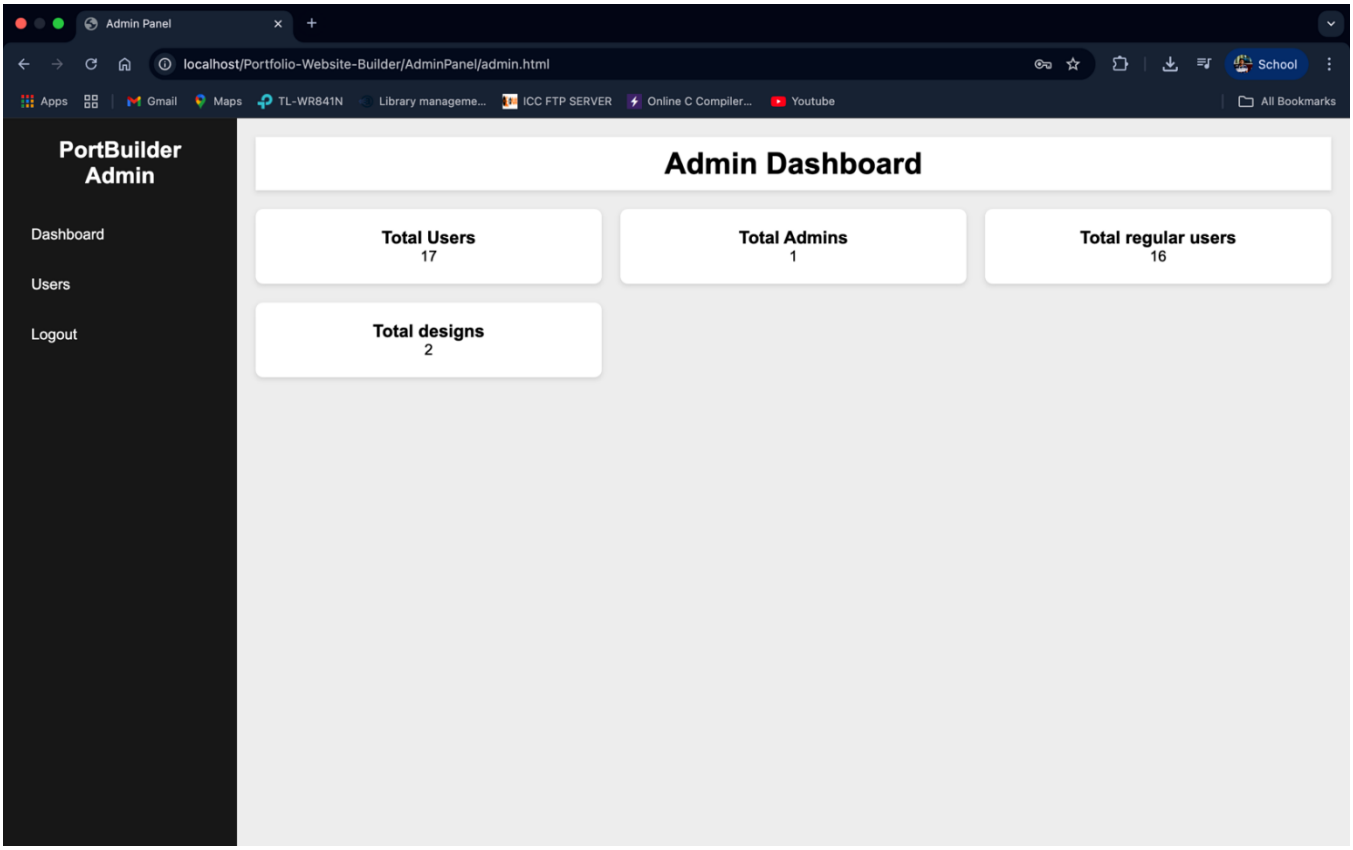
6.10 Admin Login page

Allows the Admin to Login by verifying email, password, and selected role, then confirms successful login with a popup



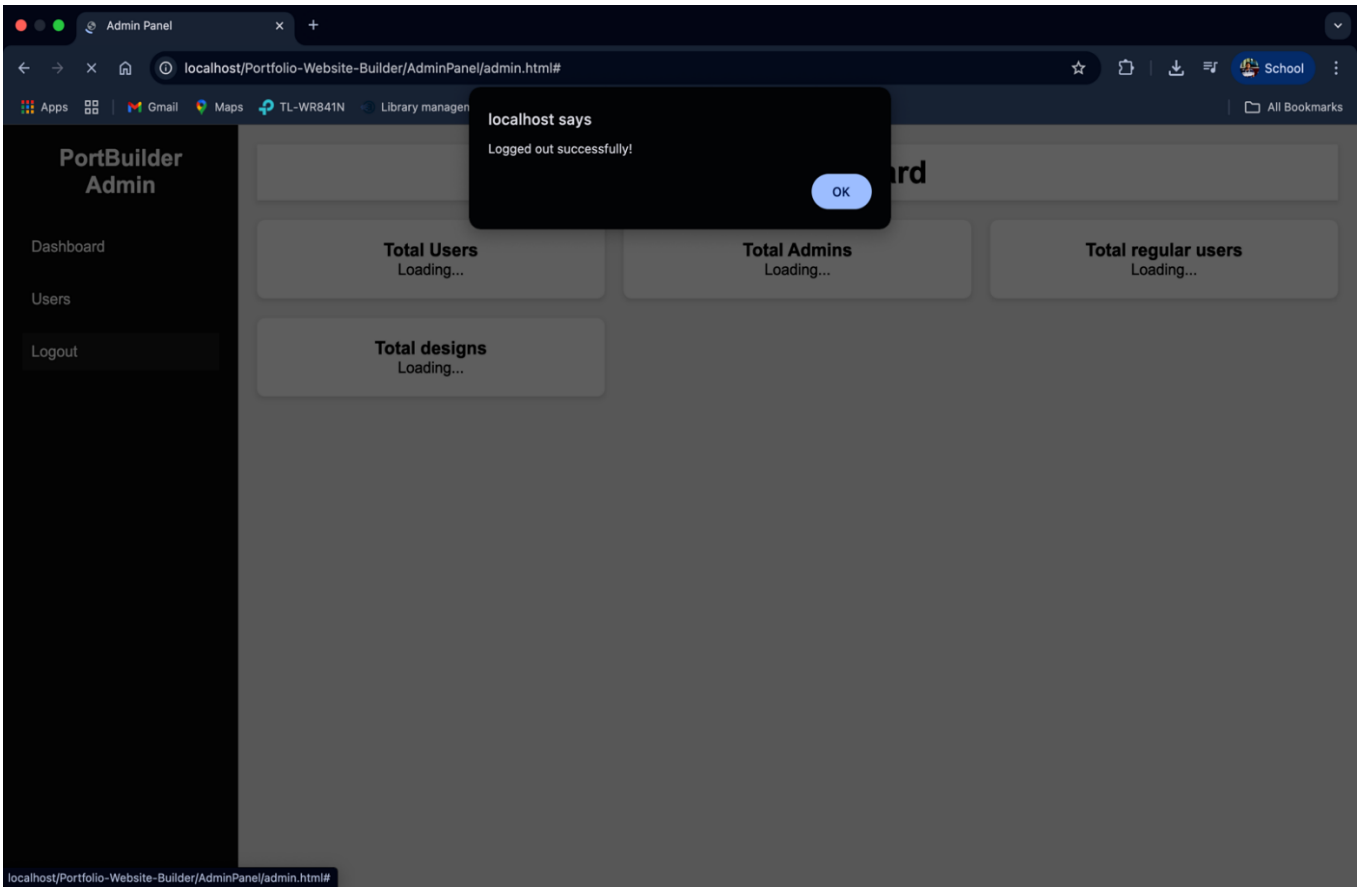
6.11 Admin Dashboard:

Displays total users, total Admins, regular users, and total design count in an organized layout.



6.12 Admin Logout

Allows the Admin to securely Logout, displaying a confirmation alert upon success.



CHAPTER 7: PROJECT SUMMARY

7.1 Overview

The Portfolio Website Builder is an online platform that helps students, freelancers, and Job seekers create personalized portfolio websites. It makes the process easier by providing a step-by-step form that gathers user information and automatically generates a custom portfolio.

Two key features improve the platform's usefulness:

- A Job Notification Bar for real-time updates on available positions.
- A Search Jobs module that takes users to external Job portals for more opportunities.

7.2 Achievements

The project fulfills all its major functional requirements, including:

- **User Registration and Login** using Firebase Authentication.
- **Dynamic Portfolio Generation** based on user-submitted content.
- **Job Notification Integration** with real-time Job display.
- **Job Search Module** allowing external Job exploration.
- **Admin Dashboard** to track user activity, total designs, and role-based stats.
- **Firebase Fire store Integration** for efficient data storage and real-time updates.

7.3 Limitations

While the system successfully meets its core objectives, a few limitations were identified:

- **Static Role Management:** Roles such as Admin or User are hardcoded and not manageable from the dashboard.
- **Limited Admin Monitoring:** There is no in-depth analytics or tracking module for Admin review.
- **Minimal Theme Customization:** Users cannot yet choose between multiple templates or customize layouts deeply.

7.4 Future Enhancement

To improve scalability, usability, and functionality, the following enhancements are proposed:

- **Dynamic Role and Permission Management:** Allow roles to be added/edited through an Admin interface.
- **Advanced Portfolio Customization:** Let users select from multiple themes, color palettes, and layouts.
- **Admin Analytics Dashboard:** Introduce charts, real-time monitoring, and detailed usage reports.
- **Email Notification System:** Notify users about Job matches or portfolio updates via email.
- **SEO Optimization:** Improve the visibility of user-generated portfolios on search engines.
- **Mobile Responsiveness Improvements:** Enhance UI for better mobile and tablet compatibility.
- **Advanced Job Matching:** Integrate machine learning to recommend Jobs based on user profiles.

7.5 Conclusion

The **Portfolio Website Builder** is a meaningful and impactful project that simplifies the process of building online portfolios while integrating career-focused features. It effectively blends frontend user experience with backend automation through Firebase, resulting in a smooth and responsive system. Although there are areas for improvement, the platform has demonstrated strong performance and is well-positioned for future expansion and innovation.

REFERENCES

- **Firebase Authentication:** https://firebase.google.com/docs/auth
- **Cloud Firestore:** https://firebase.google.com/docs/firestore
- **Firebase Hosting:** https://firebase.google.com/docs/hosting
- **JavaScript MDN Web Docs:** https://developer.mozilla.org/en-US/docs/Web/JavaScript
- **HTML Living Standard:** https://html.spec.whatwg.org/
- **CSS Reference:** https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
- **Bootstrap Documentation:** https://getbootstrap.com/docs/5.3/getting-started/introduction/ –
- **GitHub Docs:** https://docs.github.com/en
- **npm Docs:** https://docs.npmjs.com/
- **Google Chrome DevTools:** https://developer.chrome.com/docs/devtools/
- **Material Design 3:** https://m3.material.io/
- **Dribbble:** https://dribbble.com/
- **Behance:** https://www.behance.net/
- **Awwwards:** https://www.awwwards.com/
- **Figma Design Community:** https://www.figma.com/community/
- **Adzuna Job API:** https://www.adzuna.com/developers
- **The Muse API:** https://www.themuse.com/developers/api/v2
- **LinkedIn for Developers:** https://developer.linkedin.com/
- **BDJobs:** https://www.bdjobs.com/
- **Google for Jobs:** https://developers.google.com/search/docs/appearance/google-jobs –
- **IEEE Author Center:** https://authorcenter.ieee.org/
- **Purdue Online Writing Lab (OWL):** https://owl.purdue.edu/owl/research_and_citation/ieee_style/index.html
- **Google Scholar:** https://scholar.google.com/
- **ACM Digital Library:** https://dl.acm.org/
- **ResearchGate:** https://www.researchgate.net/
- **Lucid chart:** https://www.lucidchart.com/pages/
- **Draw.io :** https://app.diagrams.net/
- **Trello:** https://trello.com/en
- **Notion:** https://www.notion.com/
- **Grammarly:** https://www.grammarly.com/

PLAGLARISM REPORT

212-35-3179

ORIGINALITY REPORT

15%	11%	1%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
3	123dok.com Internet Source	1%
4	Submitted to Institute of Technology Blanchardstown Student Paper	1%
5	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
6	Areias, Davide Figueiredo. "Development of Cloud-Ready Platform with 3D Modeling and Energy Calculation Engine", Universidade de Coimbra (Portugal) Publication	<1%
7	Submitted to University of Sydney Student Paper	<1%
8	Submitted to Higher Education Commission Pakistan Student Paper	<1%
9	glasnost.itcarlow.ie Internet Source	<1%
10	Submitted to Institute of Technology, Nirma University Student Paper	<1%
11	Submitted to Middle East College of Information Technology	<1%

	Student Paper	
12	crackhow4.com Internet Source	<1%
13	Submitted to Informatics Education Limited Student Paper	<1%
14	www.bartleby.com Internet Source	<1%
15	Submitted to Colorado State University, Global Campus Student Paper	<1%
16	Submitted to Griffith College Dublin Student Paper	<1%
17	essay.utwente.nl Internet Source	<1%
18	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
19	opendata.com.pk Internet Source	<1%
20	tsecurity.de Internet Source	<1%
21	www.coursehero.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

LIBRARY CLEARNCE

ACCOUNT CLERANCE