**PROJECT REPORT**

**ON**

**CAREER COUNSELLING PORTAL**

**Project-I**



Department of Computer Science and Engineering

**CHANDIGARH ENGINEERING COLLEGE JHANJERI , MOHALI**

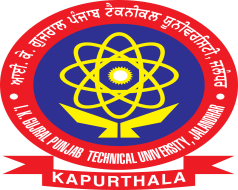
**In partial fulfillment of the requirements for the award of the Degree of**   **Bachelor of Technology in Computer Science & Engineering**

**SUBMITTED BY: Under the Guidance of**

Prateek Singh Jadon Dr. Ashima Shahi

2230829 Assistant Professor

MAY, 2025



**DECLARATION**

I, Prateek Singh Jadon hereby declare that the report of the project entitled “ Career Counselling Portal” has not presented as a part of any other academic work to get my

degree or certificate except Chandigarh Engineering College Jhanjeri, Mohali, affiliated to I.K. Gujral Punjab Technical University, Jalandhar, for the fulfillment of the requirements for the degree of B.Tech in Computer Science & Engineering.

(Student Signature with Date) (Mentor Signature with Date)

**Prateek Singh Jadon DR. Ashima Shahi**

2230829 Assistant professor, CSE Semester: 6

Signature of the Head of Department

**ACKNOWLEDGEMENT**

It gives me great pleasure to deliver this report on Project-I, which I worked on for my B.Tech in Computer Science & Engineering 3rd year, which was titled

"Career Counselling Portal “. I am grateful to my university for presenting me with such a wonderful and challenging opportunity. I also want to convey my sincere gratitude to all coordinators for their unfailing support and encouragement . I am extremely thankful to the HOD and Project Coordinator of Computer Science & Engineering at Chandigarh Engineering College Jhanjeri , Mohali (Punjab) for valuable suggestions and the heartiest cooperation. I am also grateful to the management of the institute and Dr. Avinash, Director Engineering, for giving me the chance to acquire the information. I also appreciate all of my faculty members, who have instructed me throughout my degree.

(Signature of Student)

**Prateek Singh Jadon**

**Table of Contents**

1. Introduction ...................................................................................................................... 17
   1. Vision Statement ................................................................................................................... 17 1.2. Related Model Analysis ......................................................................................................... 17

1.3. Project Deliverables ............................................................................................................... 18 1.4. System Limitations/Constraints.............................................................................................. 18 1.5. Tools and Technologies ......................................................................................................... 18

* 1. Relevance to Course Modules ................................................................................................ 19
  2. Conclusion ............................................................................................................................ 19

1. Problem Definition............................................................................................................ 21
   1. Problem Statement ................................................................................................................. 21
   2. Problem Solution ................................................................................................................... 21
   3. Objectives of the Proposed System ........................................................................................ 21
   4. Scope .................................................................................................................................... 22
   5. Modules/ System components ................................................................................................ 22
      1. Module 1: User Registration and Authentication.............................................................. 22
      2. Module 2: Career Guidance Chatbot ................................................................................ 22
      3. Module 3: Blog Management .......................................................................................... 23
      4. Module 4: Counsellor Interaction .................................................................................... 23
      5. Module 5: Notification System ........................................................................................ 23 2.5.6 Module 6: Reviews and Ratings ...................................................................................... 23
   6. Conclusion ............................................................................................................................ 23
2. Requirement Analysis ....................................................................................................... 25
   1. User classes and characteristics .............................................................................................. 25
   2. Requirement Identifying Technique ....................................................................................... 26
      1. Usecase Diagram ............................................................................................................. 26 3.2.2 UsecaseDescription ......................................................................................................... 28
   3. Functional Requirements ....................................................................................................... 39
   4. Non-Functional Requirements ............................................................................................... 40
      1. Performance ..................................................................................................................... 40
      2. Safety .............................................................................................................................. 40
      3. Reliability ........................................................................................................................ 40
      4. Robustness ....................................................................................................................... 40 3.4.5 Interoperability ................................................................................................................ 40
   5. External Interface Requirements ............................................................................................ 40
      1. User Interfaces Requirements ........................................................................................... 40
      2. Software interfaces ........................................................................................................... 41
      3. Hardware interfaces ......................................................................................................... 42
      4. Communications interfaces .............................................................................................. 42
3. Design and Architecture .................................................................................................... 45
   1. Architectural Design .............................................................................................................. 45
   2. Design Models....................................................................................................................... 47
   3. Data Design ........................................................................................................................... 61
      1. Data Dictionary ................................................................................................................ 62
   4. Human Interface Design ........................................................................................................ 75
      1. Screen Images and Actions............................................................................................... 75
   5. Conclusion ............................................................................................................................ 92
4. Implementation ................................................................................................................. 94
   1. Algorithm .............................................................................................................................. 94 5.2. External APIs/SDKs .............................................................................................................. 98

5.3. User Interface ........................................................................................................................ 99

* + 1. Login Screen .................................................................................................................... 99
    2. SignUp Screen ............................................................................................................... 100
    3. HomeScreen.................................................................................................................. 10 5.4. Conclusion .......................................................................................................................... 102

1. Testing and Evaluation .................................................................................................... 104
   1. Unit Testing ......................................................................................................................... 104
   2. Functional Testing ............................................................................................................... 109
   3. Integration Testing ............................................................................................................... 117
2. Conclusion and Future Work........................................................................................... 125
   1. Conclusion .......................................................................................................................... 125
   2. Future Work ........................................................................................................................ 125

**List of Figures**

Figure 1: Usecase Diagram ....................................................................................................... 27

Figure 2: Architectural Diagram ............................................................................................... 46

Figure 3: Sequence Diagrams ................................................................................................... 47

Figure 4: Context Diagram ....................................................................................................... 58

Figure 5: Data Flow Digram ..................................................................................................... 60

Figure 6: Entity Relationship Diagram ...................................................................................... 65

Figure 7: Screen Images............................................................................................................ 75

Figure 8: Login Screen ............................................................................................................. 99

Figure 9: SignUp Screen ......................................................................................................... 100 Figure 10: Home Screen ......................................................................................................... 101**.**

**List of Tables**

Table 1: Related Model Analysis with proposed project solution .............................................. 17

Table 2: Tools and Technologies for Proposed Project .............................................................. 18

Table 3: User classes and characteristics ................................................................................... 25

Table 4: Usercase Descriptions ................................................................................................. 28

Table 5: Algorithms .................................................................................................................. 94

Table 6: Details of APIs used in the project .............................................................................. 98

Table 7: Unit Testing .............................................................................................................. 104

Table 8: Functional Testing .................................................................................................... 109

Table 9: Integration Testing .................................................................................................... 117

Chapter 1

# Introduction

## Vision Statement

In today's landscape, students encounter numerous challenges when it comes to selecting their career paths. Many find themselves uncertain about their interests, often turning to peers or external influences for guidance rather than exploring their own passions. Enter CareerGPT, designed to assist individuals in uncovering their unique interests and passions, guiding them towards the most suitable career paths.

## Related Model Analysis

**Table 1: Related Model Analysis with proposed project solution**

|  |  |  |
| --- | --- | --- |
| **Model Name** | **Weakness** | **Proposed Project Solution** |
| OpenAI GPT Models | Paid service after a limited number of free requests, which can be costly for extended use. | The Career Counselling Portal will use the LLAMA2 model, which is free and can be finetuned, making it more accessible and cost-effective for users. |
| Chatbase | Limited dataset size before requiring payment, which can restrict the depth of information provided. | Our portal will leverage  LLAMA2, which does not  impose strict limits on dataset size and supports more extensive and detailed interactions without additional costs. |
| Falcon 7B | Does not store chat history, leading to potential disruptions in continuous user support and lack of context in interactions. | LLAMA2's ability to store chat history will ensure seamless and context-aware interactions, enhancing the user experience by maintaining continuity in counselling sessions. |

## Project Deliverables

The project deliverables for the "Career Counselling Portal" include:

**Project Proposal**: Detailed project proposal outlining the project scope, objectives, and milestones.

**Software Requirements Specification (SRS)**: Comprehensive documentation of functional and non-functional requirements, including user stories, use case diagram, data flow diagram and entity relationship diagram.

**Prototypes**: Initial paper prototypes of the user interface and key system components to gather early feedback and refine the design.

**Project Implementation**: The fully developed and deployed Career Counselling Portal.

**Project Report**: Final project report summarizing the development process, challenges encountered, solutions implemented, and project outcomes.

## System Limitations/Constraints

Counsellor registration is restricted to individuals from India only; individuals from other countries are unable to register as counsellor. However, the platform's services are available to international students, despite its geographical focus on India and nearby cities.

## Tools and Technologies

**Table 2: Tools and Technologies for Proposed Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools**  **And**  **Technologies** | **Tools** | **Version** | **Rationale** |
| Visual Studio Code | 1.90 | IDE |
| Kaggle | --- | Model Training |
| Draw.io | --- | Design Work |
| **Technology** | **Version** | **Rationale** |
| React JS | 18 | Front-end Development |
| Python | 3.12.0 | Back-end Development |
| Django | 5.0.1 | Back-end Development |
|  | SQLite | --- | Database |

## Relevance to Course Modules

The Career Counselling Portal project integrates several courses studied during the BSSE program, reflecting a comprehensive application of the theoretical and practical knowledge acquired. Courses such as Software Engineering Principles, Database Management Systems, and Human-Computer Interaction were crucial in designing and implementing the platform’s architecture, ensuring robust data management, and enhancing user experience through an intuitive interface. The development of the AI chatbot for personalized career guidance drew heavily on Artificial Intelligence and Machine Learning courses, employing algorithms and techniques to provide tailored recommendations. Additionally, Web Development and Software Project Management courses facilitated the creation of a responsive and efficient system while adhering to project timelines and quality standards. Overall, this project exemplifies the practical integration of multiple disciplines, showcasing the relevance and application of the BSSE curriculum in addressing real-world problems

## Conclusion

In summary, this chapter outlined the vision and objectives of the Career Counselling Portal project. The related model analysis highlighted the advantages of using the LLAMA2 model to provide extensive career guidance. The project deliverables, from the proposal to the final report, were delineated, ensuring a structured development process. The system's constraints, specifically the regional limitation on counsellor registration, were acknowledged, and the tools and technologies employed were listed to illustrate the project's technical foundation. Finally, the project's relevance to various BSSE course modules was discussed, demonstrating the practical application of academic knowledge to solve real-world problems. This chapter sets the stage for a comprehensive understanding of the project's scope, methodology, and significance.

Chapter 2

# Problem Definition

## Problem Statement

In today's educational landscape, students face significant challenges in selecting their career paths, often grappling with uncertainty about their interests and potential professions. Many students rely on peer opinions or external influences rather than exploring their own passions, leading to dissatisfaction and misaligned career choices. The lack of personalized, accessible, and cost-effective career counselling escalates this issue, leaving students without the necessary guidance to make informed decisions. Existing solutions, such as paid AI models and limited dataset services, are often expensive and inaccessible to a broad audience, restricting their utility. Moreover, the absence of continuous support and contextual interactions in these models hinders the counselling process. To address these problems, we are developing the Career Counselling Portal, a platform that leverages the LLAMA2 model to provide free, personalized, and extensive career guidance. This system will facilitate continuous, context-aware interactions, ensuring students receive consistent and tailored advice. By offering a comprehensive and accessible counselling solution, the Career Counselling Portal aims to empower students to make informed career choices aligned with their interests and aspirations, ultimately fostering a more satisfied and productive future workforce.

## Problem Solution

* Provide free and accessible career counselling services to students globally.
* Utilize the LLAMA2 model to offer personalized and detailed career guidance.
* Maintain continuous and context-aware interactions to enhance user experience.
* Overcome the limitations of existing paid models and limited dataset services.
* Facilitate the identification of students' interests and passions to guide them towards suitable career paths.  Ensure a user-friendly interface that simplifies the counselling process.
* Support counsellor registration, focusing on professionals from India, while offering services to international students.
* Achieve a responsive and efficient system through the use of modern web development technologies.
* Develop comprehensive documentation and reports to facilitate future enhancements and scalability.

## Objectives of the Proposed System

* Provide free career counselling to all students in the world.
* Achieve high satisfaction rate among users, measured through feedback and surveys.
* Ensure smooth uptime for the portal to provide continuous and reliable access to career guidance.
* Reduce the cost of career counselling services for users by 100% compared to existing paid models.
* Maintain a response time of under 1 minute for AI-driven career advice queries.
* Support career counselling sessions for international students.
* Reduce the manual effort required by counsellor through automated data management and AI-driven recommendations.
* Provide training and support materials that lead to high satisfaction rate among counsellor using the platform.

## Scope

The scope of the project encompasses the development and deployment of a user-friendly web interface, designed using modern front-end technologies like React JS, and a secure back-end infrastructure built with Python and Django. The database management will be handled by SQLite, ensuring efficient storage and retrieval of user data. The project will include the integration of extensive datasets to enhance the depth and accuracy of the counselling provided.

The geographical focus of the counsellor registration will be limited to individuals from India, although the platform's services will be available to students globally. The project will also involve the development of comprehensive documentation, including Software Requirements Specification (SRS), design diagrams, and user manuals. Key deliverables will include a detailed project proposal, initial prototypes, and a final project report summarizing the development process and outcomes.

Overall, the Career Counselling Portal aims to bridge the gap in accessible and effective career counselling, providing a scalable solution that can be expanded to include additional features and support more users in the future. The boundaries of the project are clearly defined to ensure focused development and successful implementation, with an emphasis on delivering high-quality, reliable, and secure services to all users.

## Modules/ System components

### Module 1: User Registration and Authentication

* **FE-1:** Sign up with personal details (name, email, password).
* **FE-2:** Login functionality for existing users.
* **FE-3:** Email verification for new accounts.

### Module 2: Career Guidance Chatbot

* **FE-1:** Personalized career guidance sessions via AI Chatbot.
* **FE-2:** Real-time interaction with students.
* **FE-3:** Integration with user profile to tailor responses.

### Module 3: Blog Management

* **FE-1:** Add career-related blogs for students.
* **FE-2:** Access and read blogs.

### Module 4: Counsellor Interaction

* **FE-1:** View counsellor profiles and expertise.
* **FE-2:** Schedule appointments with counsellors.
* **FE-3:** Chat functionality for real-time communication.

### Module 5: Notification System

**FE-1:** Receive notifications for new messages from counsellors.

### Module 6: Reviews and Ratings

* **FE-1:** Provide feedback on counselling sessions.
* **FE-2:** Rate counsellors based on performance.
* **FE-3:** View aggregated ratings and reviews of counsellors.

## Conclusion

Fulfilling the outlined objectives, this chapter presents a comprehensive solution to the challenges surrounding career counselling. By developing the Career Counselling Portal, we've addressed the need for accessible and personalized guidance, leveraging AI technologies to offer free, context-aware interactions. The system's user-friendly interface, integrated modules, and emphasis on counsellor support ensure a seamless experience for both students and professionals. Through continuous improvement and scalability, this initiative aims to empower students globally in making informed career decisions, ultimately contributing to a more satisfied and productive future workforce.

Chapter 3

# Requirement Analysis

## User classes and characteristics

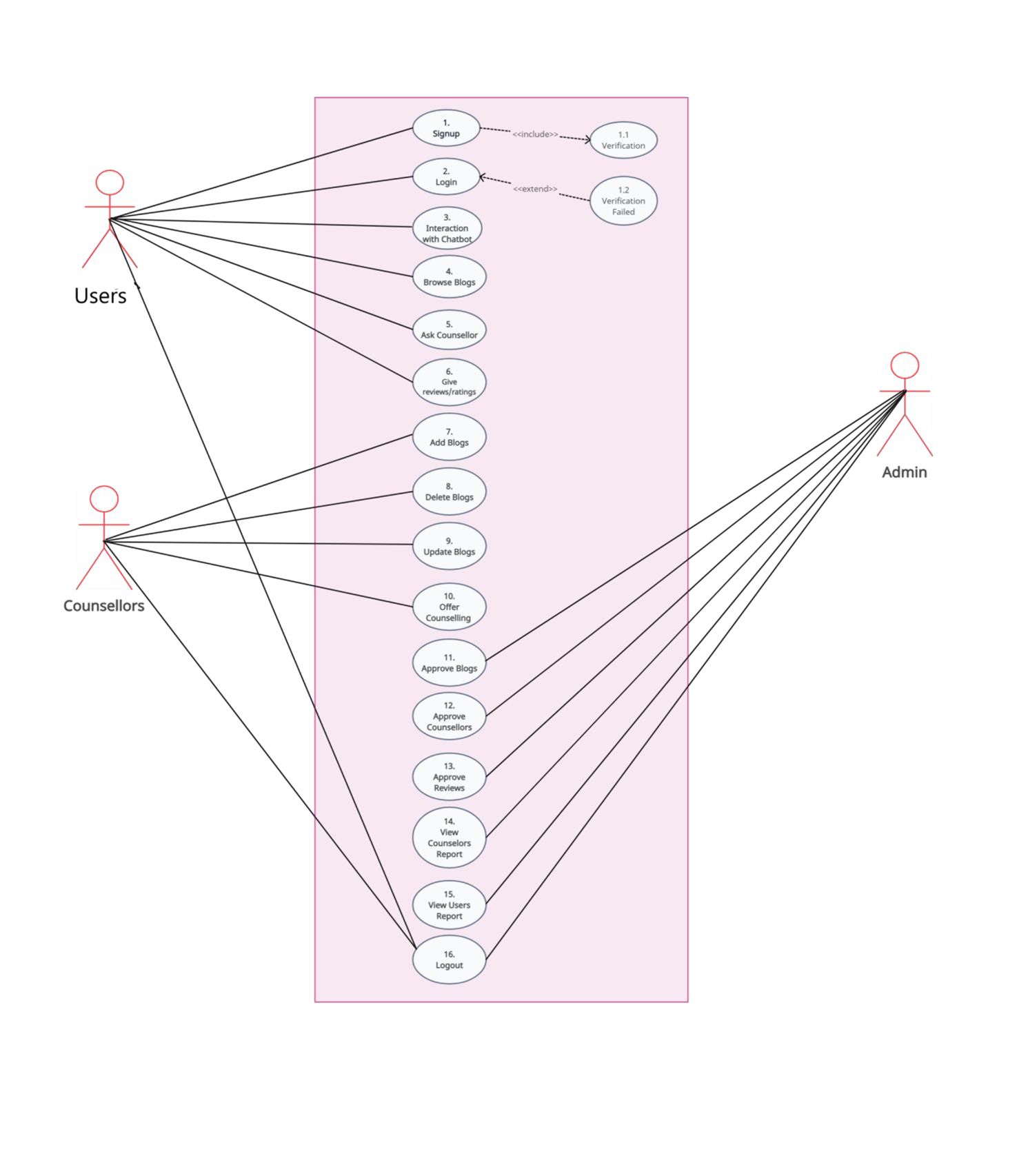
**Table 3: User classes and characteristics**

|  |  |
| --- | --- |
| **User Class** | **Description** |
| **Students** | Students are the primary users seeking career guidance. They range from high school to university level and use the portal to access career advice, read blogs, and interact with counsellors. Students are expected to use the system frequently, especially during career planning phases. |
| **Counsellors** | Counsellors provide professional guidance and support to students. They will use the portal to manage their profiles, publish blogs, interact with students via chat. |
| **Administrators** | Administrators oversee the system's operations, manage user accounts, approve counsellor registrations, and monitor the platform's overall functionality. They need access to detailed reports and system monitoring tools to ensure smooth operations. |

## Requirement Identifying Technique

### Use-case Diagram

**Figure 1: Use-case Diagram**



### Use case Description

**Table 4: Use case Descriptions**

|  |  |
| --- | --- |
| **ID** | **UC-1** |
| **NAME** | Signup |
| **PRIMARY ACTORS** | User, Counsellor |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | Users access the system and enter his information to make an account on the system. |
| **PRECONDITIONS** | User opens system on browser. |
| **POSTCONDITIONS** | User has created account in system. |
| **NORMAL FLOWS** | 1. User selects Sign Up from the main page. 2. System displays screen asking user to enter some basic details. 3. User enters details and clicks submit button. 4. System sends the OPT code to user to enter for verification. 5. System displays success message saying account has been created and redirects user to home page. |
| **EXCEPTIONS** | **2a**- Incorrect Email Address:  **2a-1** System displays an error message saying Invalid or non-existent email address. **2a-2** Prompt the user to enter a valid email address.  **2b**- Invalid Password:  **2b-1** System displays an error message saying to enter a valid Password should include uppercase characters, special characters, and be at least 8 characters long.  **4a**-Invalid OTP enter:  **4a-1** System displaying the error message saying to enter a correct OTP code for verification  **4a-2** If you haven't received the OTP due to an internet connection issue or any other reason, please press the "Resend" button to receive the OTP again. |
| **PRIORITY** | High |
| **Owner** | Prateek Jadon |

|  |  |
| --- | --- |
| **ID** | **UC-2** |
| **NAME** | Log in |
| **PRIMARY ACTORS** | Users, Counsellor |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | Users enter their existing account details for system access. Once verified, they can use the counselling services on our website. |
| **PRECONDITIONS** | User must have an existing account. |
| **POSTCONDITIONS** | User is successfully logged into the system and can now access career counselling services. |
| **NORMAL FLOW** | 1. User selects login button from main screen. 2. System displays login screen where user enters his email and password. 3. System checks login details for verification. 4. System displays welcome message and redirect user to main screen of the system. |
| **EXCEPTIONS** | **3a-** Data entered by User is invalid.  **3a1**- System displays error message and ask user to reenter details.  **3a2**- User either enters correct details or select forgot password option. **3a3-** Administrator sends a verification code to user and by using Code, user can recover his account. |
| **PRIORITY** | High |
| **Owner** | Prateek Jadon |

|  |  |
| --- | --- |
| **ID** | **UC-3** |
| **NAME** | Interaction with Chatbot |
| **PRIMARY ACTORS** | Users |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | Users can engage in an interactive chat with a CareerGPT chatbot, during which the CareerGPT asks about their educational background, interests, skills, and personality traits. Based on this information, CareerGPT predict the most suitable bachelor’s degree program that aligns with their personality and interests. |
| **PRECONDITIONS** | Users must be logged into the system. |
| **POSTCONDITIONS** | Users have a clear understanding of both the degree program they can choose and the major subjects associated with that degree. |
| **NORMAL FLOW** | 1. Users initiate their interaction with CareerGPT by selecting the "Chat with CareerGPT" option in the Hero section. 2. The screen opens that has the chat icon on the bottom right. 3. Users clicks on the icon and asks their initial question related to degree prediction. 4. CareerGPT’s first question is about the user’s educational background. |
|  | 1. Users provide details about their intermediate degree level. 2. CareerGPT then asks users to specify their preferred subjects, interests, skills, and personality traits. 3. Users provide their responses accordingly. 4. At the end, Career GPT not only predict the degree that matches their personality traits, interests, and skills but also suggests the major subjects for that degree. |
| **EXCEPTIONS** | **2a-** Users are not logged in.  **2a-1** The system redirects the users to the login screen.  **7a-** Users’ responses are incomplete or irrelevant.  **7a-1** CareerGPT generates the polite response for additional information. **8a-** Users are not satisfied with the suggested degree program.  **8a-1** CareerGPT asks them to contact counsellors by clicking the “Ask Counsellor” button on the Home page. |
| **PRIORITY** | High |
| **Owner** | Payal Rawat |

|  |  |
| --- | --- |
| **ID** | **UC-4** |
| **NAME** | Search Blogs |
| **PRIMARY ACTORS** | Users |
| **DESCRIPTION** | User enters name of Counsellor or name of fields to search specific Blogs. |
| **PRECONDITIONS** | User must have opened our website and clicked ‘Blogs’ option from navbar. |

|  |  |
| --- | --- |
| **POSTCONDITIONS** | Blogs are displayed from different Fields or Names of Counsellor that match the name entered by user. |
| **NORMAL FLOWS** | **3.0 User searches by Field name**   1. User enters name of the field he wants to search 2. Blogs of that field are displayed on screen. 3. User selects the blog that he wants to view. |
| **ALTERNATIVE FLOW** | **3.1 User searches by Counsellor name**   1. User enters name of the Counsellor whose blogs he wants to search. 2. Blogs of that counsellor are displayed on screen. 3. User selects the blogs that he wants to view. |
| **EXCEPTIONS** | **1a-** User enters name of a fields or counsellor that do not exists in system. **1a-1** User re-enters name of counsellor or fields. |
| **PRIORITY** | High |
| **Owner** | Prateek Jadon |

|  |  |
| --- | --- |
| **ID** | **UC-5** |
| **NAME** | Ask Counsellors |
| **PRIMARY ACTORS** | Users |
| **SECONDARY ACTORS** | Counsellors |
| **DESCRIPTION** | Users can seek guidance and advice by asking questions to the counsellor through real time communication with their specified counsellor. |
| **PRECONDITIONS** | User must be logged into the system. |
| **POSTCONDITIONS** | Message is sent successfully to the specified counsellor. |
| **NORMAL FLOW** | 1. The user goes to the "Ask Counsellor" section accessible from the navigation bar. 2. The system presents a screen that displays the registered counsellor. 3. The user chooses a counsellor of interest from the available list. 4. A screen with instructions or a privacy policy is provided to instruct users on how to initiate a conversation with a counsellor. 5. After reviewing the instructions and privacy policy, the user clicks "Next" to proceed. 6. A screen showing real time communication with the counsellor appears. 7. User sent message to the counsellor. 8. A message is dispatched to the specified counsellor containing the user's request. |
| **EXCEPTIONS** | **5a-** User Declines to Proceed  **5a-1** User, after reviewing the instructions and privacy policy, chooses not to proceed by clicking "Next".  **5a-2** The system should return them to the counsellor selection or a suitable screen without initiating the conversation process.  **8a-** Message not sent  **7a-1** Due to network connection error, the message is not sent.  **7a-2** System should prompt them to check their internet connection and then resend the message. |
| **PRIORITY** | High |
| **Owner** | Sakshi Kumari |

|  |  |
| --- | --- |
| **ID** | **UC-6** |
| **NAME** | Give Reviews/Ratings |
| **PRIMARY ACTORS** | Users |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | Users can provide feedback about the website's functionality and ease of use, while users who have engaged with counsellors can assign ratings and can also give reviews to them. |
| **PRECONDITIONS** | User must have used the website's features and they must have interacted with counsellors. |
| **POSTCONDITIONS** | User feedback is recorded and may be used to improve the website's functionality and counsellor ratings and reviews are updated, providing valuable information to other users considering counselling services. |
| **NORMAL FLOW** | 1. User navigates to the "Give Reviews/Ratings" section on the website. 2. The system presents options for either providing feedback on the website's functionality or assigning ratings to counsellors. 3. **For Website Review:**  * User selects the "Website Review" option. * The system displays a feedback form, allowing the user to provide comments regarding the website's functionality. * User fills out the feedback form with their name, email address and comments. * User submits the feedback form to the system.       **4. For Counsellor Reviews & Ratings:**   * User selects the "Counsellor Reviews & Ratings" option. * The system displays a list of counsellors with whom the user has had interactions. * User selects a counsellor from the list. * User assigns a rating and give reviews to the selected counsellor based on their experience. * User submits the reviews & rating. |
| **EXCEPTIONS** | **3a-** User doesn't complete the feedback form for website review.  **3a-1** The system may remind them to do so or allow them to submit incomplete feedback. |
| **PRIORITY** | Medium |
| **Owner** | Sakshi Kumari |

|  |  |
| --- | --- |
| **ID** | **UC-7** |
| **NAME** | Add Blogs |
| **PRIMARY ACTORS** | Counsellors |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | The Counsellors add blogs by providing the necessary information for the blogs. |
| **PRECONDITIONS** | The Counsellors must be registered and logged into the system. |
| **POSTCONDITIONS** | The blog information is sent to the admin for verification. |
| **NORMAL FLOW** | 1. The Counsellors navigate to the “Add Blogs” option from their dashboard. 2. The system displays blog entry fields that included Title, Cover Photo, Author, Area of Field and Description. 3. The Counsellors fill the fields with the necessary information. 4. The Counsellors press the submit button. 5. The system sends the blog details to the admin for verification. 6. The counsellors receive notification for the approval or rejection of blogs. |
| **EXCEPTIONS** | **3a-** The information entered in the fields is against the format or limit.  **3a-1** The Error message is displayed showing the correct format.  **4a-** The submit button is not displayed.  **4a-1** The Counsellor is prompted for missing required fields. |
| **PRIORITY** | High |
| **Owner** | Payal Rawat |

|  |  |
| --- | --- |
| **ID** | **UC-8** |
| **NAME** | Delete Blogs |
| **PRIMARY ACTORS** | Counsellors |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | The Counsellors delete blogs by clicking the trash icon. |
| **PRECONDITIONS** | The Counsellors must be registered and logged into the system. |
| **POSTCONDITIONS** | The blog is successfully deleted from the database. |
| **NORMAL FLOW** | 1. The Counsellors navigate to the “Show Blogs” option from their dashboard. 2. The system displays the Blog title and the date of creation of blogs. 3. The Counsellors press the trash icon. 4. A dialog box opens that asks to confirm deletion. 5. The Counsellors press the “Yes, sure” button. |
| **PRIORITY** | Medium |
| **Owner** | Payal Rawat |

|  |  |
| --- | --- |
| **ID** | **UC-9** |
| **NAME** | Update Blogs |
| **PRIMARY ACTORS** | Counsellors |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | The Counsellors update blogs by clicking the Edit icon. |
| **PRECONDITIONS** | The Counsellors must be registered and logged into the system. |
| **POSTCONDITIONS** | The blog is successfully updated and stored in the database. |
| **NORMAL FLOW** | 1. The Counsellors navigate to the “Show Blogs” option from their dashboard. 2. The system displays the Blog title and the date of creation of blogs. 3. The Counsellors press the edit icon. 4. The blog entry fields displayed with the pre-populated content. 5. The Counsellors make the necessary modifications. 6. The Counsellors press the submit button. |
| **EXCEPTIONS** | **5a-** The modifications made are against the format or limit.  **5a-1** The Error message is displayed showing the correct format. |
| **PRIORITY** | Medium |
| **Owner** | Payal Rawat |

|  |  |
| --- | --- |
| **ID** | **UC-10** |
| **NAME** | Offer Counselling |
| **PRIMARY ACTORS** | Counsellors |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | The Counsellors register by sharing their personal information and professional experiences to offer counselling services. |
| **PRECONDITIONS** | The Counsellors have read the instructions and eligibility criteria. |
| **POSTCONDITIONS** | The Counsellors detail is successfully sent to the admin for verification. |
| **NORMAL FLOW** | 1. The Counsellors press the “Offer Counselling” button from the Home Page. 2. The Registration form appears. 3. The Counsellors enter their name, email address, phone number, gender, CNIC no., CNIC front and back photos, qualification with transcripts and working experience with certificates. Adding the profile photo is optional. 4. The counsellors submit their request by clicking the “Send Request” button. |
| **EXCEPTIONS** | **3a-** The Counsellor is entering data that is against the format of the fields. **3a-1** The Error message is displayed that shows the correct format.  **5a-** The “Send Request” button is not displayed.  **5a-1** The Counsellor is prompted for missing required fields. |
| **PRIORITY** | High |
| **Owner** | Payal Rawat |

|  |  |
| --- | --- |
| **ID** | **UC-11** |
| **NAME** | Approve Blogs |
| **PRIMARY ACTORS** | Admin |
| **SECONDARY ACTORS** | Counsellor |
| **DESCRIPTION** | Admin approves the blogs that are added by counsellors so that website contains only the verified blogs. |
| **PRECONDITIONS** | Admin login into the system. |
| **POSTCONDITIONS** | Admin may approve/disapprove the blogs. If admin approves the blogs, then these blogs are displayed on website otherwise, they are discarded. |
| **NORMAL FLOWS** | 1. Firstly, Admin login into the system and go to approve blogs option. 2. System displays the added blogs information. 3. Blogs information contain title, author, description and cover photo of blogs. 4. Admin approves the blogs by clicking on the approve button and system will send the confirmation notification to counsellor. |
| **EXCEPTIONS** | **4a** Wrong/Irrelevant Blog Content:  **4a-1** Admin decline the blog by giving a proper reason of rejection. **4a-2** System sends the rejection notification reason to counsellor. |
| **PRIORITY** | High |
| **Owner** | Ravi |

|  |  |
| --- | --- |
| **ID** | **UC-12** |
| **NAME** | Approve Counsellors |
| **PRIMARY ACTORS** | Admin |
| **SECONDARY ACTORS** | Counsellor |
| **DESCRIPTION** | Admin approves the counsellors whenever they register themselves in the system. |
| **PRECONDITIONS** | Admin login into the system. |
| **POSTCONDITIONS** | Admin may approve/disapprove the counsellors. If admin approves the counsellors, then they will able to access the system. |
| **NORMAL FLOWS** | 1. Firstly, Admin login into the system and go to approve counsellor’s option. 2. System displays the unapproved counsellor’s information. 3. Counsellors’ information contains their personal data, education info, and working experience. 4. Admin approves the counsellors by clicking on the approve button and system will send the confirmation email to counsellor. |
| **EXCEPTIONS** | **2a** Wrong/Irrelevant Details:  **2a-1** Admin reject the counsellor by giving a proper reason of rejection. **2a-2** System sends the reject reason to counsellor through email. |
| **PRIORITY** | High |
| **Owner** | Ravi |

|  |  |
| --- | --- |
| **ID** | **UC-13** |
| **NAME** | Approve Reviews |
| **PRIMARY ACTORS** | Admin |
| **SECONDARY ACTORS** | Users |
| **DESCRIPTION** | Admin approves the reviews given by the user. Website only displays the reviews that are approved by the admin. |
| **PRECONDITIONS** | Admin login into the system. |
| **POSTCONDITIONS** | Admin may approve/disapprove the reviews. If admin approves the reviews, then they will be displayed on the website. |
| **NORMAL FLOWS** | **1-** Firstly, Admin login into the system and go to approve reviews option. **2-** System displays the unapproved reviews list to admin.   1. Reviews contain users’ name, email and their reviews. 2. Admin approves the reviews by clicking on the approve button. |
| **EXCEPTIONS** | **2a** Wrong/Irrelevant Reviews:  **2a-1** Admin rejects the reviews that are irrelevant in order to maintain the quality and integrity of website. |
| **PRIORITY** | Medium |
| **Owner** | Ravi |

|  |  |
| --- | --- |
| **ID** | **UC-14** |
| **NAME** | View Counsellors Report |
| **PRIMARY ACTORS** | Admin |
| **SECONDARY ACTORS** | Counsellor |
| **DESCRIPTION** | System shows the counsellors report to admin. Admin also have the ability to remove the counsellors from the system. |
| **PRECONDITIONS** | Admin login into the system. |
| **POSTCONDITIONS** | System shows the list of all registered counsellors of the system. |
| **NORMAL FLOWS** | **1-** Firstly, Admin login into the system and go to view counsellor’s option. **2-** System displays the list of registered counsellors to admin.  **3-** Admin will also have the ability to remove the counsellors from the system. |
| **PRIORITY** | Low |
| **Owner** | Ravi |

|  |  |
| --- | --- |
| **ID** | **UC-15** |
| **NAME** | View Users Report |
| **PRIMARY ACTORS** | Admin |
| **SECONDARY ACTORS** | User |
| **DESCRIPTION** | System shows the users report to admin. Admin also have the ability to remove the users from the system. |
| **PRECONDITIONS** | Admin login into the system. |
| **POSTCONDITIONS** | System shows the list of all registered users of the system. |
| **NORMAL FLOWS** | 1. Firstly, Admin login into the system and go to view user’s option. 2. System displays the list of registered users to admin. 3. Admin will also have the ability to remove the users from the system. |
| **PRIORITY** | Low |
| **Owner** | Ravi |

|  |  |
| --- | --- |
| **ID** | **UC-16** |
| **NAME** | Log out |
| **PRIMARY ACTORS** | Users, Counsellors |
| **SECONDARY ACTORS** | Administrator |
| **DESCRIPTION** | Users or counsellors, after accessing career counselling services on the website, choose to log out of their accounts, ending their session. |
| **PRECONDITIONS** | User or counsellor must be logged into the system. |
| **POSTCONDITIONS** | User or counsellor is successfully logged out of the system and will need to log in again to access career counselling services. |
| **NORMAL FLOW** | 1. User navigates to the account or profile settings. 2. User selects the "Log Out" option from the available settings. 3. System displays a confirmation message that the user has been successfully logged out and may provide an option to return to the main screen or exit the application. |
|  | **2a-** User cancels the logout action. |
| **EXCEPTIONS** | **2a-1** The system returns them to the previous state without logging them out.  **3a-** Network or server issues during the logout process.  **3a-1** System should provide an error message to the user and advise them to try again later.  **3a-2** User remains logged in until a successful logout is achieved. |
| **PRIORITY** | High |
| **Owner** | Prateek Jadon |

## Functional Requirements

* The students will be able to register and log into the system securely.
* The AI chatbot will offer career recommendations tailored to students at the intermediate level, suggesting bachelor's degree programs based on their interests.
* Administrators will possess the authority to review and approve counsellors’ information, assess and authorize blog content, as well as approve user reviews.
* Qualified counsellors will be able to register their accounts and will provide career guidance to students.
* Users will also receive timely notifications regarding counselling sessions.
* Users will be able to provide feedback and ratings for counselling sessions to evaluate counsellors’ performance.
* Administrators will also have the ability to approve and manage informative blog content for users.

## Non-Functional Requirements

### Performance

 When the students will ask queries from CareerGPT, the system will respond to their requests within 1 minute due to excessive computation of the LLM model (Meta LLAM-2).

### Safety

* The chatbot will provide best career suggestions to students according o their pre-trained features.
* The system will also make sure that the users will properly communicate with counsellors through our platform in case of any queries.

### Reliability

* The system can work efficiently without failures apart from network failure.
* The students as well as counsellors can rely on the system actions.
* The system will take care the conversation secrecy between the counsellors and users.

### Robustness

* The system will response quickly to every user regardless of the number of users accessing the system at the same time.
* The system can track large amount of data and store it in the database.

### Interoperability

* The system has the ability to interact with other applications.
* The interaction with other applications will not affect the working of system.

## External Interface Requirements

### User Interfaces Requirements

* Use fonts like Philosopher, with appropriate sizes (minimum 14px for body text).
* Implement a color scheme (light pink #FED9E8, lavender #E7BCDE, purple #BB9CC0, blue #67729D) that is visually appealing and provides good contrast for readability.
* Use intuitive icons for navigation and actions (e.g., a house icon for the home page, a magnifying glass for search).
* Include relevant images to enhance the visual appeal and user engagement, such as career-related graphics.
* Design for a minimum screen resolution of 1024x768 pixels, ensuring compatibility with larger screens and responsive design for mobile devices.
* Include standard buttons such as "Submit," "Cancel," "Next," and "Previous" on forms and dialogs.
* Provide a persistent navigation bar with links to major sections: Home, Chatbot, Blogs, Counsellor, Notifications, Reviews & Ratings.

### Software interfaces

**SI-1: AI Chatbot Integration**

* **SI-1.1:** The Career Counselling Portal shall integrate with an AI Chatbot module for personalized career guidance. The chatbot will provide real-time responses to user queries, leveraging a pre-trained model (Meta LLAMA-2).
* **SI-1.2:** The AI Chatbot will seamlessly interact with the user interface to facilitate smooth communication between users and the system. **SI-1: AI Chatbot Integration**

* **SI-1.1:** The Career Counselling Portal shall integrate with an AI Chatbot module for personalized career guidance. The chatbot will provide real-time responses to user queries, leveraging a pre-trained model (Meta LLAMA-2).
* **SI-1.2:** The AI Chatbot will seamlessly interact with the user interface to facilitate smooth communication between users and the system.

**SI-2: Database Management System**

* **SI-2.1:** The portal shall connect to a relational database to manage user data, counsellor information, blogs, reviews, and notifications.
* **SI-2.2:** The database will store user credentials securely, including encrypted passwords and personal information.

**SI-3: Notification System**

* **SI-3.1:** The system shall interface with a notification service to alert users about messages from counsellor.  **SI-3.2:** Notifications will be sent in real-time to ensure timely communication.

**SI-4: Third-Party Chat Integration (SendBird)**

* **SI-4.1:** The Career Counselling Portal shall integrate with SendBird, a third-party chat service, to facilitate real-time communication between users and counsellor.
* **SI-4.2:** SendBird will provide robust chat functionalities, including text ensuring a seamless and secure interaction platform.

### Hardware interfaces

* The system is hosted on a high-performance server to handle multiple user requests simultaneously.
* The server hardware includes a multi-core processor, substantial RAM, and SSD storage to ensure quick response times and data retrieval.
* The portal is accessible via various user devices, including desktop computers, laptops, tablets, and smartphones.
* The system supports multiple screen resolutions and layouts to provide a consistent user experience across devices.
* Browsers supported include the latest versions of Chrome, Firefox, Safari, and Edge.
* The system relies on reliable networking hardware such as routers, switches, and firewalls to maintain secure and stable internet connections.
* High-speed internet connectivity is essential for real-time interactions between users and counsellor.
* Users may interact with the system using standard input devices such as keyboards and mice.

### Communications interfaces

**CI-1: Email Notifications**

**Description**: The system shall send email notifications to users and counsellor for various events.

**Purpose**: To keep users and counsellor informed about important actions and updates.

**Functionality**:

* Notifications for new messages from counsellor.
* Approval notifications for counsellor registrations.

**CI-2: Web Browsers**

**Description**: The system shall be accessible via standard web browsers.

**Purpose**: To ensure cross-platform compatibility and accessibility.

**Supported Browsers**: Chrome, Firefox, Safari, and Edge.

**Functionality**:

* Users shall access all features of the portal, including registration, chatting, and viewing blogs through a web browser.
* The user interface shall be responsive and work on different screen sizes and devices.

**CI-3: Network Protocols**

**Description**: The system shall utilize standard network protocols for communication.

**Purpose**: To ensure secure and reliable data transmission.

**Protocols**:

* HTTPS for secure communication between the client and server.
* SMTP for sending email notifications.

**CI-4: Chat Integration with SendBird**

**Description**: The system shall integrate with the SendBird platform for real-time chat functionality between users and counsellor.

**Purpose**: To facilitate seamless and efficient communication.

**Functionality**:

* Real-time messaging between users and counsellor.
* Notification of new messages.
* Storage and retrieval of chat history.

**3.6 Conclusion**

In conclusion, detailed analysis and consideration of user classes, coupled with stringent requirements identification techniques, the system encompasses a wide array of functional and non-functional features aimed at ensuring its effectiveness, reliability, and security. With seamless integration of AI chatbot capabilities, realtime communication channels, and robust performance targets, the portal emerges as a valuable resource in guiding students through their career planning journey while providing counsellors and administrators with the tools necessary for efficient platform management.

Chapter 4

# Design and Architecture

## Architectural Design

**Scenario 1: Website Visit**

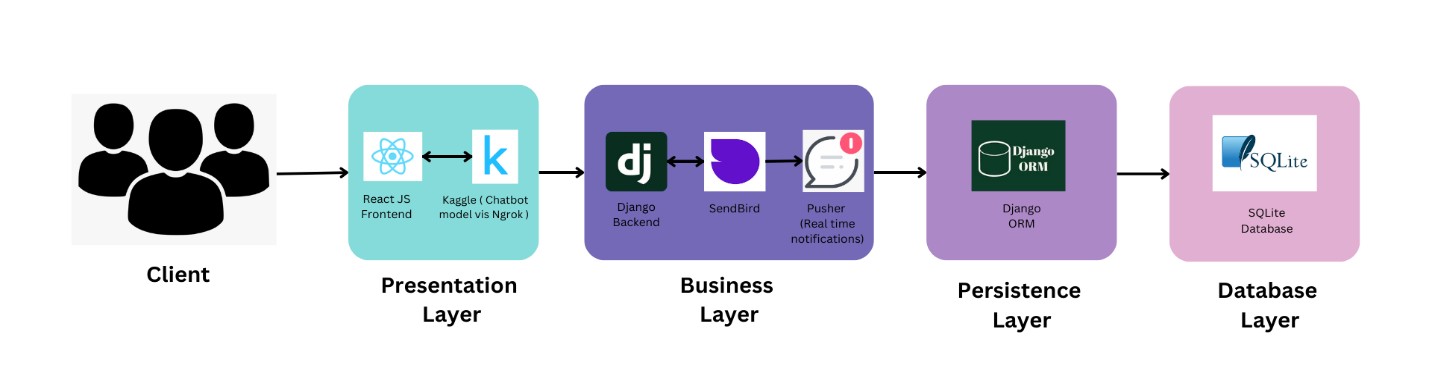
Users who simply visit the website will interact solely with the Presentation Layer. This layer is implemented using React JS for the frontend.

**Scenario 2: Using CareerGPT Feature**

1. ***Login/Signup****:*
   * When a user wants to use the CareerGPT feature (an AI model providing career suggestions to students), they need to log in or sign up first.
   * The request from the Presentation Layer (React JS) will be sent to the Django server.
   * The Django server will handle the request and save the user data in the SQLite database via the Django ORM.
2. ***Using Career GPT****:*
   * After logging in, the user can use the CareerGPT feature.
   * The React frontend will call the API of the CareerGPT model, hosted on Kaggle, through Ngrok.
   * The model will process the user’s query and respond accordingly.
   * This interaction involves continuous queries and responses between the user and the model.

**Scenario 3: Chat Functionality**

1. ***Initiating Chat****:* 
   * When a user wants to use the chat functionality, the request will go from the Presentation Layer (React JS) to the Django server in the Business Layer.
   * The Django server will then forward the request to the SendBird server.
   * SendBird will create a user and a channel for the chat and return the data to the Django server.
   * The React frontend will then display this data to the user.
2. ***Real-Time Notifications****:* 
   * Pusher is used for real-time notifications. It employs WebSockets in the background.
   * When a user or counsellor sends a message, SendBird triggers a webhook (an HTTP callback) to the Django server.
   * The Django server receives this data and uses Pusher to publish it on a specific channel.
   * The React frontend subscribes to this channel, receiving the data and displaying it to the user.
   * This mechanism ensures that both users and counsellor are notified in real-time about new messages.



**Figure 2: Architectural Diagram**

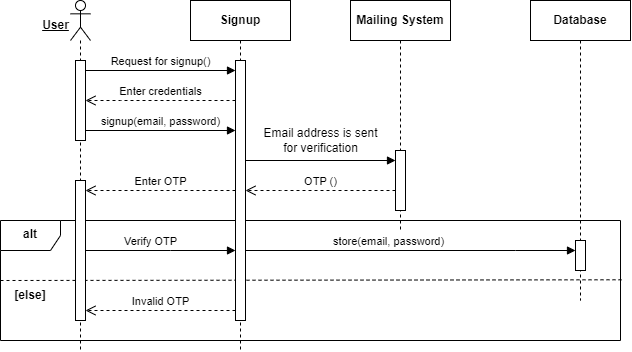
## Design Models

***Design Models for Object Oriented Development Approach***

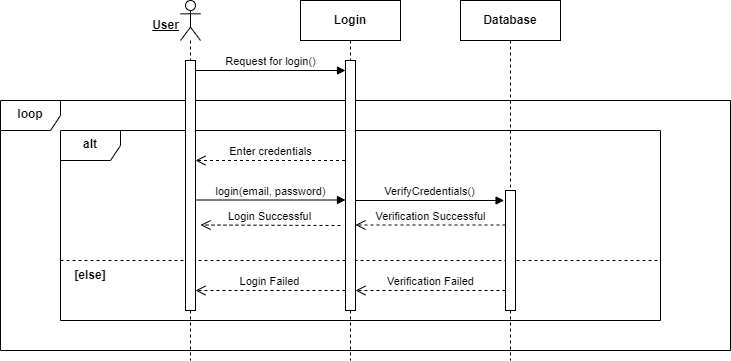
**Sequence Diagrams**

* **Signup**

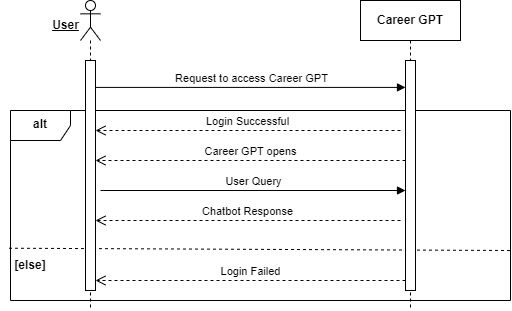
**Figure 3: Sequence Diagrams**



* **Login**



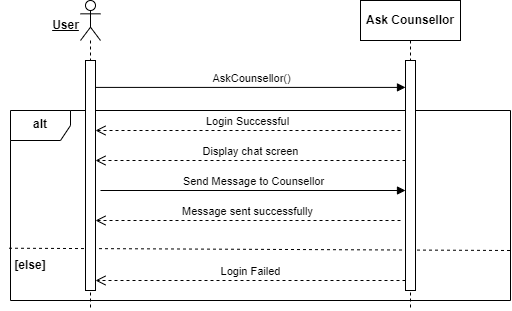
* **Interaction with Chatbot**



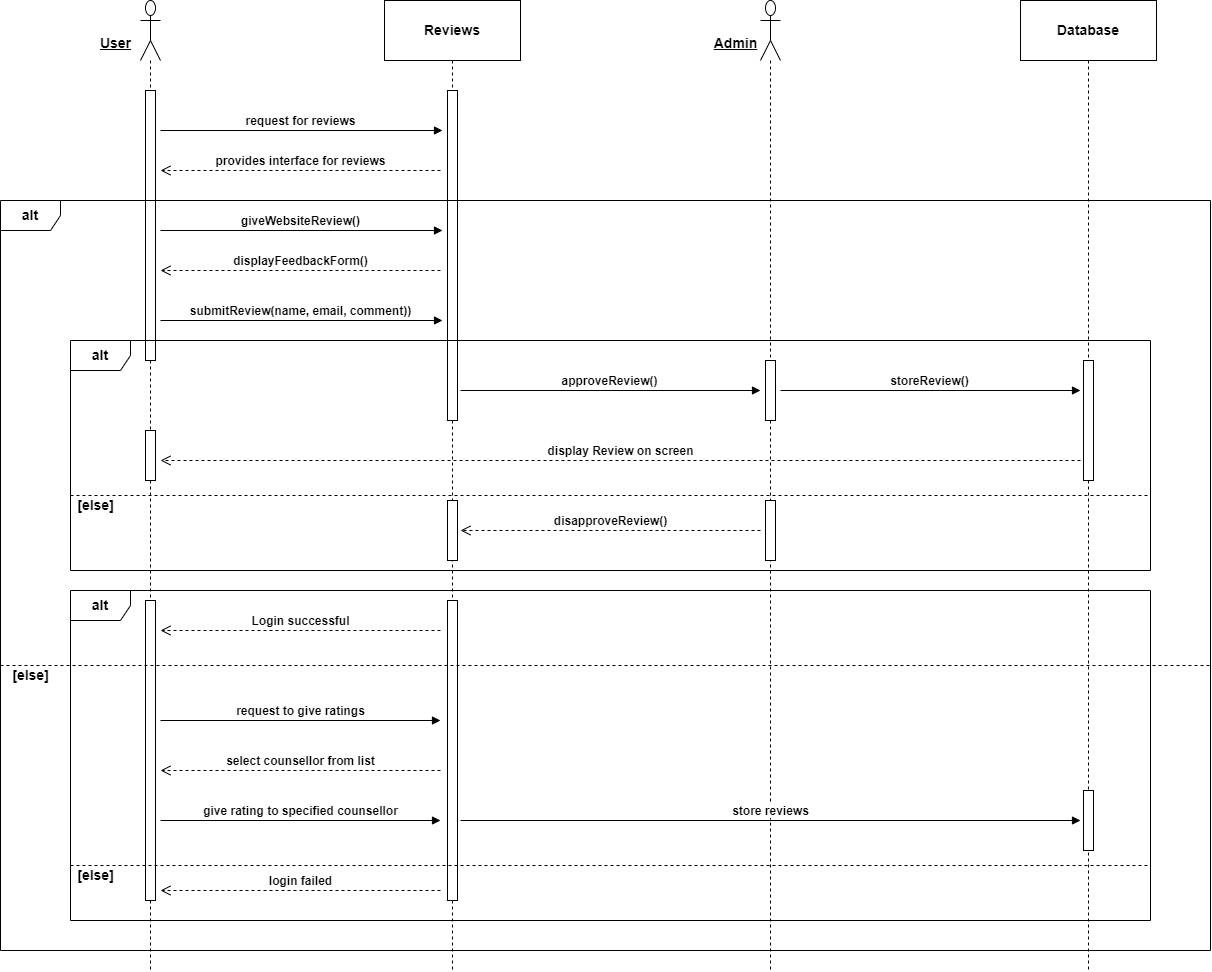
* **Search Blogs**



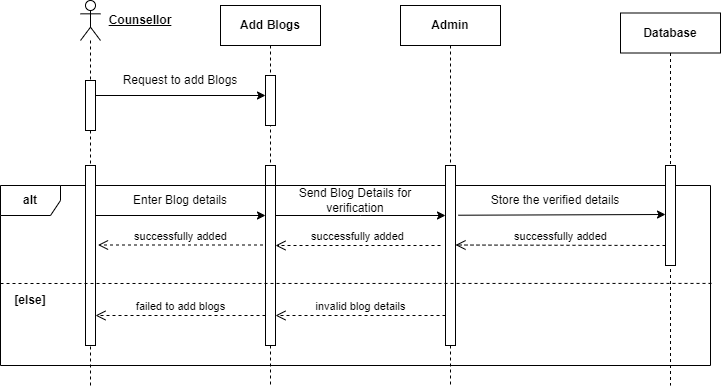
* **Ask Counsellor**



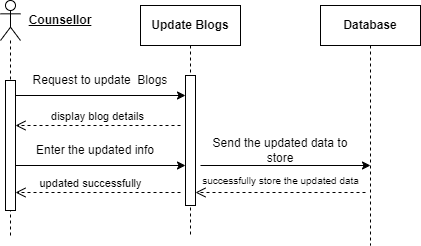
* **Give Reviews/Ratings**



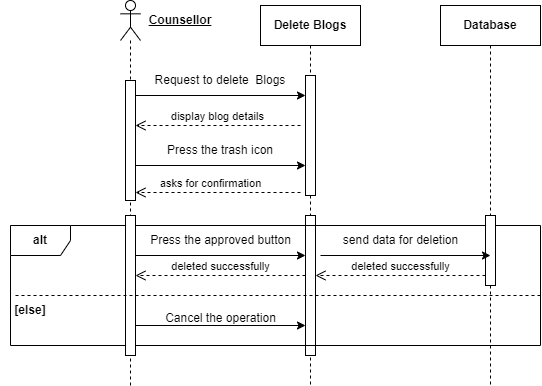
* **Add Blogs**



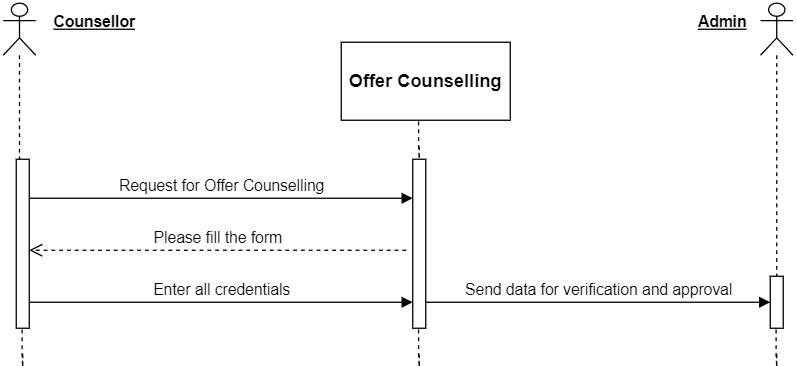
* **Update Blogs**



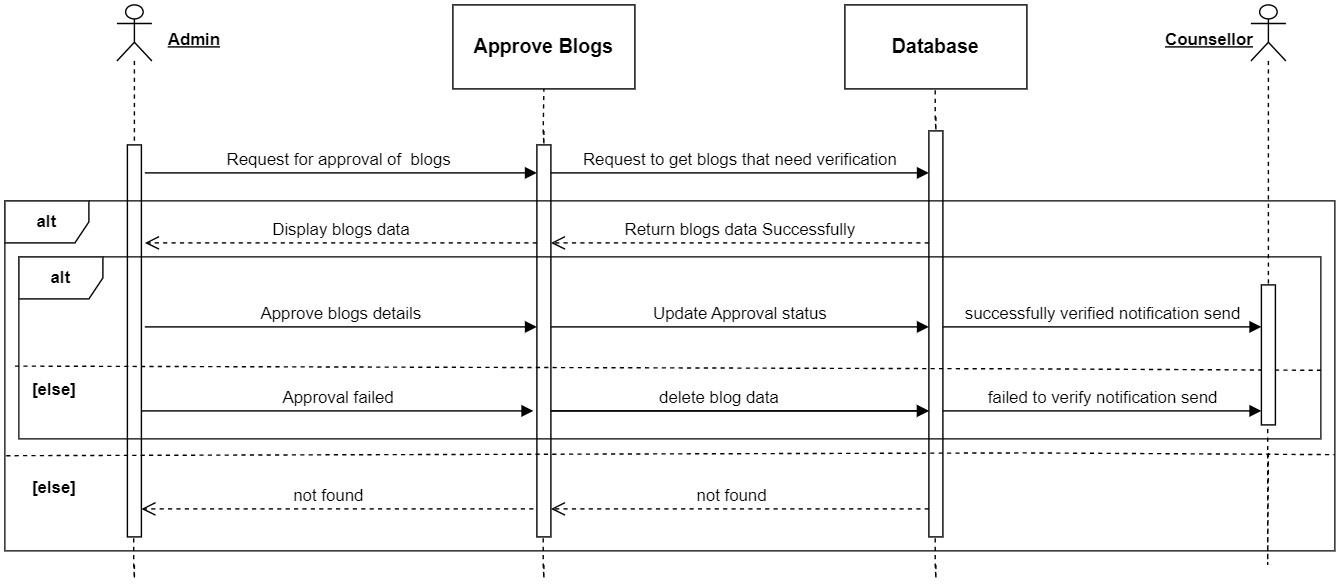
* **Delete Blogs**



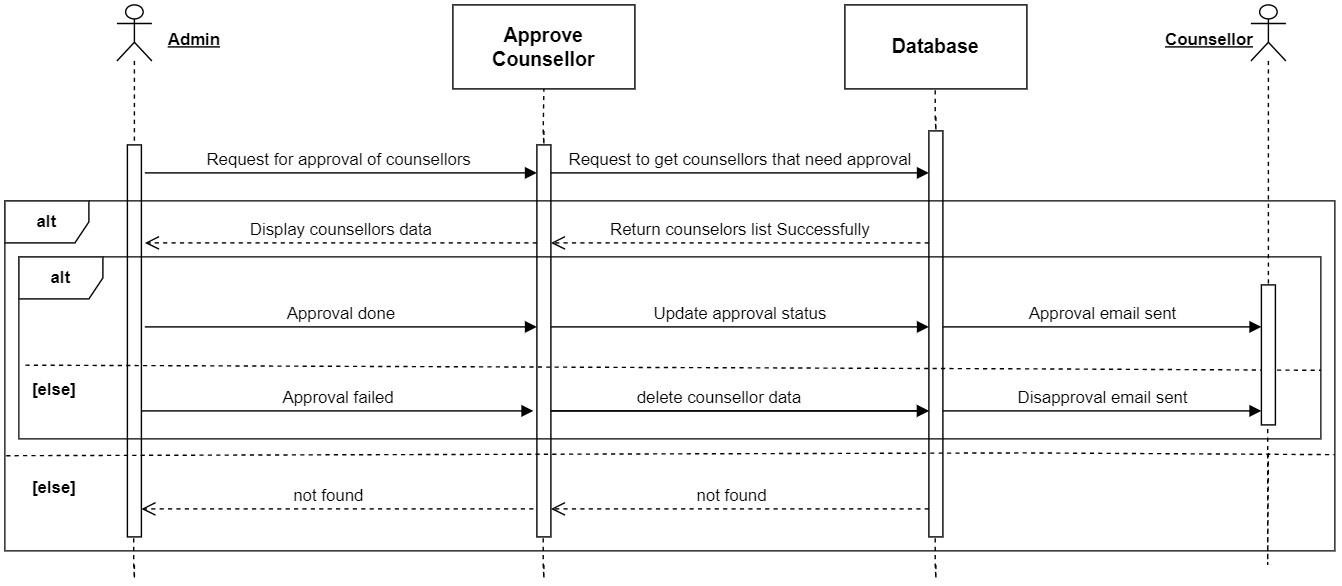
* **Offer Counselling**



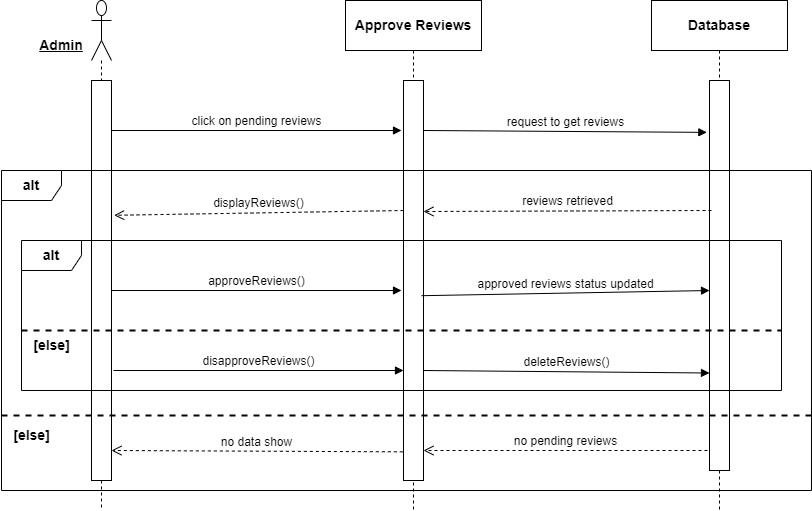
* **Approve Blogs**



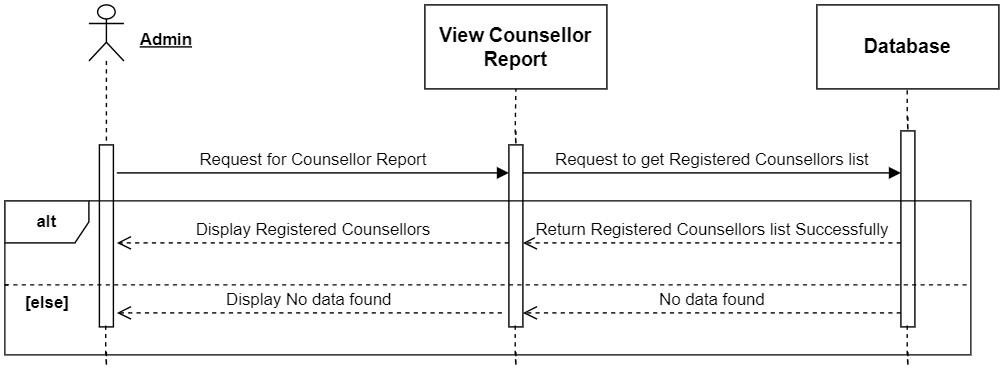
* **Approve Counsellors**



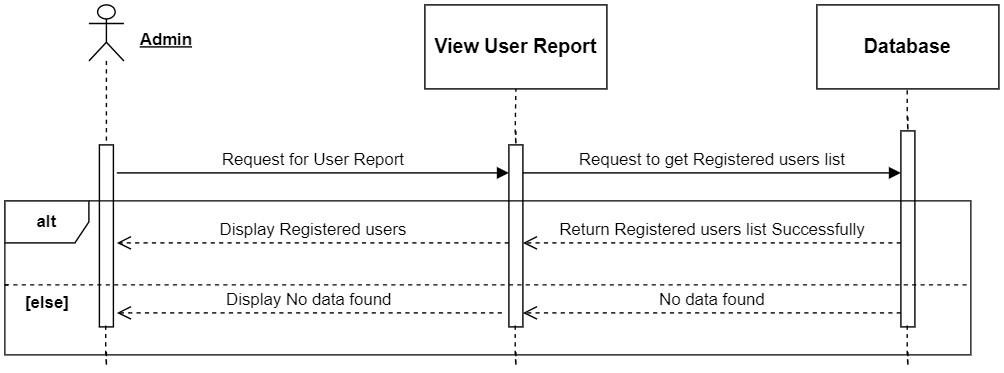
* **Approve Reviews**



* **View Counsellors Report**



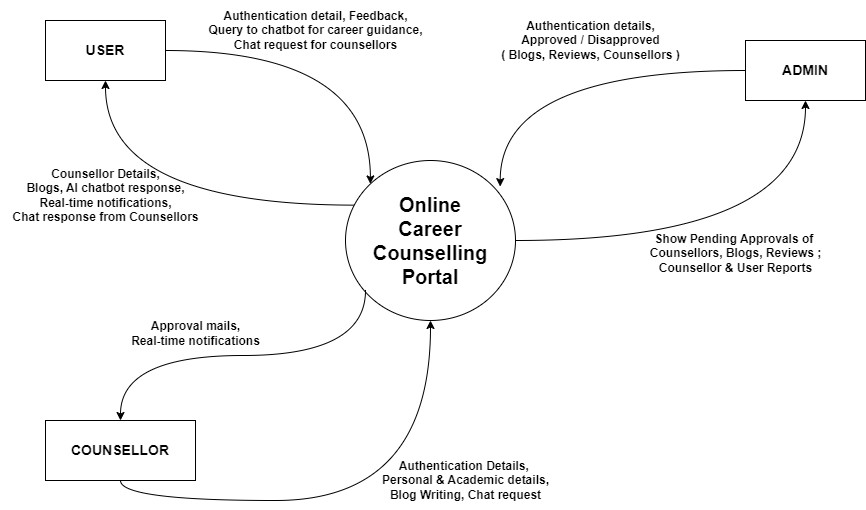
* **View Users Report**



***Design Models for Procedural Approach***

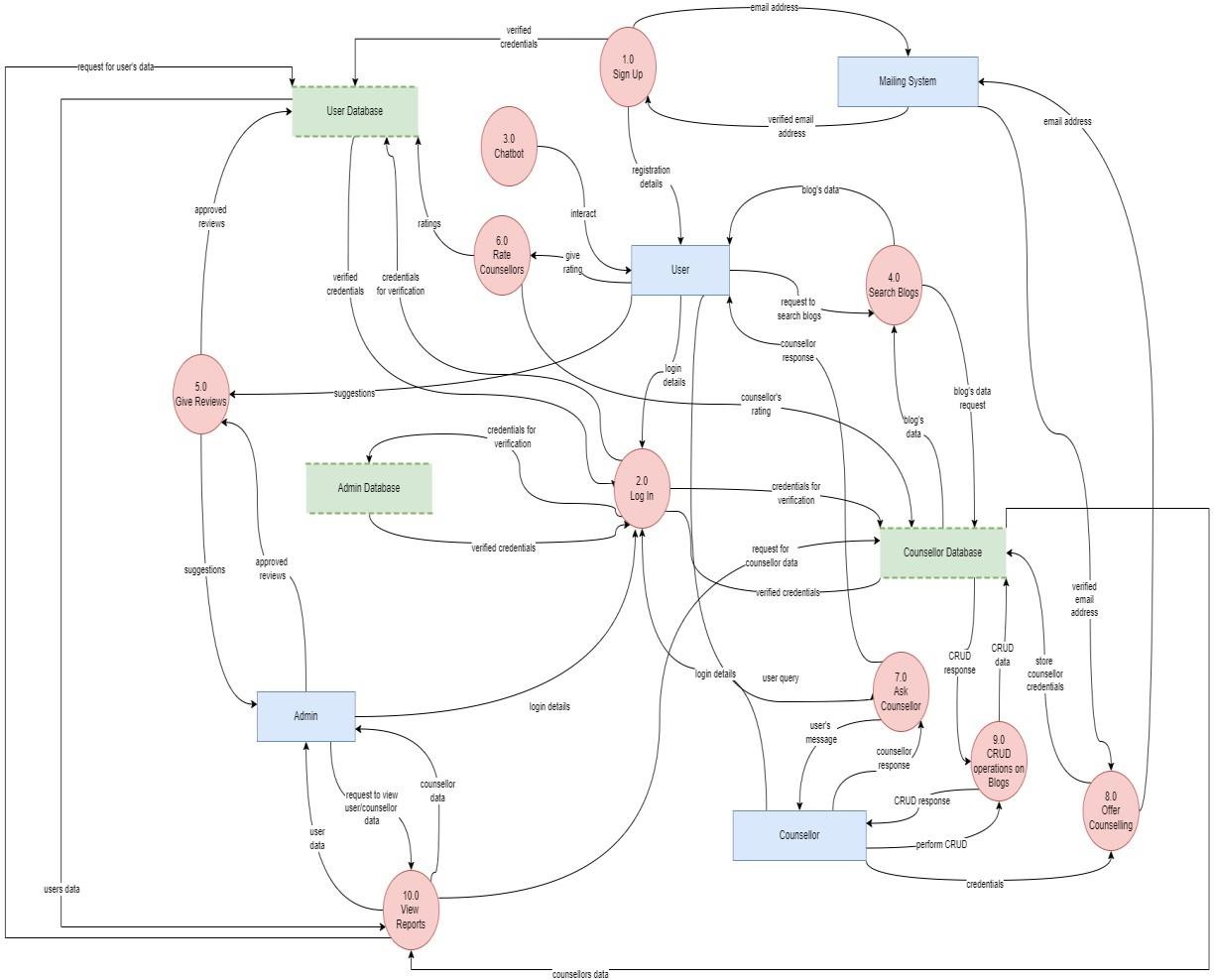
**Data Flow Diagrams**

* **Level 0**



**Figure 4: Context Diagram**

* **Level 1**



**Figure 5: Data Flow Digram**

## Data Design

In our career counselling portal, the information domain is transformed into structured data that supports the various functionalities of the system. The design focuses on storing, processing, and organizing data effectively to ensure smooth operation and efficient data retrieval.

***Major Data or System Entities***

1. **User Data (ACU)**:
   * Information about users who sign up and log in to the portal, including their name, email, password, role, and creation timestamp.
   * Stored in the SQLite database using the ACU model.
   * Processed for authentication, authorization, and interaction with AI models and chat functionalities.
2. **Counsellor Data**:
   * Additional details for users who are counsellor, including phone number, gender, CNIC, profile picture, CNIC images, approval status, and creation timestamp.
   * Stored in the Counsellor model and linked to the ACU model.
3. **Qualification Data**:
   * Academic qualifications of counsellor, including qualification details, field of study, transcript image, and creation timestamp.
   * Stored in the Qualification model and linked to the Counsellor model.
4. **Working Experience Data**:
   * Professional experience of counsellor, including institute name, starting and ending years, certificates image, and creation timestamp.
   * Stored in the Working Experience model and linked to the Counsellor model.
5. **Ratings and Reviews**:
   * Ratings and reviews given by users to counsellor, including rating value, review description, user details, approval status, and creation timestamp.
   * Stored in the Ratings and Reviews models, linked to the Counsellor and ACU models respectively.
6. **Blog Data**:
   * Blog posts written by counsellor, including title, author name, area of field, description, cover image, approval status, and creation date.
   * Stored in the Blogs model and linked to the Counsellor model.
7. **Career GPT Interaction History**:
   * History of interactions with the CareerGPT feature, including message ID, message content, message type (bot/user), and associated user.
   * Stored in the CareerGPTHistory model and linked to the ACU model.
8. **Chat Data**:
   * Messages exchanged between users and counsellor.

***Data Storage Items***

1. **SQLite Database**:
   * Stores all user-related data such as profiles, counsellor details, qualifications, working experiences, ratings, reviews, blogs, and CareerGPT history.  Utilizes Django ORM for handling database operations.
2. **SendBird Server**:
   * Manages chat-related data, including user creation, channel creation, and message storage.  Provides real-time communication features.
3. **Pusher**:
   * Handles real-time notifications for chat messages using WebSocket.
   * Publishes and subscribes to data channels to notify users of new messages.

### Data Dictionary

***Entities and Attributes***

**1. ACU (User)**

* **id** (Integer): Unique identifier for the user.
* **name** (Text): User's name.
* **email** (EmailField): User's email address, unique.
* **password** (Text): Encrypted password for the user.
* **role** (CharField): Role of the user, default is 'U'.
* **created\_at** (DateTimeField): Timestamp when the user was created. **2. Counsellor**
* **id** (Integer): Unique identifier for the counsellor.
* **counsellor\_id** (ForeignKey): Link to the ACU model.
* **phone\_no** (Text): Phone number of the counsellor.
* **gender** (CharField): Gender of the counsellor.
* **cnic** (CharField): CNIC of the counsellor.
* **profile\_pic** (Text): Profile picture URL.
* **cnic\_front\_img** (Text): CNIC front image URL.
* **cnic\_back\_img** (Text): CNIC back image URL.
* **is\_approved** (BooleanField): Approval status of the counsellor.
* **created\_at** (DateTimeField): Timestamp when the counsellor was created.

**3. Qualification**

* **counsellor\_id** (OneToOneField): Link to the Counsellor model.
* **qualification** (Text): Qualification details.

**field\_of\_study** (Text): Field of study.

* **transcript\_img** (Text): Transcript image URL.
* **created\_at** (DateTimeField): Timestamp when the qualification was created. **4. WorkingExperience**
* **id** (Integer): Unique identifier for the working experience.
* **counsellor\_id** (ForeignKey): Link to the Counsellor model.
* **institute\_name** (Text): Name of the institute.
* **starting\_year** (Text): Starting year of the experience.
* **ending\_year** (Text): Ending year of the experience.
* **certificates\_image** (Text): Certificates image URL.
* **created\_at** (DateTimeField): Timestamp when the experience was created.

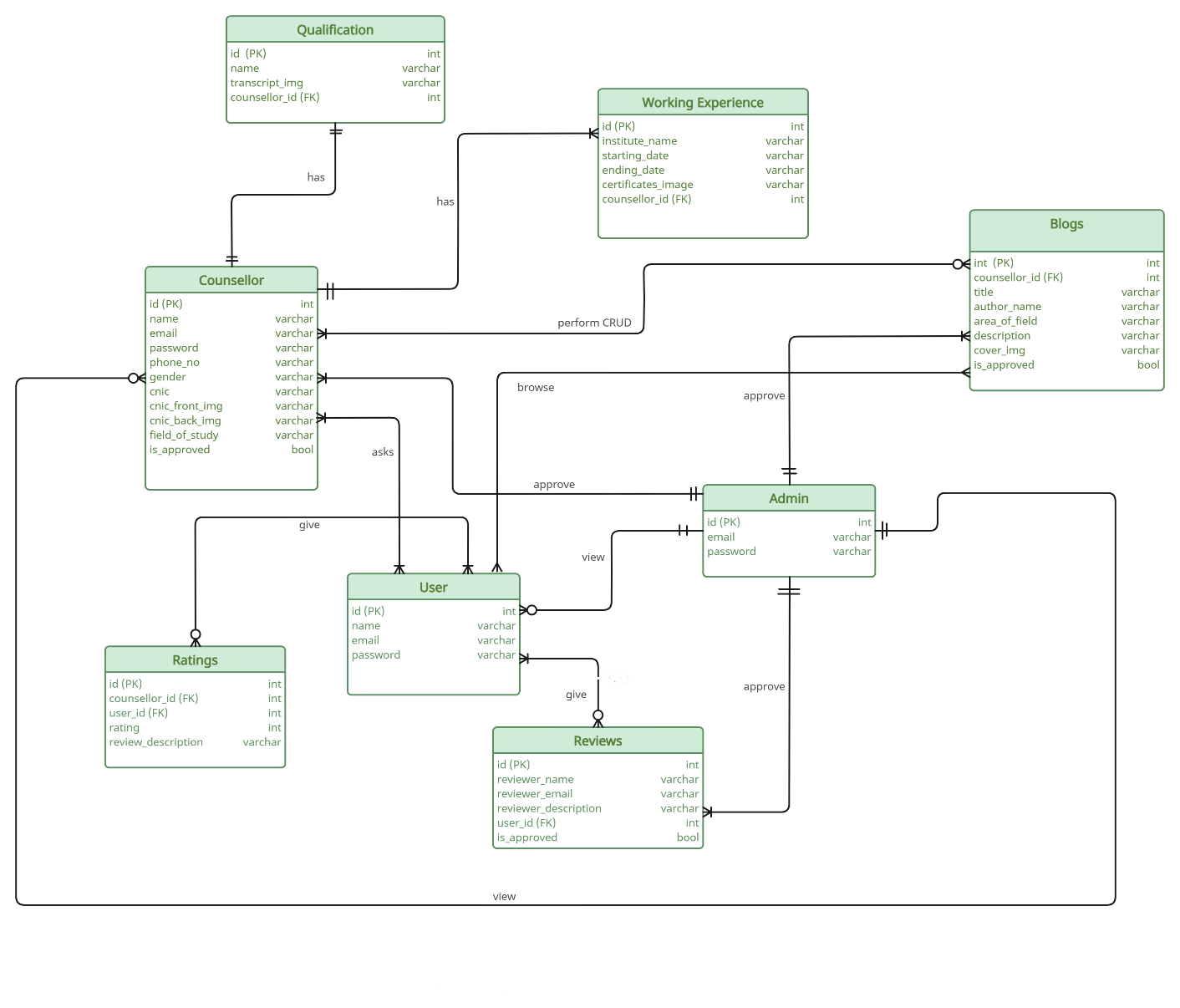
**5. Ratings**

* **id** (Integer): Unique identifier for the rating.
* **counsellor\_id** (ForeignKey): Link to the Counsellor model.
* **rating** (Integer): Rating value.
* **review\_description** (Text): Description of the review.
* **user\_id** (ForeignKey): Link to the ACU model.
* **created\_at** (DateTimeField): Timestamp when the rating was created. **6. Reviews**
* **id** (Integer): Unique identifier for the review.
* **user\_id** (ForeignKey): Link to the ACU model.
* **reviewer\_name** (Text): Name of the reviewer.
* **reviewer\_email** (EmailField): Email of the reviewer.
* **reviewer\_description** (Text): Description of the review.
* **is\_approved** (BooleanField): Approval status of the review.
* **created\_at** (DateTimeField): Timestamp when the review was created.

**7. Blogs**

* **id** (Integer): Unique identifier for the blog.
* **counsellor\_id** (ForeignKey): Link to the Counsellor model.
* **title** (Text): Title of the blog.
* **author\_name** (Text): Author's name.
* **area\_of\_field** (Text): Area of field.
* **description** (Text): Description of the blog.
* **cover\_image** (Text): Cover image URL.
* **is\_approved** (BooleanField): Approval status of the blog.
* **created\_at** (DateField): Date when the blog was created. **8. CareerGPTHistory**
* **id** (Integer): Unique identifier for the history record.
* **user\_id** (ForeignKey): Link to the ACU model.
* **message** (Text): Message content.
* **type** (CharField): Type of the message (bot/user).
* **created\_at** (DateTimeField): Timestamp when the history was recorded.

***Entity Relationship Diagram***



**Figure 6: Entity Relationship Diag**

***Methods and Parameters***

1. ***register User*** 
   * **Description:** Registers a new user.  **Method:** POST
   * **Parameters:** request: HTTP request object.
   * **Returns:** JSON response indicating success or error.
2. **Check Email**
   * **Description:** Checks if an email is already registered.
   * **Method:** POST 
   * **Parameters:** request: HTTP request object.
   * **Returns:** JSON response indicating if the email exists or not. **3. checkCounsellorEmail**
   * **Description:** Checks if a counsellor's email is already registered and returns their role.
   * **Method:** POST
   * **Parameters:** request: HTTP request object.
   * **Returns:** JSON response containing the role and existence status.
3. **Register Counsellor**
   * + **Description:** Registers a new counsellor and saves their details.
     + **Method:** POST
     + **Parameters:** request: HTTP request object.

 **Returns:** JSON response indicating success or error.

1. **Send VerificationEmail**
   * + **Description:** Sends a verification email to the counsellor.
     + **Method:** POST
     + **Parameters:** request: HTTP request object.

 **Returns:** JSON response indicating success or error.

1. **Login User**

 **Description:** Logs in a user.  **Method:** POST  **Parameters:** request: HTTP request object.

 **Returns:** JSON response indicating login success or failure.

1. **Get Session Data**
   * **Description:** Retrieves the current session data.
   * **Method:** GET
   * **Parameters:**

o request: HTTP request object.

* + **Returns:** JSON response containing session data.

1. **Delete Session Data**
   * **Description:** Deletes the current session data.
   * **Method:** GET
   * **Parameters:** o request: HTTP request object.
   * **Returns:** JSON response indicating success or error.
2. **Get Top Counsellors**
   * **Description:** Retrieves the top counsellors.  **Method:** GET
   * **Parameters:**

o request: HTTP request object.

* + **Returns:** JSON response containing top counsellors' data.

1. **Save Reviews**
   * **Description:** Saves a review.  **Method:** POST  **Parameters:**

o request: HTTP request object.

* + **Returns:** JSON response indicating success or error.

1. **Get Reviews**
   * **Description:** Retrieves all reviews.  **Method:** GET  **Parameters:**

o request: HTTP request object.

* + **Returns:** JSON response containing reviews data.

1. **Get Counsellors By UID**
   * **Description:** Retrieves counsellors by user ID.
   * **Method:** GET
   * **Parameters:**

o request: HTTP request object.

* + **Returns:** JSON response containing counsellors list.

1. **saveRatings**
   * **Description:** Saves a rating for a counsellor.
   * **Method:** POST  **Parameters:**

o request: HTTP request object.

* + **Returns:** JSON response indicating success or error.

1. **fetchBlogsData** 
   * + **Description**: Fetches data for all blogs.
     + **Method**: GET
     + **Parameters**: None
     + **Returns**: JSON response containing blogs data. **15. blogDetails**
     + **Description**: Retrieves details of a specific blog.
     + **Method**: POST
     + **Parameters**: o *id*: ID of the blog.
     + **Returns**: JSON response containing blog details and user email.
2. **Create Send Bird Channel**

 **Description**: Creates a Send Bird channel. 

**Method**: POST 

**Parameters**: o *counsellor Id*: ID of the counsellor.

*Counsellor Nick Name*: Nickname of the counsellor.

*Counsellor Profile URL*: Profile URL of the counsellor.  **Returns**: JSON response containing the channel URL.

1. **Save History**

 **Description**: Saves history data.  **Method**: POST  **Parameters**:

o *Data*: History data.

 **Returns**: JSON response containing the success or failure.

1. **Get History** 
   * **Description**: Retrieves history data.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing history data.
2. **Get Counsellor Data** 
   * **Description**: Retrieves data for a counsellor.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing counsellor data.
3. **Get Counsellor Cards Data** 
   * **Description**: Retrieves data for counsellor cards.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing counsellor cards data.
4. **Get Counsellor Profile Data** 
   * **Description**: Retrieves profile data for a counsellor.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing counsellor profile data.
5. **Get Counsellor Settings** 
   * **Description**: Retrieves settings data for a counsellor.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing counsellor settings data.
6. **Update Counsellor Settings** 
   * **Description**: Updates settings for a counsellor. 
   * **Method**: POST 
   * **Parameters**:
     + - *Phone No*: Phone number of the counsellor. o *password*: Password of the counsellor.
       - *Profile Pic*: Profile picture of the counsellor.
   * **Returns**: JSON response indicating success or error.
7. **Get Counsellor Blogs** 
   * **Description**: Retrieves blogs associated with a counsellor.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing blogs data.
8. **Add Blog** 
   * **Description**: Adds a new blog.  **Method**: POST  **Parameters**:
     + - *Add Blog Data*: Data for the new blog.
       - *cover\_image*: Cover image for the new blog.
   * **Returns**: JSON response indicating success or error.
9. **Edit Blog** 
   * **Description**: Edits an existing blog.  **Method**: POST  **Parameters**:
     + - *addBlogData*: Updated data for the blog. o *cover\_image*: Updated cover image for the blog.
       - *blogId*: ID of the blog to be edited.
   * **Returns**: JSON response indicating success or error.
10. **Delete Blog** 
    * **Description**: Deletes a blog.  **Method**: DELETE  **Parameters**: o *bid*: ID of the blog to be deleted.
    * **Returns**: JSON response indicating success or error.
11. **Get Users Count** 
    * **Description**: Retrieves the total count of users.
    * **Method**: GET
    * **Parameters**: None
    * **Returns**: JSON response containing the count of users.
12. **getBlogsCount** 
    * **Description**: Retrieves the total count of blogs.
    * **Method**: GET
    * **Parameters**: None
    * **Returns**: JSON response containing the count of blogs.
13. **getCounsellorsCount** 
    * **Description**: Retrieves the total count of counsellors.
    * **Method**: GET
    * **Parameters**: None
    * **Returns**: JSON response containing the count of counsellors.
14. **getReviewsCount** 
    * **Description**: Retrieves the total count of reviews.
    * **Method**: GET
    * **Parameters**: None
    * **Returns**: JSON response containing the count of reviews.
15. **getAdminProfile** 
    * **Description**: Retrieves profile data for the admin.
    * **Method**: GET
    * **Parameters**: None
    * **Returns**: JSON response containing the admin profile data.
16. **updateAdminProfile** 
    * **Description**: Updates profile data for the admin.  **Method**: PUT  **Parameters**:

o *profileForm*: Updated profile data.

* + **Returns**: JSON response indicating success or error.

1. **getUsers** 
   * **Description**: Retrieves a list of users.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing the list of users.
2. **deleteUser** 
   * **Description**: Deletes a user.  **Method**: POST  **Parameters**: o *selectedRow*: ID of the user to be deleted.
   * **Returns**: JSON response indicating success or error.
3. **getUnapprovedReviews** 
   * **Description**: Retrieves unapproved reviews.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing unapproved reviews.
4. **deleteReview** 
   * **Description**: Deletes a review.  **Method**: POST  **Parameters**:

o *selectedRow*: ID of the review to be deleted.

* + **Returns**: JSON response indicating success or error.

1. **approveReview** 
   * **Description**: Approves a review.  **Method**: PUT  **Parameters**:

o *selectedRow*: ID of the review to be approved.  **Returns**: JSON response indicating success or error.

1. **getUnapprovedBlogs** 
   * **Description**: Retrieves unapproved blogs.  **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing unapproved blogs.
2. **rejectBlog** 
   * **Description**: Rejects a blog.  **Method**: POST 
   * **Parameters**:

*blog\_id*: ID of the blog to be rejected.

o *counsellor\_email*: Email of the counsellor associated with the blog.  **Returns**: JSON response indicating success or error.

1. **approveBlog** 
   * **Description**: Approves a blog.  **Method**: PUT  **Parameters**:

o *blog\_id*: ID of the blog to be approved.

* + **Returns**: JSON response indicating success or error.

1. **getCounsellorsData** 
   * **Description**: Retrieves data for unapproved counsellors.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing data for unapproved counsellors.
2. **deleteCounsellor** 
   * **Description**: Deletes a counsellor.  **Method**: DELETE  **Parameters**: o *userEmail*: Email of the counsellor to be deleted.

o *rejectionReason*: Reason for the rejection.

* + **Returns**: JSON response indicating success or error.

1. **approveCounsellor** 
   * **Description**: Approves a counsellor.  **Method**: GET  **Parameters**:
     + - *userEmail*: Email of the counsellor to be approved.
       - *greetingMessage*: Greeting message for the approved counsellor.  **Returns**: JSON response indicating success or error.
2. **getApprovedCounsellors** 
   * **Description**: Retrieves data for approved counsellors.
   * **Method**: GET
   * **Parameters**: None
   * **Returns**: JSON response containing data for approved counsellors.
3. **sendBirdWebHook** 
   * **Description**: Receives webhook data from SendBird.
   * **Method**: POST
   * **Parameters**: None
   * **Returns**: JSON response indicating success or error.

## Human Interface Design

### Screen Images and Actions

* **Signup and validate account on a website**

**Figure 7: Screen Images**



**1**



Navigate

to



**http://127.0.0.1:5173/**

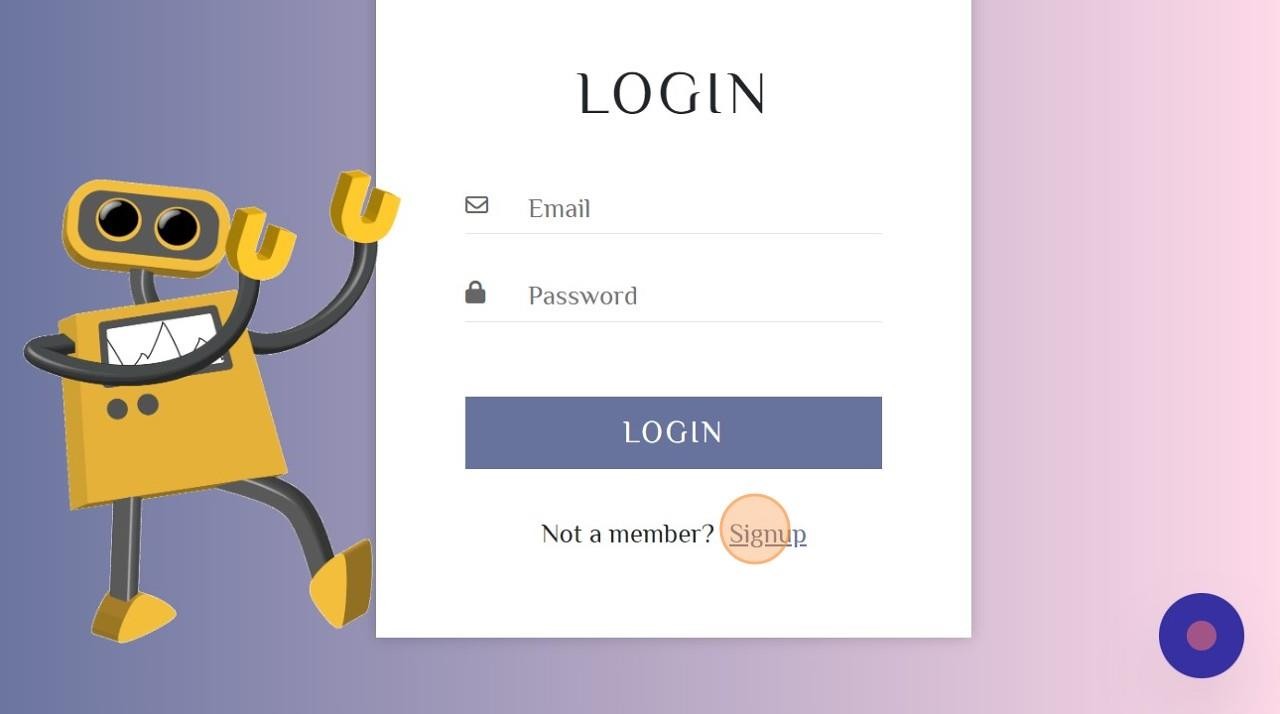
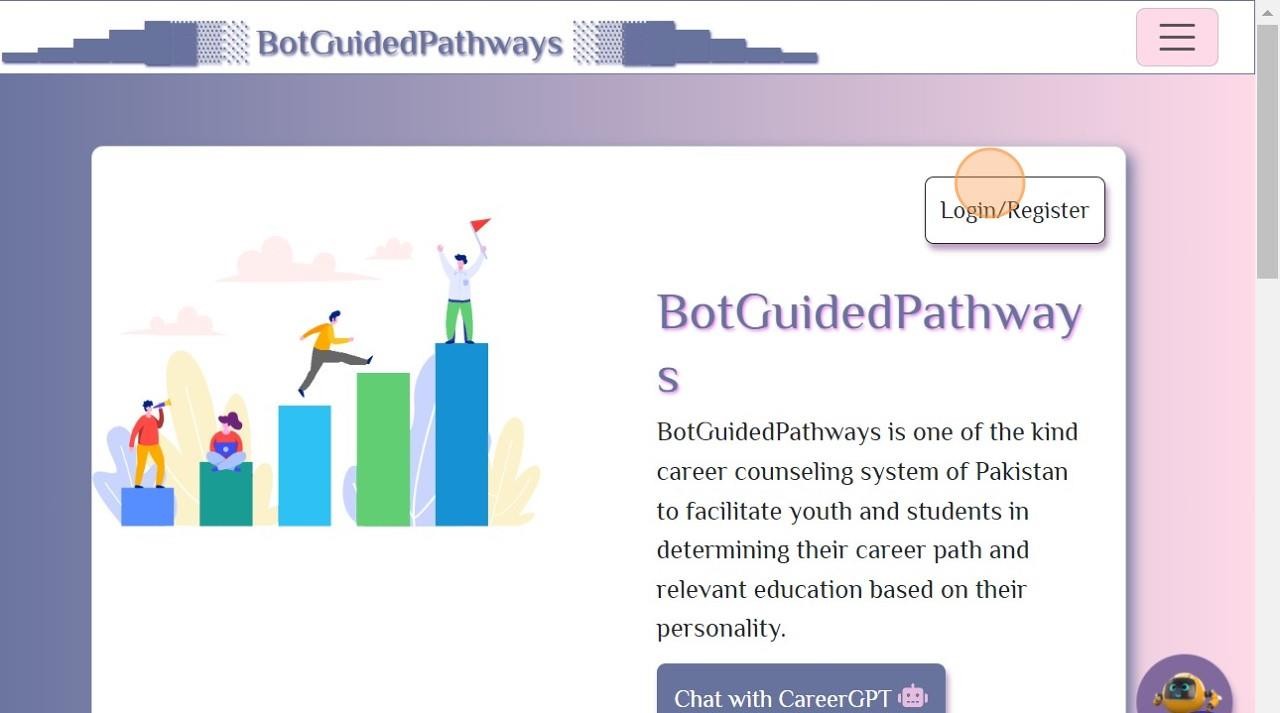


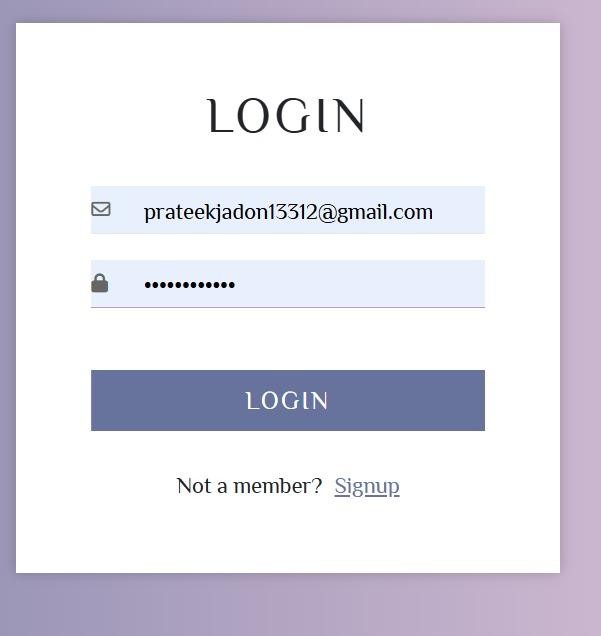
**2**

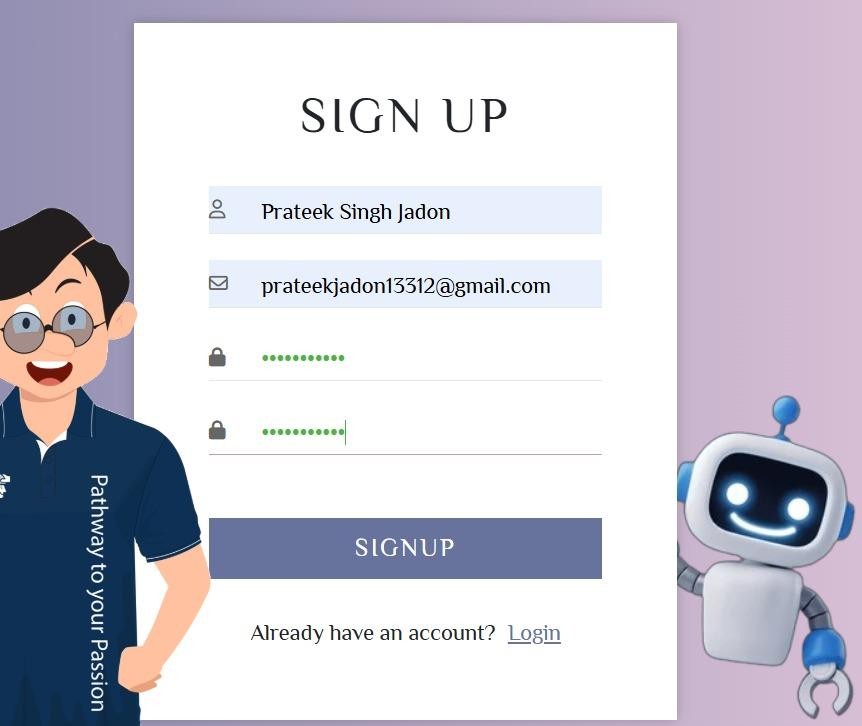


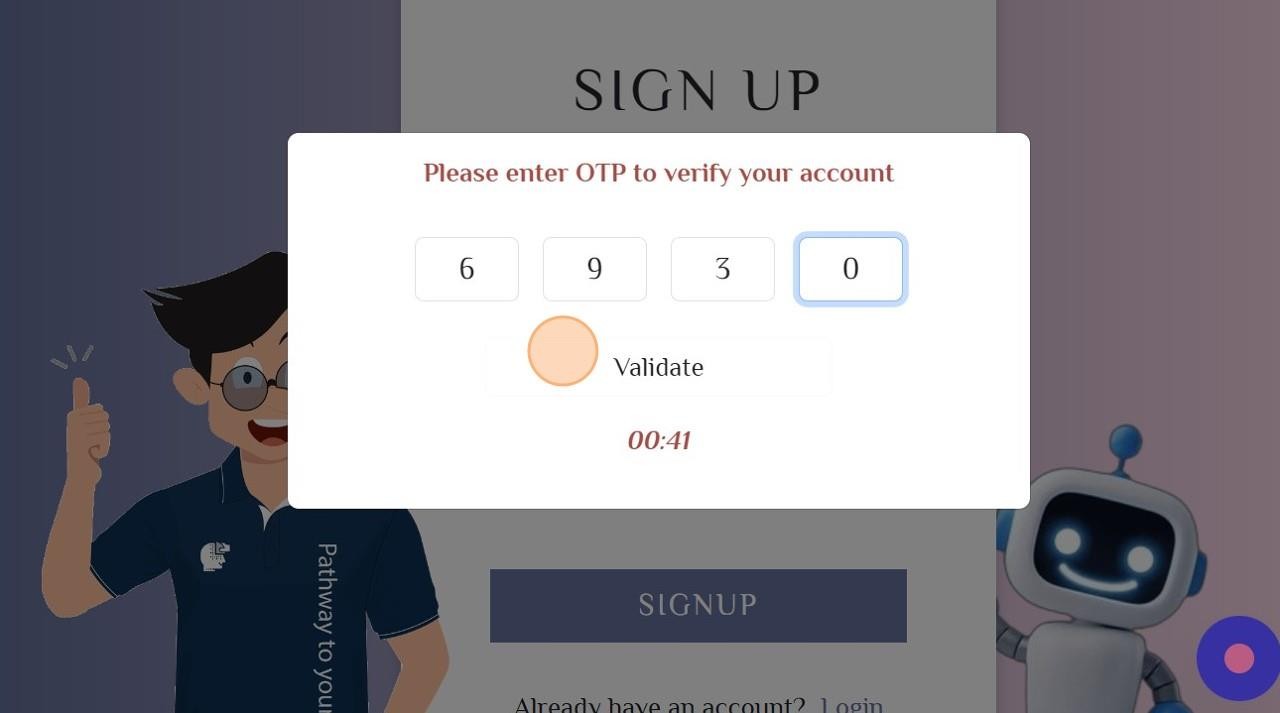
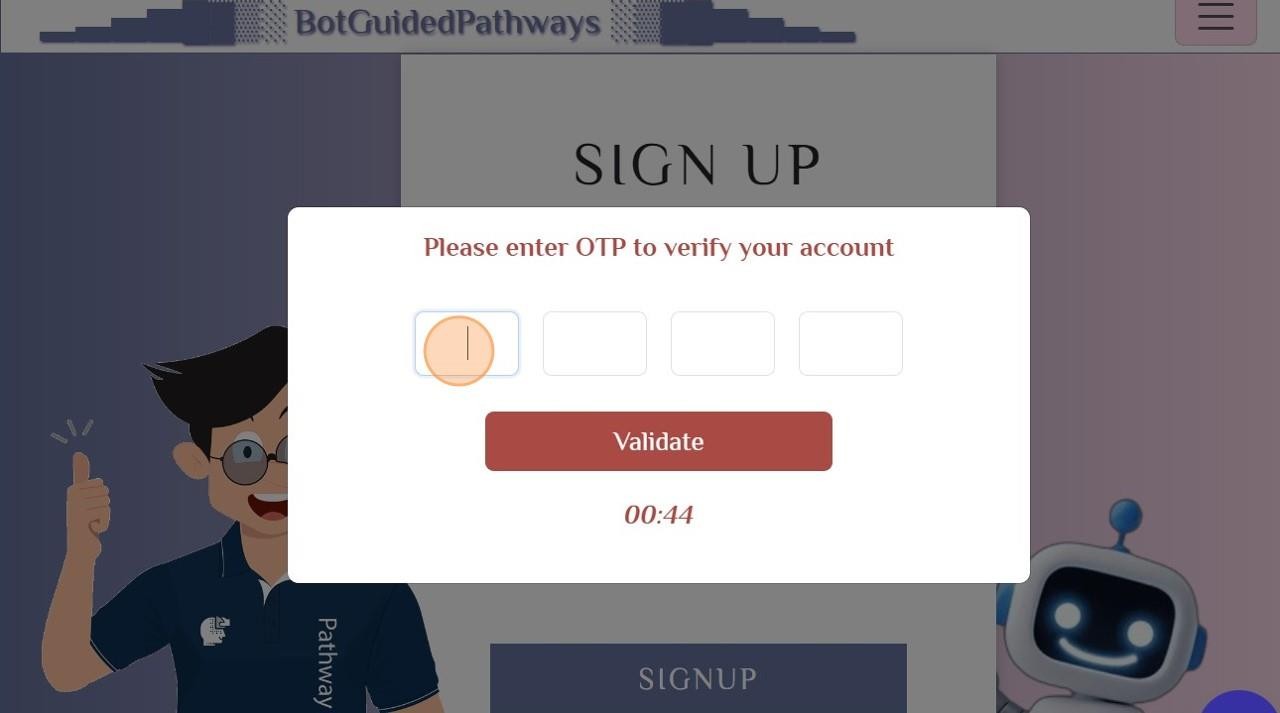
Click

"Login/Register"

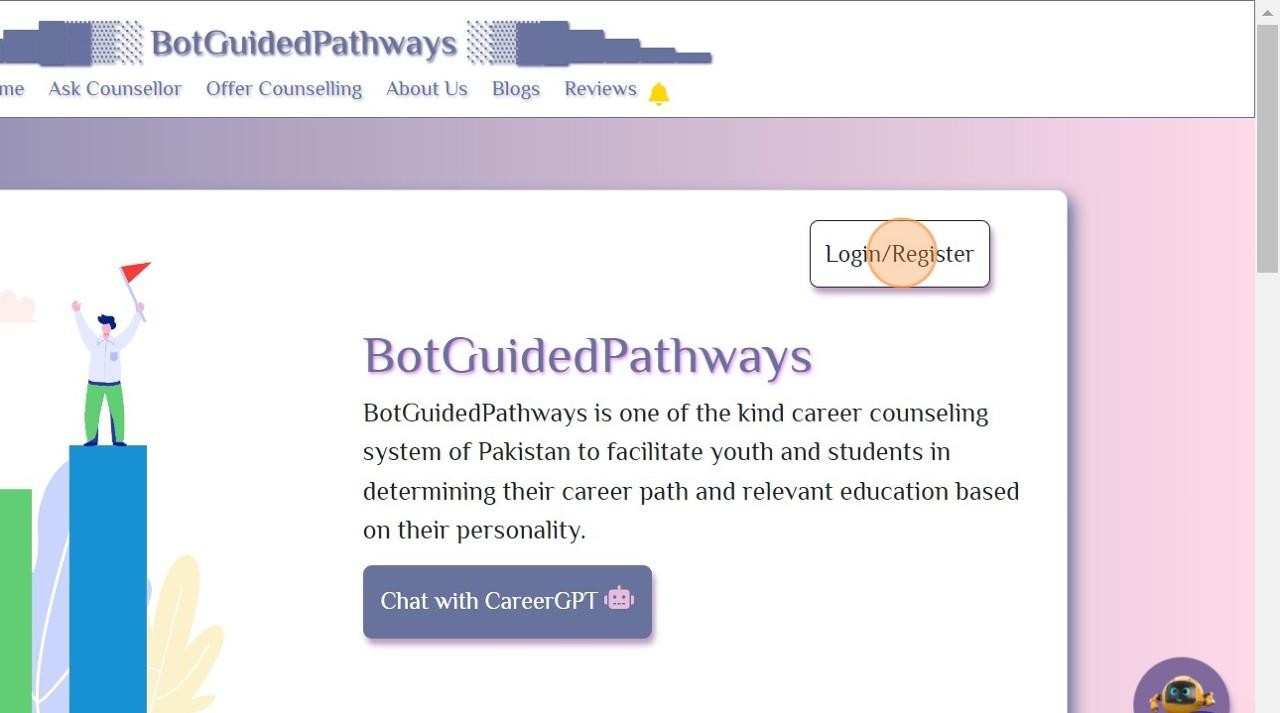


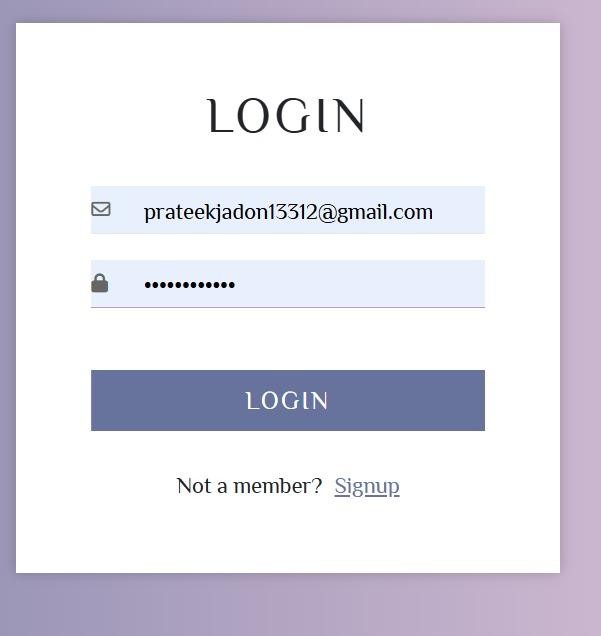




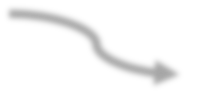
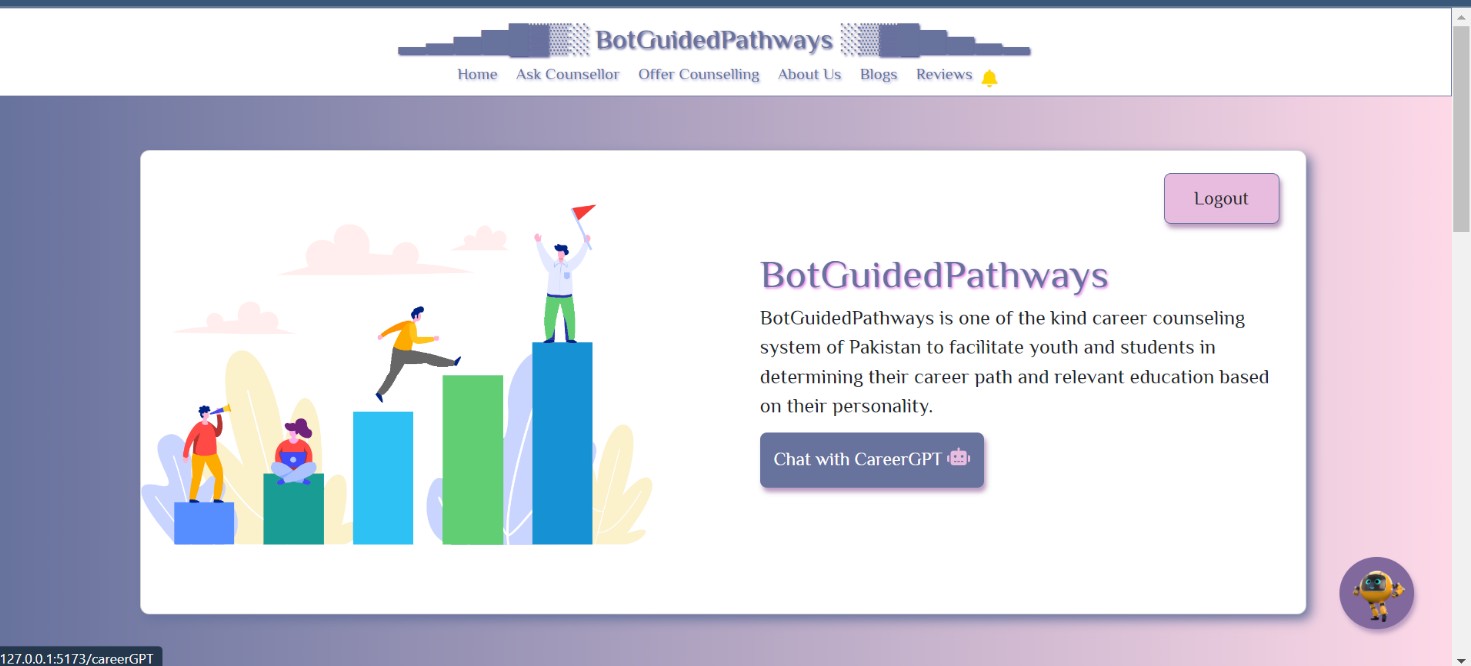


* **Login your account**





* **To Chat with our CareerGPT**🤖



**1**



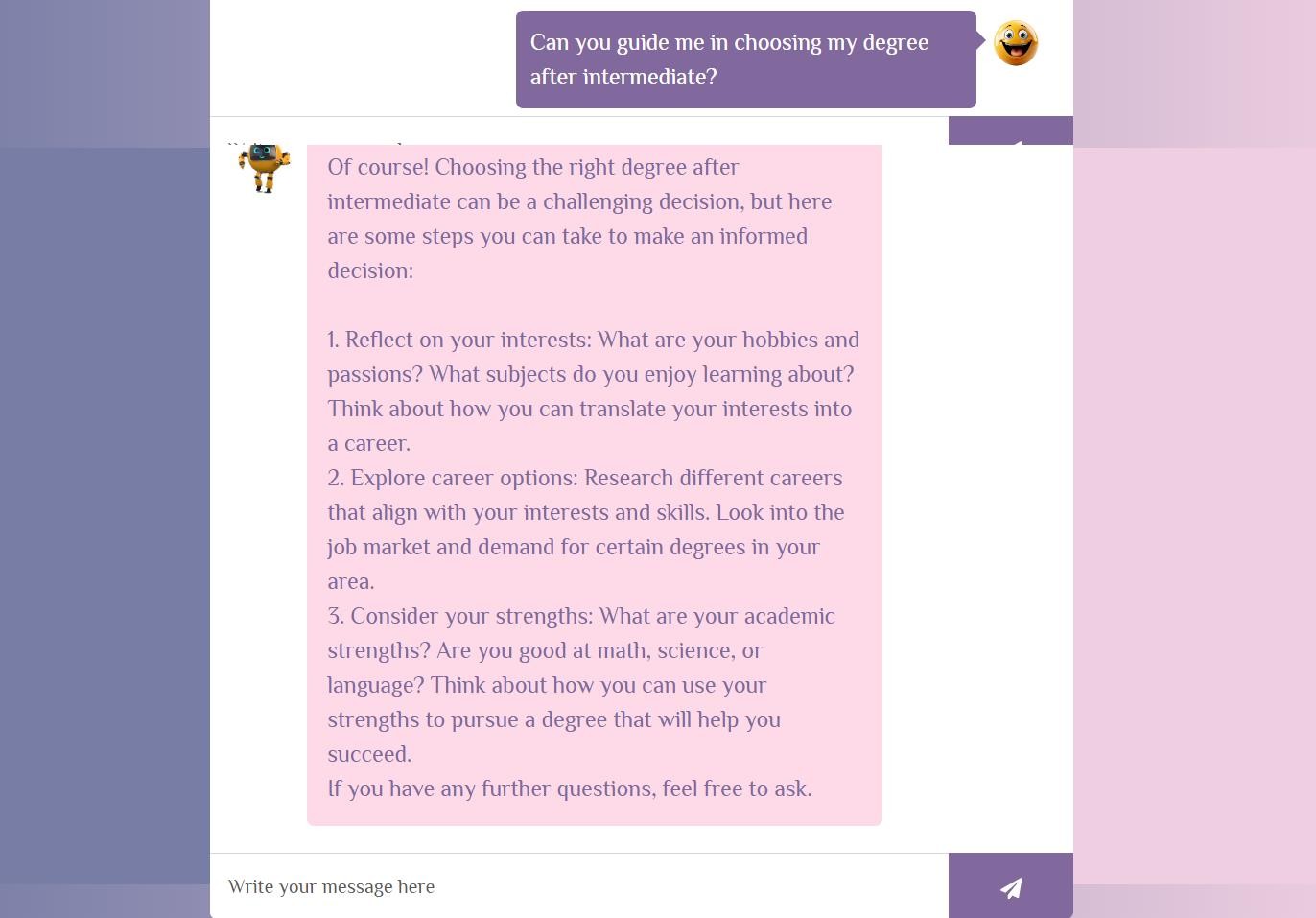
Click

on

CareerGPT

to

chat.



**2**



Give

your

prompt

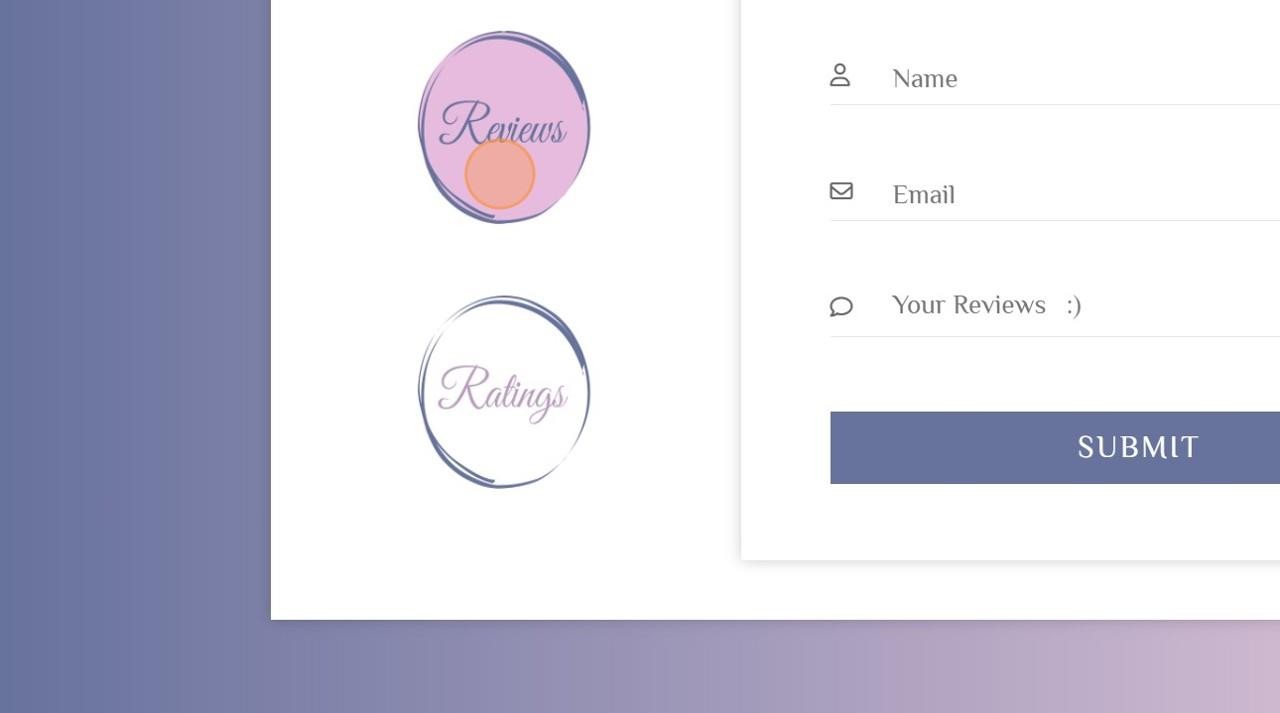
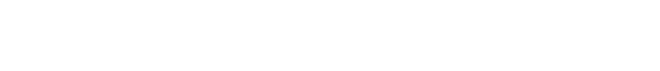
to

get

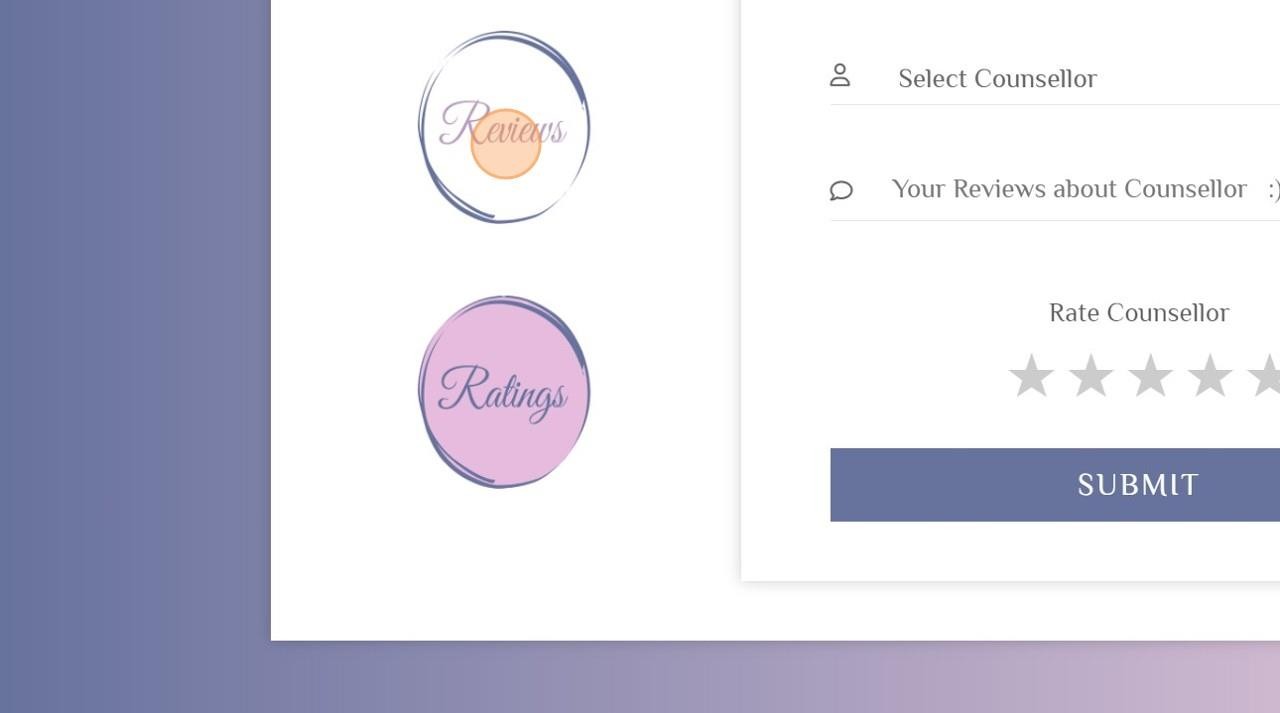
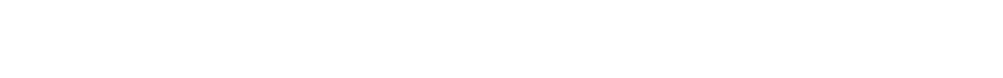
your

response.

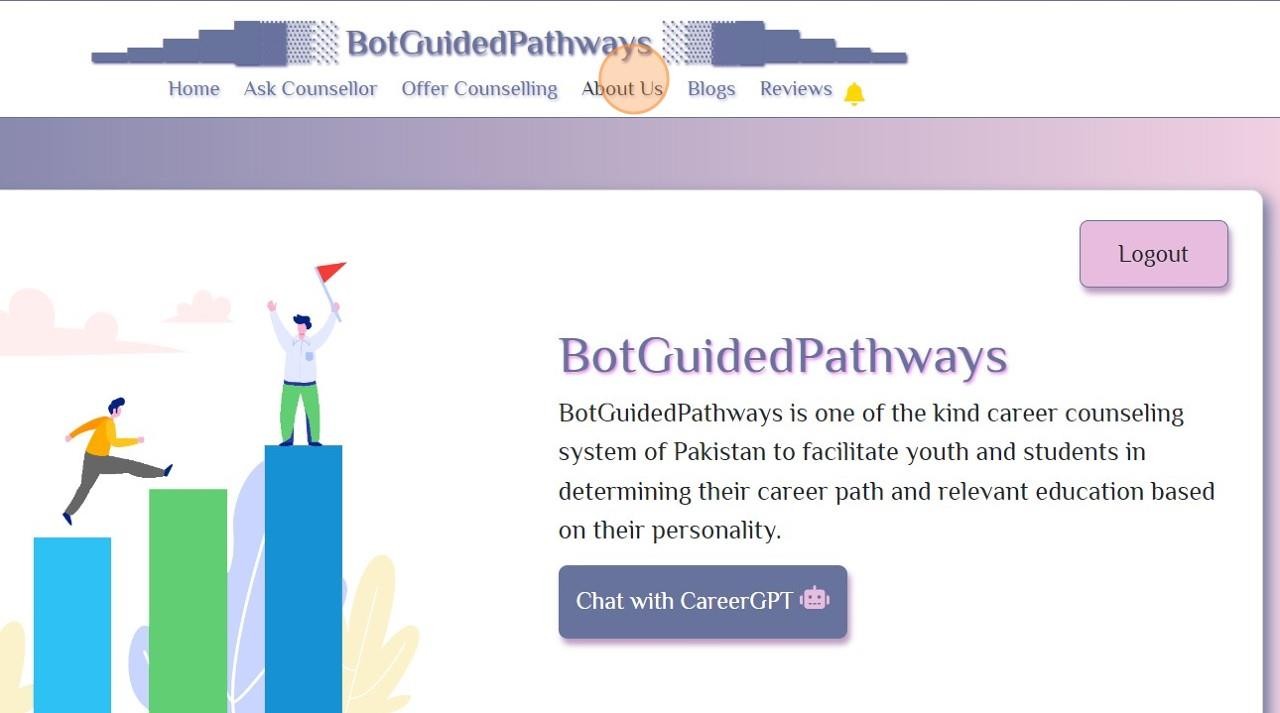
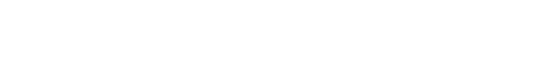
* **To Read Blogs**



* **To read and give Reviews**



* **To know about website**





## Conclusion

The architectural design of our career counselling portal outlines three main scenarios : website visit, utilizing the CareerGPT feature, and engaging with the chat functionality. Each scenario involves interactions between different layers of the system, including the Presentation Layer (implemented using React JS), the Business Layer (handled by Django server), and external services such as Send Bird for chat functionality and Pusher for real-time notifications. Additionally, the design models, including Sequence Diagrams and Data Flow Diagrams, provide a visual representation of various user interactions and system processes. These models help in understanding the flow of data and actions within the system, facilitating efficient development and maintenance. Furthermore, the data design focuses on structuring and organizing data entities such as user profiles, counsellor details, blogs, reviews, and chat interactions. These entities are stored in the SQLite database and managed through Django ORM, ensuring data integrity and efficient retrieval.

Overall, the architectural design, design models, and data design collectively contribute to the functionality, usability, and scalability of our career counselling portal, providing users with a seamless and effective experience.

Chapter 5

# Implementation

## Algorithm

**Table 5: Algorithms**

|  |
| --- |
| **Algorithm 1 User Registration and Authentication** |
| **Input:** Request containing user data |
| **Output:** HTTP Response with success or error message |
| 1: FUNCTION generate\_otp():  2: RETURN random\_number\_between(1000, 9999)  3:  4: FUNCTION sendOTP(request):  5: IF request.method == 'POST':  6: data = parse\_json(request.body)  7: email = data['email']  8: otp = generate\_otp()  9: send\_email('Verification Code', 'Your verification code is: ' + otp, email) 10: RETURN HTTPResponse({'otp': otp}) 11: ELSE:  12: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) 13:  14: FUNCTION registerUser(request):  15: IF request.method == 'POST':  16: data = parse\_json(request.body)  17: signup\_data = data['signupData']  18: name = signup\_data['name']  19: email = signup\_data['email']  20: password = hash\_password(signup\_data['password'])  21: create\_user(name, email, password)  22: RETURN HTTPResponse({'status': 'success'}) 23: ELSE:  24: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) 25:  26: FUNCTION checkEmail(request):  27: IF request.method == 'POST':  28: data = parse\_json(request.body) 29: email = data['email'] 30: IF email\_exists(email):  31: RETURN HTTPResponse({'isExist': True}) 32: ELSE:  33: RETURN HTTPResponse({'isExist': False}) 34: ELSE:  35: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) |

**Complexity:** O(1) for each function, assuming the operations within the functions are O(1)

|  |
| --- |
| **Algorithm 2 Counselling Sessions** |
| **Input:** Request containing user and counsellor data |
| **Output:** HTTP Response with channel URL or error message |
| 1: FUNCTION createSendBirdChannel(request):  2: IF request.method == 'POST':  3: data = parse\_json(request.body)  4: counsellorId = data['counsellorId'] 5: user\_id = get\_session\_user\_id(request) 6: IF user\_id:  7: user = get\_user\_by\_id(user\_id)  8: IF not sendbird\_user\_exists(user\_id):  9: create\_sendbird\_user(user\_id, user.name) 10: IF not sendbird\_user\_exists(counsellorId):  11: create\_sendbird\_user(counsellorId, data['counsellorNickName'])  12: channel\_url = create\_sendbird\_channel(user\_id, counsellorId) 13: RETURN HTTPResponse({'channel\_url': channel\_url}) 14: ELSE:  15: RETURN HTTPResponse({'error': 'User not logged in'}, status=401) 16: ELSE:  17: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) |
| **Complexity:** O(1) for each function, assuming the operations within the functions are O(1) |

|  |
| --- |
| **Algorithm 3 CareerGPT History** |
| **Input:** Request containing user history data |
| **Output:** HTTP Response with success or error message |
| 1: FUNCTION saveHistory(request):  2: IF request.method == 'POST':  3: data = parse\_json(request.body)  4: user\_id = get\_session\_user\_id(request)  5: delete\_user\_history(user\_id) // O(n) where n is the number of history items  6: FOR item IN data: // O(n) where n is the number of items in the data  7: save\_history\_item(user\_id, item['id'], item['message'], item['type']) // O(1) 8: RETURN HTTPResponse({'status': 'success'}) 9: ELSE:  10: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) 11:  12: FUNCTION getHistory(request):  13: IF request.method == 'GET':  14: user\_id = get\_session\_user\_id(request)  15: history = get\_user\_history(user\_id) // O(1) assuming fetching history is constant time  16: RETURN HTTPResponse({'history': history}) 17: ELSE:  18: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) |
| **Complexity:** O(n) for saveHistory (where n is the number of history items), O(1) for getHistory |

|  |
| --- |
| **Algorithm 4 Content Management** |
| **Input:** Request for blog or review data |
| **Output:** HTTP Response with the requested data or success/error message |
| 1: FUNCTION fetchBlogsData(request):  2: IF request.method == 'GET':  3: blogsData = get\_all\_blogs() // O(m) where m is the number of blogs  4: truncate\_blog\_descriptions(blogsData) // O(m) 5: RETURN HTTPResponse({'blogsData': blogsData}) 6: ELSE:  7: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405)  8:  9: FUNCTION blogDetails(request):  10: IF request.method == 'POST':  11: data = parse\_json(request.body)  12: blog\_id = data['id']  13: blogDetails = get\_blog\_details(blog\_id) // O(1)  14: RETURN HTTPResponse({'blogDetails': blogDetails}) 15: ELSE:  16: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) 17:  18: FUNCTION getUnapprovedReviews(request):  19: IF request.method == 'GET':  20: reviewsData = get\_unapproved\_reviews() // O(1)  21: RETURN HTTPResponse({'unapprovedReviews': reviewsData}) 22: ELSE:  23: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) 24:  25: FUNCTION deleteReview(request):  26: IF request.method == 'POST':  27: data = parse\_json(request.body)  28: review\_id = data['selectedRow']  29: delete\_review(review\_id) // O(1)  30: RETURN HTTPResponse({'status': 'Review deleted successfully'}) 31: ELSE:  32: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) 33:  34: FUNCTION approveReview(request):  35: IF request.method == 'PUT':  36: data = parse\_json(request.body)  37: review\_id = data['selectedRow']  38: approve\_review(review\_id) // O(1)  39: RETURN HTTPResponse({'status': 'Review approved successfully'}) 40: ELSE:  41: RETURN HTTPResponse({'error': 'Method not allowed'}, status=405) |
| **Complexity:** O(m) for fetchBlogsData where m is the number of blogs, O(1) for other functions assuming constant-time operations within the functions. |

## External APIs/SDKs

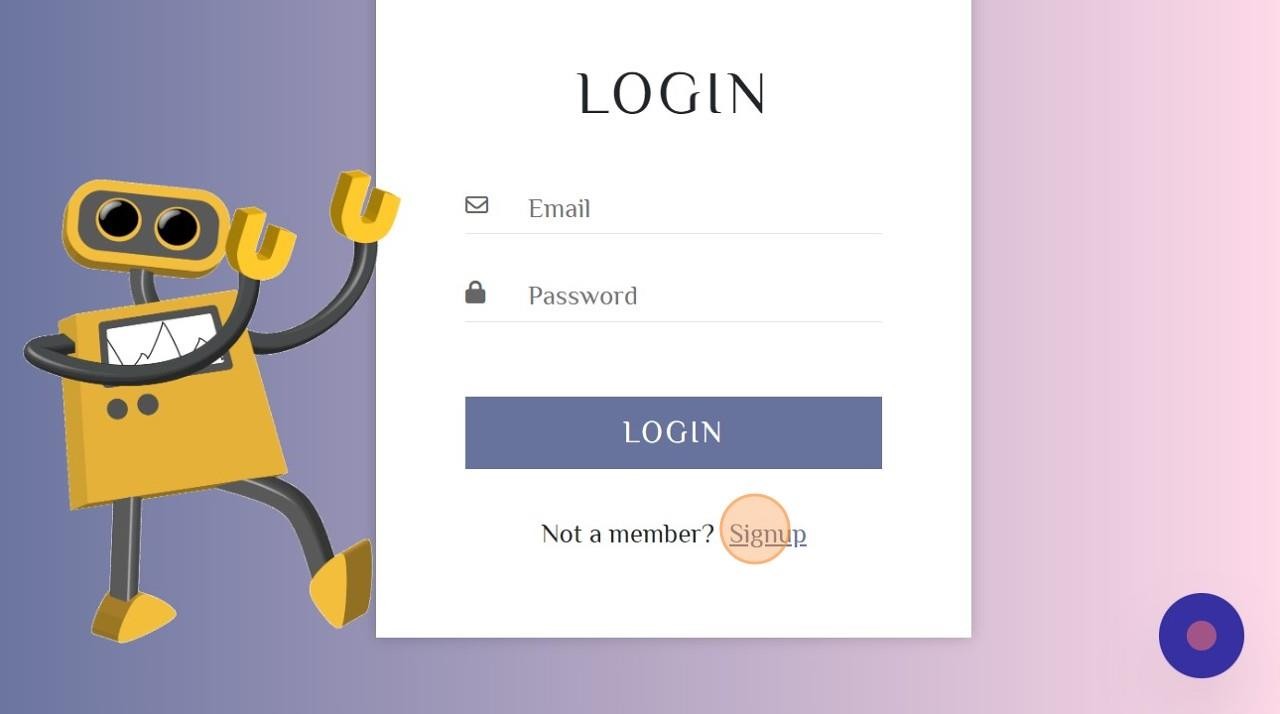
**Table 6: Details of APIs used in the project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of API and version** | **Description of API** | **Purpose of usage** | **List down the API**  **endpoint/function/class in which it is used** |
| Sendbird v3 | A messaging API that provides chat functionality for applications. | To create users, channels, and manage messaging features within the application. | createUser, createChannel, getUser, sendBirdWebHook |
| Pusher | A real-time API that enables instant notifications and updates within applications. | To send real-time notifications to users about new messages or updates. | Used within the  sendBirdWebHook function and frontend code for subscribing to channels and binding events. |

## User Interface

### Login Screen

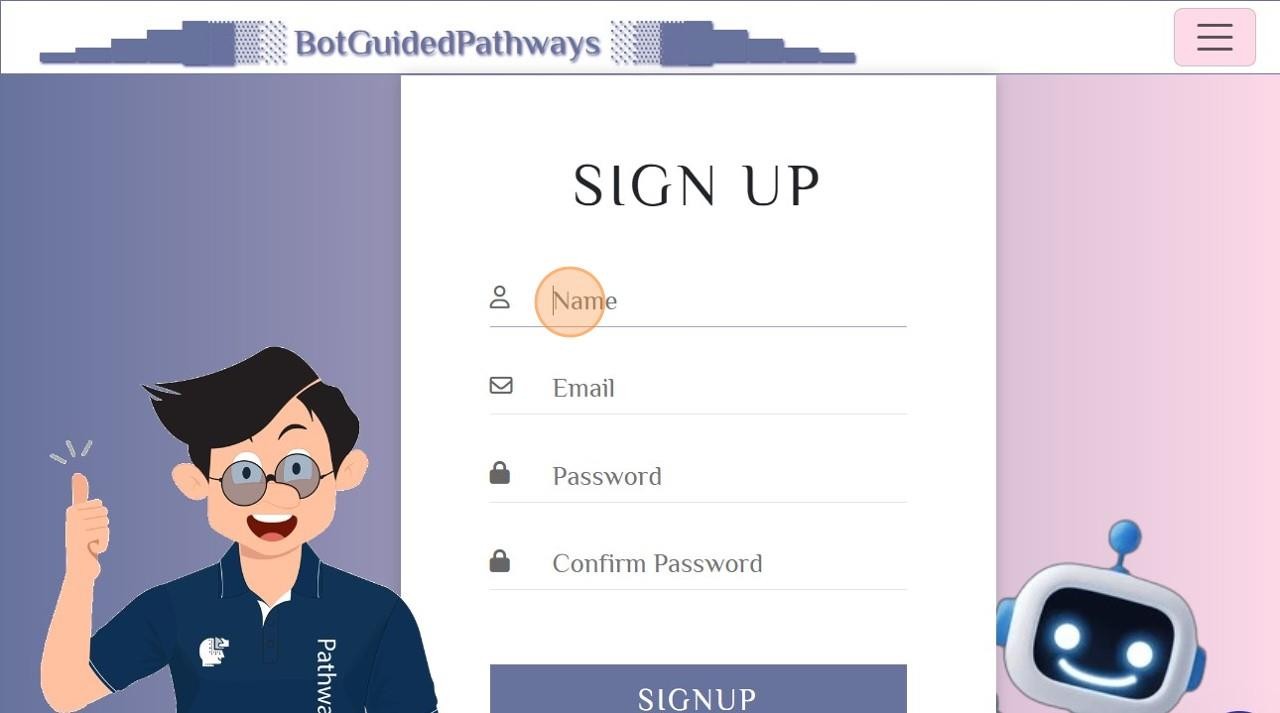
Login screen presents email and password field and the user enters the login details to log into their account.



**Figure 7: Login Screen**

### SignUp Screen

SignUp screen presents name, email and password field and the user enters his details to register in this website.



**Figure 8: SignUp Screen**

### Home Screen

Home Screen presents the navigation bar and popular blogs and latest reviews for the user to have a quick view of blogs and other informative data.



**Figure 9: Home Screen**

## Conclusion

This chapter outlines a comprehensive system for user registration and authentication, counselling session management, history tracking, and content management. The algorithms presented ensure efficient and secure operations with constant time complexity for most functions, ensuring scalability and performance. The use of external APIs like SendBird and Pusher enhances the application's real-time communication and notification capabilities, providing a seamless user experience. Detailed steps for implementing these functionalities are provided, ensuring clarity and ease of development. The user interface design, featuring a login screen, sign-up screen, home screen, and counsellor list screen, is user-friendly and intuitive, facilitating easy navigation and access to essential features.

Overall, this project provides a robust framework for a feature-rich application, balancing functionality, security, and user experience.

Chapter 6

# Testing and Evaluation

## Unit Testing

* **Authentication Module:**

**Unit Testing 1: User Registration**

**Testing Objective:** To ensure the registration form is working correctly with valid and invalid inputs, including OTP validation and password requirements.

**Table 7: Unit Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Register with valid inputs | **Name:** "John Doe",  **Email:** "john@example.com",  **Password:** "Password123!" | Registration successful, user  receives a verification email with OTP |
| 2 | Register with invalid email | **Name:** "John Doe",  **Email:** "john@.com",  **Password:** "Password123!" | **Error message:** "Invalid or non-existent email address" |
| 3 | Register with weak password (short) | **Name:** "John Doe",  **Email:** "john@example.com",  **Password:** "pass" | **Error message:** "Password should include uppercase characters, special characters, and be at least 8 characters  long" |
| 4 | Register with weak password (no uppercase) | **Name:** "John Doe",  **Email:** "john@example.com",  **Password:** "password123!" | **Error message:** "Password should include uppercase characters, special characters, and be at least 8 characters  long" |
| 5 | Register with weak password (no special  characters) | **Name:** "John Doe",  **Email:** "john@example.com",  **Password:** "Password123" | **Error message:** "Password should include uppercase characters, special characters, and be at least 8 characters  long" |
| 6 | Enter correct OTP | **OTP:** "123456" | OTP validated successfully; user registered successfully |
| 7 | Enter incorrect OTP | **OTP:** "654321" | **Error message:** "Invalid OTP, please try again" |

**Unit Testing 2: User Login**

**Testing Objective:** To ensure the login form works correctly with valid and invalid credentials.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Login with valid credentials | **Email:** "john@example.com", **Password:** "Password123!" | Login successful, user redirected to home page |
| 2 | Login with invalid email | **Email:** "john@.com",  **Password:** "Password123!" | **Error message:** "Invalid email or password" |
| 3 | Login with incorrect password | **Email:** "john@example.com", **Password:** "WrongPass123" | **Error message:** "Invalid email or password" |

* **Chatbot Module**

**Unit Testing 3: Chatbot Interaction**

**Testing Objective:** To ensure the chatbot correctly interacts with users and provides accurate career guidance.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Interaction with valid inputs | **Educational background:** "Science", **Interests:** "Programming" | Career advice provided for suitable bachelor's degree programs |
| 2 | Incomplete inputs | **Educational background:** "Science",  **Interests:** "" | Prompt for more information |
| 3 | User not logged in | **User status:** "Not logged in" | User redirected to login page |
| 4 | User logged in | **User status:** "Logged in", **Educational background:** "Arts", **Interests:**  "Design" | Career advice provided for suitable degree programs in Design |

* **Chat Module**

**Unit Testing 4: Chat Interaction**

**Testing Objective:** To ensure users must be logged in before chatting with counsellors and the chat functionality works correctly using SendBird.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | User not logged in | **User status:** "Not logged in" | User redirected to login page |
| 2 | User logged in and starts chat | **User status:** "Logged in",  **Initiate chat with:** "Counsellor1" | Chat session initiated with Counsellor |
| 3 | Send message to counsellor | **Chat session:** Active,  **Message:** "Hello, I need guidance on career options." | Message sent successfully and counsellor receives the message |
| 4 | Receive message from counsellor | **Chat session:** Active,  **Incoming message:** "Sure, how can I help you today?" | Message received successfully by the user |

* **Notification Module**

**Unit Testing 5: User Notification**

**Testing Objective:** To ensure users receive notifications correctly using Pusher.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Notification on new message | **User ID:** 1234,  **Message:** "New message from counsellor" | User receives a notification about the new message through Pusher |
| 2 | Notification failure | **User ID:** 9999,  **Message:** "New message from counsellor" | **Error message:** "User not found" |
| 3 | Notification delivery failure | **User ID:** 1234,  **Message:** "New message from counsellor", Pusher service down | **Error message:** "Notification delivery failed, please try again later" |

* **Offer Counselling Module**

**Objective:** To ensure the counsellor registration form works correctly, including personal details, academic details, working experience details, and OTP validation.

**Unit Testing 6: Counsellor Registration**

**Testing Objective:** To ensure the registration form is working correctly with valid and invalid inputs, including OTP validation.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Register with all valid inputs | **Name:** "Jane Doe",  **Email:** "jane@example.com", **Personal Details:** Complete,  **Academic Details:** Complete,  **Working Experience:** Complete | Registration successful, user  receives a verification email with OTP |
| 2 | Register with missing personal details | **Name:** "",  **Email:** "jane@example.com",  **Academic Details:** Complete,  **Working Experience:** Complete | **Error message:** "Please fill all fields" |
| 3 | Register with missing academic details | **Name:** "Jane Doe",  **Email:** "jane@example.com",  **Personal Details:** Complete,  **Working Experience:** Complete | **Error message:** "Please fill all fields" |
| 4 | Register with missing working experience details | **Name:** "Jane Doe",  **Email:** "jane@example.com",  **Personal Details:** Complete,  **Academic Details:** Complete | **Error message:** "Please fill all fields" |
| 5 | Enter correct OTP | **OTP:** "123456" | OTP validated successfully; details sent to admin for verification |
| 6 | Enter incorrect OTP | **OTP:** "654321" | **Error message:** "Invalid OTP, please try again" |

* **Reviews & Ratings Module**

**Unit Testing 7: Submit Review & Rating**

**Testing Objective:** To ensure users can submit reviews and ratings after logging in.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Submit valid review | **Login:** valid credentials | Review submitted successfully, displayed after admin approval |
| 2 | Submit empty review about counsellor | **Login:** valid credentials | **Error message:** "Review cannot be empty" |
| 3 | Submit review with empty review about website | **Login:** valid credentials | **Error message**: "Review cannot be empty" |
| 4 | Submit review without logging in | **Login:** not logged in | **Error message:** "Login required to submit a review" and navigate to login page. |

* **Counsellors’ Module**

**Unit Testing 8: Add Blog**

**Testing Objective:** To ensure counsellors can add blogs correctly and must be logged in to use this functionality. All fields must be filled, otherwise an error is displayed.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Counsellor not logged in | **User status:** "Not logged in" | User redirected to login page |
| 2 | Add blog with valid inputs | **User status:** "Logged in",  **Title:** "Career in IT",  **Content Body:** "Content of blog" | Blog submitted for approval |
| 3 | Add blog with missing  title | **User status:** "Logged in",  **Title:** "",  **Content:** "Content of blog" | **Error message:** "Title is required" |
| 4 | Add blog with missing content | **User status:** "Logged in",  **Title:** "Career in IT", **Content:** "" | **Error message:** "Content is required" |
| 5 | Add blog with all fields missing | **User status:** "Logged in", **Title:** "", **Content:** "" | **Error message:** "Title and Content are required" |

* **Admin’s Module:**

**Approve Reviews**

**Unit Testing 9: Approve Reviews**

**Testing Objective:** To ensure the admin can approve reviews and the reviews are displayed on the home screen after approval.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Approve valid review | **Review ID:** 1234, **Review:** "xyz" | Review status updated to "Approved", review displayed on home screen |
| 2 | Disapprove invalid review | **Review ID:** 9999, **Review:** "xyzirrlevant review ", | Review status updated to  "Disapproved", review delete from database. |

**Approve Blogs**

**Unit Testing 10: Approve Blogs**

**Testing Objective:** To ensure the admin can approve blogs and the blogs are displayed on the home screen and the counsellor's dashboard after approval.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Approve valid blog | **Blog ID:** 5678,  **Blog Content:** "xyz" | Blog status updated to "Approved", blog displayed on home screen and counsellor dashboard |
| 2 | Disapprove invalid blog | **Blog ID:** 8888,  **Blog Content:** "xyz" | Blog status updated to "Disapproved", blog data delete from database. |

**Approve Counsellor**

**Unit Testing 11: Approve Counsellor**

**Testing Objective:** To ensure the admin can approve counsellor registrations, sending appropriate emails and allowing login only after approval.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** |
| 1 | Approve valid counsellor registration | **Counsellor ID:** 1234 | Counsellor status updated to  "Approved", notification email sent with verification and welcome message, can login |
| 2 | Disapprove counsellor registration | **Counsellor ID:** 1234 | Counsellor status updated to  "Disapproved", rejection email sent with reason. |

## Functional Testing

**Functional Testing 1:** Login with different roles ((User, Counsellor, Admin))

**Objective:** To ensure that the correct page with the correct navigation bar is loaded.

**Table 8: Functional Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Login as a 'User' | **Username:** U001, **Password:** 1234 | Main page for the User is loaded with the User navigation bar | Logged in, redirected to User main page |
| 2 | Login as a 'Counsellor' | **Username:** C002, **Password:** 1234 | Main page for the Counsellor is loaded with the Counsellor navigation bar | Logged in, redirected to Counsellor main page |
| 3 | Login as an 'Admin' | **Username:** A003, **Password:** 1234 | Main page for the admin is loaded with the admin navigation bar | Logged in, redirected to Admin main page |
| 4 | Login with invalid credentials | **Username:** invalid, **Password:** invalid | **Error message:** "Invalid username or password" | Error message displayed |

**Functional Testing 2: User Registration**

**Objective:** To ensure the registration form is working correctly with valid and invalid inputs, including OTP validation and password requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Register with valid inputs | **Name:** "John Doe",  **Email:**"john@example.com",  **Password:** "Password123!" | Registration successful, user receives a verification email with OTP | Registration successful, user received a verification  email with  OTP |
| 2 | Register with invalid email | **Name:** "John Doe",  **Email:** "john@.com",  **Password:** "Password123!" | **Error message:** "Invalid or nonexistent email address" | **Error message:** "Invalid or non-existent email address" |
| 3 | Register with weak password (short) | **Name:** "John Doe", **Email:**  "john@example.com",  **Password:** "pass" | **Error message:** "Password should include uppercase characters, special characters, and | **Error message:** "Password should include uppercase characters, special characters, and |
|  |  |  | be at least 8 characters long" | be at least 8 characters long" |
| 4 | Register with weak password (no uppercase) | **Name:** "John Doe", **Email:**  "john@example.com",  **Password:** "password123!" | **Error message:** "Password should include uppercase characters, special characters, and be at least 8 characters long" | **Error message:** "Password should include uppercase characters, special characters, and be at least 8 characters long" |
| 5 | Register with weak password (no special  characters) | **Name:** "John Doe", **Email:**  "john@example.com",  **Password:** "Password123" | **Error message:** "Password should include uppercase characters, special characters, and be at least 8 characters long" | **Error message:** "Password should include uppercase characters, special characters, and be at least 8 characters long" |
| 6 | Enter correct OTP | **OTP:** "123456" | OTP validated successfully; user registered successfully | OTP validated successfully; user registered successfully |
| 7 | Enter incorrect OTP | **OTP:** "654321" | **Error message:** "Invalid OTP, please try again" | **Error message:** "Invalid OTP, please try again" |

**Functional Testing 3: User Login**

**Objective:** To ensure the login form works correctly with valid and invalid credentials.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Login with valid credentials | **Email:** "john@example.com", **Password:** "Password123!" | Login successful, user  redirected to home page | Login successful, user redirected to home page |
| 2 | Login with invalid email | **Email:** "john@.com",  **Password:** "Password123!" | **Error message:** "Invalid email or password" | **Error message:** "Invalid email or password" |
| 3 | Login with incorrect password | **Email:** "john@example.com", **Password:** "WrongPass123" | **Error message:** "Invalid email or password" | **Error message:** "Invalid email or password" |

**Functional Testing 4: Chatbot Interaction**

**Objective:** To ensure the chatbot correctly interacts with users and provides accurate career guidance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Interaction with valid inputs | **Educational background:** "Science", **Interests:**  "Programming" | Career advice provided for suitable bachelor's degree programs | Career advice provided for suitable bachelor's degree programs |
| 2 | Incomplete inputs | **Educational background:**  "Science",  **Interests:** "" | Prompt for more information | Prompt for more information |
| 3 | User not logged in | **User status:** "Not logged in" | User redirected to login page | User redirected to login page |
| 4 | User logged in | **User status:** "Logged in",  **Educational background:** "Arts", **Interests:** "Design" | Career advice provided for suitable degree programs in Design | Career advice provided for suitable degree programs in Design |

**Functional Testing 5: Chat Interaction**

**Objective:** To ensure users must be logged in before chatting with counsellors and the chat functionality works correctly using Sendbird.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | User not logged in | **User status:** "Not logged in" | User redirected to login page | User redirected to login page |
| 2 | User logged in and starts chat | **User status:**  "Logged in",  **Initiate chat with:**  "Counsellor1" | Chat session initiated with Counsellor | Chat session initiated with Counsellor |
| 3 | Send message to counsellor | **Chat session:**  **Active, Message:** "Hello, I need guidance on career  options." | Message sent successfully and counsellor  receives the message | Message sent successfully and counsellor receives the message |
| 4 | Receive message from counsellor | **Chat session:**  Active,  **Incoming message:** "Sure, how can I help you today?" | Message received successfully by the user | Message received successfully by the user |

**Functional Testing 6: Add Blog**

**Objective:** To ensure counsellors can add blogs correctly and must be logged in to use this functionality. All fields must be filled, otherwise an error is displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Counsellor not logged in | **User status:** "Not logged in" | User redirected to login page | User redirected to login page |
| 2 | Add blog with valid inputs | **User status:**  "Logged in", | Blog submitted for approval | Blog submitted for approval |
|  |  | **Title:** "Career in IT", **Content Body:**  "Content of blog" |  |  |
| 3 | Add blog with missing title | **User status:**  "Logged in",  **Title:** "",  **Content:** "Content of blog" | **Error message:** "Title is required" | **Error message:** "Title is required" |
| 4 | Add blog with missing content | **User status:**  "Logged in",  **Title:** "Career in IT",  **Content:** "" | **Error message:** "Content is required" | **Error message:** "Content is required" |
| 5 | Add blog with all fields missing | **User status:**  "Logged in",  **Title:** "", **Content:** "" | **Error message:** "Title and Content are required" | **Error message:** "Title and Content are required" |

**Functional Testing 7: User Notification**

**Objective:** To ensure users receive notifications correctly using Pusher.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Notification on new message | **User ID:** 1234, **Message:** "New message from counsellor" | User receives a notification about the new message through Pusher | User receives a notification about the new message through Pusher |
| 2 | Notification failure | **User ID:** 9999, **Message:** "New message from counsellor" | **Error message:**  "User not found" | **Error message:** "User not found" |
| 3 | Notification delivery failure | **User ID:** 1234, **Message:** "New message from counsellor", Pusher service down | **Error message:** "Notification delivery failed, please try again later" | **Error message:** "Notification delivery failed, please try again later" |

**Functional Testing 8: Submit Review & Rating**

**Objective:** To ensure users can submit reviews and ratings after logging in.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Submit valid review | **Login:** valid credentials | Review submitted successfully, | Review submitted successfully, |
|  |  |  | displayed after admin approval | displayed after admin approval |
| 2 | Submit empty review about counsellor | **Login:** valid credentials | **Error message:** "Review cannot be empty" | **Error message:** "Review cannot be empty" |
| 3 | Submit review with empty review about website | **Login:** valid credentials | **Error message:** "Review cannot be empty" | **Error message:** "Review cannot be empty" |
| 4 | Submit review without logging in | **Login:** not logged in | **Error message:** "Login required to submit a review" and navigate to login page | **Error message:** "Login required to submit a review" and navigate to login page |

**Functional Testing 9: Approve Reviews**

**Objective:** To ensure the admin can approve reviews and the reviews are displayed on the home screen after approval.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Approve valid review | **Review ID:** 1234, **Review:** "xyz" | Review status updated to "Approved", review displayed on home screen | Review status updated to  "Approved", review displayed on home screen |
| 2 | Disapprove invalid review | **Review ID:** 9999, **Review:** "xyzirrlevant review" | Review status updated to "Disapproved", review deleted from database | Review status updated to "Disapproved", review deleted from database |

**Functional Testing 10: Approve Blogs**

**Objective:** To ensure the admin can approve blogs and the blogs are displayed on the home screen and the counsellor's dashboard after approval.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Approve valid blog | **Blog ID:** 5678, Blog **Content:** "xyz" | Blog status updated to "Approved", | Blog status updated to "Approved", blog displayed on |
|  |  |  | blog displayed on home screen and counsellor dashboard | home screen and counsellor dashboard |
| 2 | Disapprove invalid blog | **Blog ID:** 8888, Blog **Content:** "xyz" | Blog status updated to "Disapproved", blog data deleted from database | Blog status updated to "Disapproved", blog data deleted from database |

**Functional Testing 11: Approve Counsellors**

**Objective:** To ensure the admin can approve counsellor registrations, sending appropriate emails and allowing login only after approval.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Approve valid counsellor registration | **Counsellor ID:** 1234 | Counsellor status updated to "Approved", notification email sent with  verification and welcome message, can login | Counsellor status updated to "Approved", notification email sent with verification and welcome message, can login |
| 2 | Disapprove counsellor registration | **Counsellor ID:** 1234 | Counsellor status updated to "Disapproved", rejection email sent with reason | Counsellor status updated to "Disapproved", rejection email sent with reason |

**Functional Testing 12: Offer Counselling Module**

**Testing Objective:** To ensure the registration form is working correctly with valid and invalid inputs, including OTP validation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Register with all valid inputs | **Name:** "Jane Doe", **Email:**  "jane@example.com", | Registration successful, user receives a | Registration successful, user received a |
|  |  | **Personal Details:**  Complete, **Academic**  **Details:** Complete, **Working Experience:**  Complete | verification email with  OTP | verification email with  OTP |
| 2 | Register with missing personal details | **Name:** "", **Email:** "jane@example.com", **Academic Details:**  Complete,  **Working Experience:**  Complete | **Error message:** "Please  fill all fields" | **Error message:** "Please  fill all fields" |
| 3 | Register with  missing academic details | **Name:** "Jane Doe", **Email:**  "jane@example.com", **Personal Details:**  Complete,  **Working Experience:**  Complete | **Error message:** "Please  fill all fields" | **Error message:** "Please  fill all fields" |
| 4 | Register with missing working experience details | **Name:** "Jane Doe", **Email:**  "jane@example.com", **Personal Details:**  Complete,  **Academic Details:**  Complete | **Error message:** "Please  fill all fields" | **Error message:** "Please  fill all fields" |
| 5 | Enter correct OTP | **OTP:** "123456" | OTP validated  successfully; details sent to admin for verification | OTP validated  successfully; details sent to admin for verification |
| 6 | Enter incorrect OTP | **OTP:** "654321" | **Error message:** "Invalid OTP, please try again" | **Error message:** "Invalid OTP, please try again" |

## Integration Testing

**Integration Testing 1: User Login and Navigation**

**Objective:** To ensure that users with different roles can log in and navigate to their respective dashboards and functionalities without errors.

**Table 9: Integration Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Login and navigate as 'User' | **Username:** U001, **Password:** 1234 | User is logged in and navigated to | User is logged in and navigated to the |
|  |  |  | the User main page with correct navigation bar. | User main page with correct navigation bar. |
| 2 | Login and navigate as 'Counsellor' | **Username:** C002, **Password:** 1234 | Counsellor is logged in and navigated to the Counsellor main page with correct navigation bar. | Counsellor is logged in and navigated to the Counsellor main page with correct navigation bar. |
| 3 | Login and navigate as 'Admin' | **Username:** A003, **Password:** 1234 | Admin is logged in and navigated to the admin main page with correct navigation bar. | Admin is logged in and navigated to the admin main page with correct navigation bar. |
| 4 | Attempt login with invalid credentials | **Username:** invalid, **Password:** invalid | **Error message**  "Invalid username or password" is displayed. | **Error message** "Invalid username or password" is displayed. |

**Integration Testing 2: User Registration and Login**

**Objective:** To ensure a seamless flow from user registration to successful login.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Register and login with valid inputs | **Name:** "John Doe", **Email:**  "john@example.com", **Password:**  "Password123!" | Registration successful, verification email with OTP received. User can log in with the same credentials. | Registration successful, verification email with OTP received. User can log in with the same credentials. |
| 2 | Register and login with invalid email | **Name:** "John Doe", **Email:** "john@.com", **Password:**  "Password123!" | **Error message:** "Invalid or non-existent email address". Login attempt with invalid email fails. | **Error message:** "Invalid or non-existent email address". Login attempt with invalid email fails. |
| 3 | Register and login with weak password | **Name:** "John Doe", **Email:**  "john@example.com",  **Password:** "pass" | **Error message:**  "Password should include uppercase characters, special characters, and be at least 8 characters long". | **Error message:**  "Password should include uppercase characters, special characters, and be at least 8 characters long". |
| 4 | Register, verify OTP, and login | **Name:** "John Doe", **Email:**  "john@example.com", **Password:**  "Password123!",  **OTP:** "123456" | OTP validated  successfully; registration completed. User can log in with registered credentials. | OTP validated  successfully; registration completed. User can log in with registered credentials. |

**Integration Testing 3: Chatbot and Chat Interaction**

**Objective:** To ensure users can interact with the chatbot for career guidance and initiate chat sessions with counsellors.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Interact with chatbot as a logged-in user | **User status:** "Logged in", **Educational background:** "Science", **Interests:**  "Programming" | Chatbot provides career advice for suitable bachelor's degree programs. | Chatbot provides career advice for suitable bachelor's degree programs. |
| 2 | Interact with chatbot without logging in | **User status:** "Not logged in", **Educational** | User is prompted to log in before accessing chatbot services. | User is prompted to log in before accessing chatbot services. |
|  |  | **background:** "Science", **Interests:**  "Programming" |  |  |
| 3 | Initiate chat with a counsellor as a logged-in user | **User status:** "Logged  in",  **Initiate chat with:**  "Counsellor1" | Chat session is successfully initiated with the counsellor. | Chat session is successfully initiated with the counsellor. |
| 4 | Send and receive messages in a chat session | **Chat session:** Active, **Message:** "Hello, I need guidance on career options." | Message is sent successfully and counsellor receives the message. | Message is sent successfully and counsellor receives the message. |

**Integration Testing 4: Blog Addition and Approval**

**Objective:** To ensure that counsellors can add blogs and admins can approve them, displaying them correctly on the home screen and counsellor dashboard.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Add a blog as a logged-in counsellor | **User status:** "Logged  in",  **Title:** "Career in IT", **Content Body:**  "Content of blog" | Blog submitted successfully and awaits admin approval. | Blog submitted successfully and awaits admin approval. |
| 2 | Approve a valid blog | **Admin status:**  "Logged in",  **Blog ID:** 5678, | Blog status updated to "Approved", blog | Blog status updated to "Approved", |
|  |  | **Blog Content:** "xyz" | displayed on the home screen and counsellor dashboard. | blog displayed on the home screen and counsellor dashboard. |
| 3 | Disapprove an invalid blog | **Admin status:**  "Logged in",  **Blog ID:** 8888,  **Blog Content:** "xyz" | Blog status updated to "Disapproved", blog data deleted from the database. | Blog status updated to "Disapproved", blog data deleted from the database. |

**Integration Testing 5: Notification System**

**Objective:** To ensure users receive notifications correctly for various events using Pusher.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Receive notification for a new message | **User ID:** 1234, **Message:** "New message from counsellor" | User receives a notification about the new message through Pusher. | User receives a notification about the new message through Pusher. |
| 2 | Handle notification failure for invalid user | **User ID:** 9999, **Message:** "New message from counsellor" | **Error message:** "User not found" displayed. | **Error message:** "User not found" displayed. |
| 3 | Handle notification delivery failure | **User ID:** 1234, **Message:** "New message from counsellor", Pusher service down | **Error message:** "Notification delivery failed, please try again later" displayed. | **Error message:** "Notification delivery failed, please try again later" displayed. |

**Integration Testing 6: Review Submission and Approval**

**Objective:** To ensure users can submit reviews, and admins can approve or disapprove them for display on the home screen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Submit a valid review | **Login:** valid  credentials,  **Review:** "Excellent guidance!" | Review submitted successfully, displayed after admin approval. | Review submitted successfully, displayed after admin approval. |
| 2 | Approve a valid review | **Admin status:**  "Logged in",  **Review ID:** 1234,  **Review:** "xyz" | Review status updated to  "Approved", review displayed on home screen. | Review status updated to "Approved", review displayed on home screen. |
| 3 | Disapprove an invalid review | **Admin status:**  "Logged in",  **Review ID:** 9999, **Review:** "xyzirrelevant review" | Review status updated to "Disapproved", review deleted from the database. | Review status updated to "Disapproved", review deleted from the database. |

**Integration Testing 7: Counsellor Registration and Approval**

**Objective:** To ensure that counsellor registrations are verified and approved by the admin, allowing them to log in and perform their duties.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** |
| 1 | Register and verify counsellor | **Counsellor ID:** 1234, **Name:** "Jane Doe", **Email:**  "jane@example.com",  **OTP:** "123456" | OTP validated successfully; details sent to admin for verification. | OTP validated successfully; details sent to admin for verification. |
| 2 | Approve a valid counsellor registration | **Admin status:**  "Logged in",  **Counsellor ID:** 1234 | Counsellor status updated to "Approved", notification email sent with verification and welcome message, can login. | Counsellor status updated to "Approved", notification email sent with verification and welcome message, can login. |
| 3 | Disapprove a counsellor registration | **Admin status:**  "Logged in",  **Counsellor ID:** 1234 | Counsellor status updated to "Disapproved", rejection email sent with reason. | Counsellor status updated to "Disapproved", rejection email sent with reason. |

Chapter 7

# Conclusion and Future Work

## Conclusion

To wrap up, our Career Counselling Portal is a helpful tool for students figuring out their career paths. We've put a lot of effort into making it user-friendly and informative, and we're pleased with the results. Our goal was to give students the guidance they need to make smart career choices, and we believe we've achieved that. We're excited about the impact our platform can have on students' lives, and we're committed to continuing to improve and expand its capabilities.

## Future Work

Looking ahead, there are a few areas where we see opportunities for growth and improvement. First, we want to make our career recommendations even better by using more advanced technology. We also want to make sure our platform can handle more users as it becomes more popular. We're thinking about adding feature like ways for students to connect with alumni. Partnering with schools and businesses could also help us reach more students. And of course, we'll keep listening to feedback from users so we can keep making our platform better and better.