

CHALET CONNECT

Booking & promoting platform

Senior Project

BY

**ALI KOMAYHA**

Submitted to the School of Arts & Science of the

Lebanese International University

In part of fulfillment of the requirements for the degree of

**BACHERLOR OF SCIENCE IN**

**COMPUTER SCIENCE**

**Supervised by: Dr. Ali Choumane**

Spring 2023-2024

**Abstract:**

We are creating an online platform that connects chalet owners with potential customers. Offering an easy attractive booking experience and promoting chalet services and their unique details.

The online system presents so many features for both chalet owners and customers. To explain more, the most important features for customers is to explore and view chalets and its details, offering a booking system connected directly to the chalet owners. Moreover, any visitor to the website can view and explore all available chalets, but they need to create an account in order to book a certain chalet. However, for chalet owners, they must create their own owner account and add their property on the system adding all its required details. The chalet owner will benefit from both managing their chalet bookings and in the same time promoting their own business.

Throughout our proposal, we explain every single feature available in our website in details.

To sum up, the purpose of our platform is to provide a special service for both customers and chalet owner that they can access from anywhere and at any time.

Chapter 1

In this chapter, we are going to state the problem that we find, and how our project helps in solving it, in addition to mentioning technology constraints and our objective, then explaining all of them in details.

1.1-Introduction:

In today’s fast-paced world, convenience and accessibility are paramount, especially when it comes to planning vacations and getaways. Traditional methods of booking holiday accommodations, such as visiting multiple websites or making numerous phone calls, can be time-consuming and inefficient. Chalet Connect aims to revolutionize the chalet rental market by providing a dedicated online platform that connects chalet owners with potential customers seamlessly. Our platform not only simplifies the booking process but also enhances the overall experience for both owners and customers. By leveraging modern web technologies, Chalet Connect ensures a user-friendly, efficient, and secure environment for exploring and booking chalets.

1.2-Objective:

The primary objective of Chalet Connect is to streamline the process of finding and booking chalets, thus enhancing user satisfaction and promoting the unique offerings of chalet owners. Our platform aims to:

* Eliminate the challenges associated with traditional chalet bookings, such as limited accessibility and inefficient communication.

• Expand the reach of chalet owners to a wider audience by providing an attractive online presence.

• Offer a robust booking system that manages availability, reservations, and customer interactions effectively.

• Increase booking efficiency and customer satisfaction by providing detailed chalet information, high-quality images, and customer reviews.

1.3-Scope:

Chalet Connect is designed to address the specific needs of the chalet rental market. The system includes:

• **For Customers:** An intuitive interface to search, view, and book chalets based on various criteria such as location, price, and amenities. The platform supports account creation, allowing users to manage their bookings and track their reservations.

• **For Owners:** Tools to create and manage listings, upload high-quality images, set availability, and handle bookings. Owners can also update chalet information and communicate directly with customers through the platform.

• **Booking Management:** A calendar feature to display available and booked dates, ensuring transparency and ease of use for both parties. The system supports full-day bookings, with potential future enhancements for more granular booking options.

• **Additional Services:** Detailed information about chalet services and amenities to help customers make informed decisions. The platform aims to provide 24/7 access, ensuring customers can book chalets at their convenience.

By implementing these features, Chalet Connect aims to offer a comprehensive solution that caters to both chalet owners and customers, ensuring a seamless, efficient, and enjoyable booking experience.

1.4-Technology Constraints:

**Software needed:** Visual studio code, Xampp, Web browser.

**Languages:** php, javascript, html, css, mysql.

**Frameworks & libraries:** bootstarp, jquery.

**Hardware:** laptop.

Chapter 2

This chapter describes both functional and non-functional requirements for our project. In addition to that it shows the UML use cases and scenarios with all related details.

2.1-Functional and Conceptual Study

2.1.1-Functional Requirements:

1- User Registration and Login:

-Sign up: Visitors can explore chalets, but only registered users can make bookings. Users must provide basic information to create an account.

-Log in: Users can log in using their email and password to access their account features.

2-Chalet Listing Management (for Owners):

- Add Chalet: Owners can add new chalets by providing details such as name, location, description, price, and capacity.

- Upload Images: Owners can upload up to 6 images per chalet, with the first image being the main display image.

- Update Chalet Info: Owners can edit the details of their listed chalets.

- Manage Bookings: Owners can view and manage the bookings for their chalets, including the status of each booking.

3- Chalet Browsing and Booking (for Customers):

-Browse Chalets: Customers can browse chalets based on various filters such as location, price, and capacity.

-View Chalet Details: Customers can view detailed information about each chalet, including images, description, services, and availability.

-Booking Chalet: Customers can book chalets for full-day periods. The system will show available and unavailable dates.

-Booking History: Registered customers can view their booking history and manage their reservations.

4- Search Functionality:

-Search Chalets: Customers can search for chalets based on keywords, location, price range, and other criteria.

5- Contact and Feedback:

-Contact Owners: Customers can contact chalet owners directly through the platform to ask questions or request more information.

-Send Feedback: Customers can send feedback about their booking experience and the chalets.

2.1.2-Non-Functional Requirements:

-Security:

- Authentication: All users must authenticate using their email and password to access account-specific features.

-Data Protection: User data, including personal details and payment information, must be securely stored and transmitted using encryption.

-Performance:

-Response Time: User Login: The average response time for user login after entering credentials should be no more than 3 seconds, with a maximum of 10 seconds.

Page Rendering: After clicking any link, the page should render within an average of 1 second, with a maximum of 10 seconds.

-Scalability: The system should support at least 1000 concurrent users without performance degradation.

-Reliability: In case of internet issues, a loading page should inform the user about the connection problem.

-Usability: The interface should be intuitive and user-friendly, ensuring a smooth browsing and booking experience.

The platform should be accessible on various devices, including desktops, tablets, and smartphones.

-Maintainability: The system should be designed for easy maintenance and updates, allowing for quick implementation of new features and bug fixes.

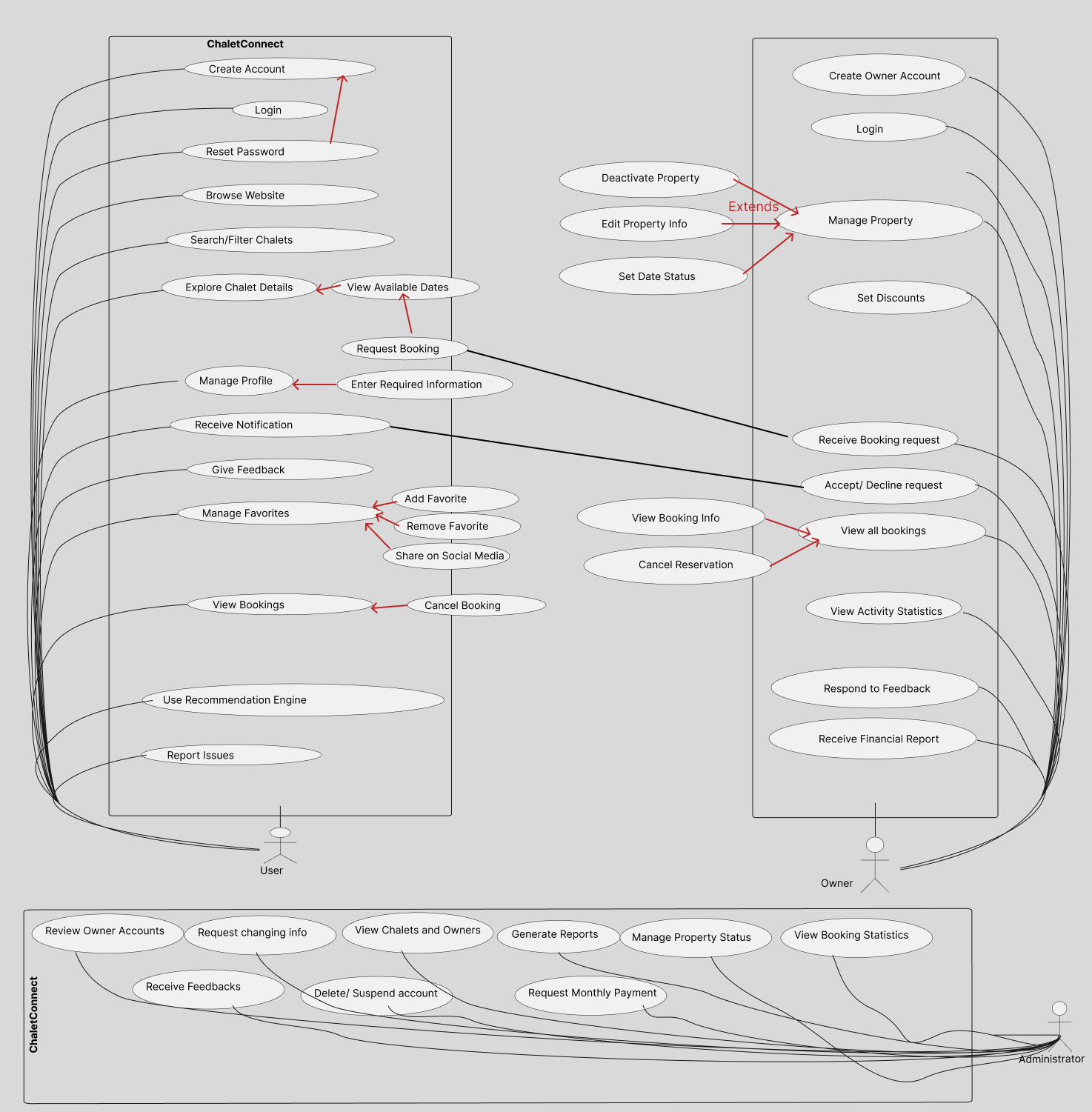
-Availability: The platform should be available 24/7, with minimal downtime for maintenance.

By addressing these functional and non-functional requirements, Chalet Connect aims to provide a robust, user-friendly, and efficient platform for both chalet owners and customers.

2.2-UML Use Case Diagram & Scenario

2.2.1-Use Case Diagram

The use case diagram illustrates interactions between actors and the system. For the Customer actor, interactions include browsing chalets, booking, and managing bookings. Customers can also create accounts, search for chalets, view detailed information, and contact owners. For the Owner actor, interactions involve managing chalets and bookings. Owners can add, edit, or delete listings, view bookings, and respond to feedback. The Admin actor has roles such as reviewing new owner accounts, requesting info changes, updating property status, and monitoring chalet listings and bookings.



2.2.2-Use Case scenarios (customer actor)

In this part we will be explaining all of the mentioned use cases of the customer in details, regarding primary actors, main flow, and the alternative one. In addition to a short description for each use case making all points clear.

Table 1- Login Case

|  |
| --- |
| Use Case Name: User Registration and Login |
| Actor(s): User |
| Brief Description:  This use case involves the process of user registration and subsequent login to the Chalet Connect platform. |
| Basic Flow:  1-The user navigates to the registration page and provides a valid email address.  2-The user creates a password and completes the registration form.  3-The system verifies the email address and creates a new user account.  4-The user logs in using their registered email and password. |
| Alternative Flow:  If the email address provided during registration is invalid, the system prompts the user to provide a valid email address. |
| Pre-conditions:  1-None |
| Post-conditions:  1- Upon successful registration, the user gains access to the platform's features and functionalities. |

Table 2- Search Use Case

|  |
| --- |
| Use Case Name: Searching and Exploring Chalets |
| Actor(s): User |
| Brief Description:  This use case involves users searching for and exploring chalets based on various criteria. |
| Basic Flow:  1-The user enters search criteria such as location, services, and price range.  2-The system displays a list of chalets that match the search criteria.  3-The user selects a chalet from the list to view detailed information.  4-The user explores pictures, location, services, and pricing of the selected chalet. |
| Alternative Flow:  1- If no chalets match the search criteria, the system displays a message indicating no results found. |
| Pre-conditions:  1-None |
| Post-conditions:  1- The user has the option to proceed with booking the selected chalet or continue browsing other options. |

Table 3- Adding to Favorites Case

|  |
| --- |
| Use Case Name: Adding Chalets to Favorites |
| Actor(s): User |
| Brief Description:  This use case involves users adding chalets to their list of favorites for future reference. |
| Basic Flow:  1- The user navigates to a chalet listing that they wish to add to their favorites.  2-The user selects the "Add to Favorites" option.  3-The system adds the chalet to the user's list of favorite chalets. |
| Alternative Flow:  If the chalet is already added to the user's favorites, the system displays a message indicating that the chalet is already in the favorites list. |
| Pre-conditions:  1- The user must be logged in to their account. |
| Post-conditions:  The selected chalet is successfully added to the user's favorites list |

Table 4- Booking use case

|  |
| --- |
| Use Case Name: Booking a Chalet |
| Actor(s): User, Owner |
| Brief Description:  This use case is the process of booking a chalet on a specific date |
| Basic Flow:  1-The user chooses preferred Chalet.  2-The user clicks on one of the available dates.  3-The user chooses request booking option.  4-The user waits for owner’s response. |
| Alternative Flow:  The owner will receive a request.  The owner will check for customer’s info.  The owner will send the approval or denial of the request. |
| Pre-conditions:  1-User must be logged in.  2-User must fill all personal information. |
| Post-conditions:  1-In case the owner accepts the request, the user will receive a notification of the approval and the booking will be added to user’s bookings.  2-In case the owner denies the request, the user will receive a notification that the booking request is rejected. |

Table 5- View Booking History Case

|  |
| --- |
| Use Case Name: Reviewing Booking History |
| Actor(s): User |
| Brief Description: This use case involves users reviewing their past bookings on the platform. |
| Basic Flow: 1-The user navigates to the "Booking History" section. 2-The user views a list of their previous bookings, including details such as dates and chalet names. 3-The user can click on a booking to view more information. |
| Alternative Flow: If there are no previous bookings, the system displays a message indicating no history. |
| Pre-conditions: 1-User must be logged in. 2-The user must have made at least one booking. |
| Post-conditions: N/A |

Table 6- Updating Chalet Info Case

|  |
| --- |
| Use Case Name: Updating Chalet Information |
| Actor(s): Owner |
| Brief Description: This use case involves owners updating information about their chalet listings. |
| Basic Flow: 1-The owner accesses the "Manage Chalets" section. 2-The owner selects the chalet they want to update. 3-The owner edits information such as description, amenities, or pricing. 4-The owner saves the changes. |
| Alternative Flow: If the owner wants to remove a chalet listing, they can choose to delete it entirely from the platform. |
| Pre-conditions: 1-Owner must be logged in. 2-The owner must have access to the "Manage Chalets" functionality. |
| Post-conditions: The chalet information is updated on the platform. |

Table 7- View Services Case

|  |
| --- |
| Use Case Name: Viewing Chalet Services |
| Actor(s): User |
| Brief Description: This use case involves users viewing detailed information about the services offered at a particular chalet. |
| Basic Flow: 1-The user navigates to the chalet details page. 2-The user scrolls down to the "Services" section. 3-The user reads descriptions of the services provided, such as catering, housekeeping, or recreational activities. |
| Alternative Flow: If the user is interested in a specific service, they can click on it to view more details. |
| Pre-conditions: N/A |
| Post-conditions: N/A |

Table 8- Giving Feedback Case

|  |
| --- |
| Use Case Name: Providing Feedback |
| Actor(s): User |
| Brief Description: This use case involves users providing feedback about their experience using the platform or a specific chalet. |
| Basic Flow: 1-The user navigates to the "Feedback" section. 2-The user fills out a feedback form, rating their experience and leaving comments if desired. 3-The user submits the feedback. |
| Alternative Flow: If the user encountered any issues during their booking process, they can choose to describe them in detail in the feedback form. |
| Pre-conditions: 1-User must be logged in. 2-The user must have completed at least one booking. |
| Post-conditions: The feedback is recorded and may be reviewed by platform administrators or chalet owners. |

Table 9-Managing Property Status

|  |
| --- |
| Use Case Name: Managing Property Status |
| Actor(s): Admin |
| Brief Description: This use case involves admins managing the status of chalet properties on the platform. |
| Basic Flow: 1-The admin accesses the admin dashboard. 2-The admin views a list of all chalet properties and their current status. 3-The admin can change the status of a property to "Available" or "Not Available" to users. |
| Alternative Flow: If a property needs to be temporarily removed from the platform, the admin can choose to deactivate it entirely. |
| Pre-conditions: 1-Admin must be logged in. 2-The admin must have access to the admin dashboard. |
| Post-conditions: The property status is updated on the platform. |

Table 10- Listing Chalet Case

|  |
| --- |
| Use Case Name: Listing a Chalet |
| Actor(s): Owner, Administrator |
| Brief Description:  This Use case is enrolling your own property on the website |
| Basic Flow:  1-Owner chooses List your property option.  2-Owner uploads required details.  3-Owner submits enrollment |
| Alternative Flow:  1-Admin receives now enrollment.  2-Admin reviews Owner’s uploaded info.  3-Admin sets property status (available/not available to users). |
| Pre-conditions:  1-Creates an owner account.  2-Logged in |
| Post-conditions:  1-Admin may request a correction/ addition on property information and keeps status not available.  2-Admin accepts uploaded information and sets status available for users. |

Table 11- Manage Bookings Case

|  |
| --- |
| Use Case Name: Managing Bookings |
| Actor(s): Owner |
| Brief Description: This use case involves owners managing bookings for their chalets. |
| Basic Flow: 1-The owner accesses the "Manage Bookings" section. 2-The owner views a list of current bookings for their chalet(s). 3-The owner can approve, deny, or modify booking requests. |
| Alternative Flow: If a booking request conflicts with existing bookings, the owner can suggest alternative dates to the user. |
| Pre-conditions: 1-Owner must be logged in. 2-Owner must have access to the "Manage Bookings" functionality. |
| Post-conditions: The booking status is updated accordingly based on the owner's actions. |

Table 12- Review Owner Account

|  |
| --- |
| Use Case Name: Reviewing New Owner Accounts |
| Actor(s): Admin |
| Brief Description: This use case involves admins reviewing and approving new owner accounts on the platform. |
| Basic Flow: 1-The admin accesses the admin dashboard. 2-The admin views a list of pending owner account registrations. 3-The admin verifies the provided information and approves or denies the account. |
| Alternative Flow: If additional information is needed, the admin can request it from the owner before making a decision. |
| Pre-conditions: 1-Admin must be logged in. 2-New owner account registrations must exist in the system |
| Post-conditions: The owner account status is updated based on the admin's decision. |

Table 13- View Chalet Booking

|  |
| --- |
| Use Case Name: Viewing Chalet Bookings |
| Actor(s): Owner |
| Brief Description: This use case involves owners viewing bookings for their chalets. |
| Basic Flow: 1-The owner accesses the booking management section. 2-The owner views a list of current and upcoming bookings for their chalets. 3-The owner can filter bookings by date, status, or chalet. |
| Alternative Flow: If needed, the owner can export booking data for further analysis or record-keeping. |
| Pre-conditions: 1-Owner must be logged in. 2-Bookings for the owner's chalets must exist in the system. |
| Post-conditions: N/A |

Table 14- View Customer Information Case

|  |
| --- |
| Use Case Name: Checking Customer Information |
| Actor(s): Owner |
| Brief Description: This use case involves owners checking the information provided by customers during booking requests. |
| Basic Flow: 1-The owner accesses the booking management section. 2-The owner views details of booking requests, including customer information. 3-The owner verifies the accuracy of the provided information before approving or denying the booking. |
| Alternative Flow: If there are discrepancies or missing information, the owner may contact the customer for clarification. |
| Pre-conditions: 1-Owner must be logged in. 2-Booking requests from customers must exist in the system. |
| Post-conditions: The booking request status is updated based on the owner's decision. |

Chapter 3

This chapter describes the system design, including the data base of our project, the related tables and the relations between them.

3.1 System Design

3.1.1- ER-Diagram

This ER (Entity Relationship) Diagram represents the model of the chalet booking system. It shows all the instruments of database tables and the relations between them. The main entities of the chalet booking system are: **bookings**, **caccount**, **chalet**, **chalet\_pictures**, **chalet\_services**, **customer**, **oaccount**, **owner**, and **services**.

Chalet booking system entities and their attributes:

● **Bookings:** id, cid, cuid, booking\_date, status with id as primary key and cid and cuid as foreign keys.

● **Caccount:** id, username, password, cid with id as primary key and cid as foreign key.

● **Customer:** id, fname, lname, phone, email, address with id as primary key.

● **Oaccount:** id, username, password, oid with id as primary key and oid as foreign key.

● **Owner:** id, fname, lname, email, phone, address with id as primary key.

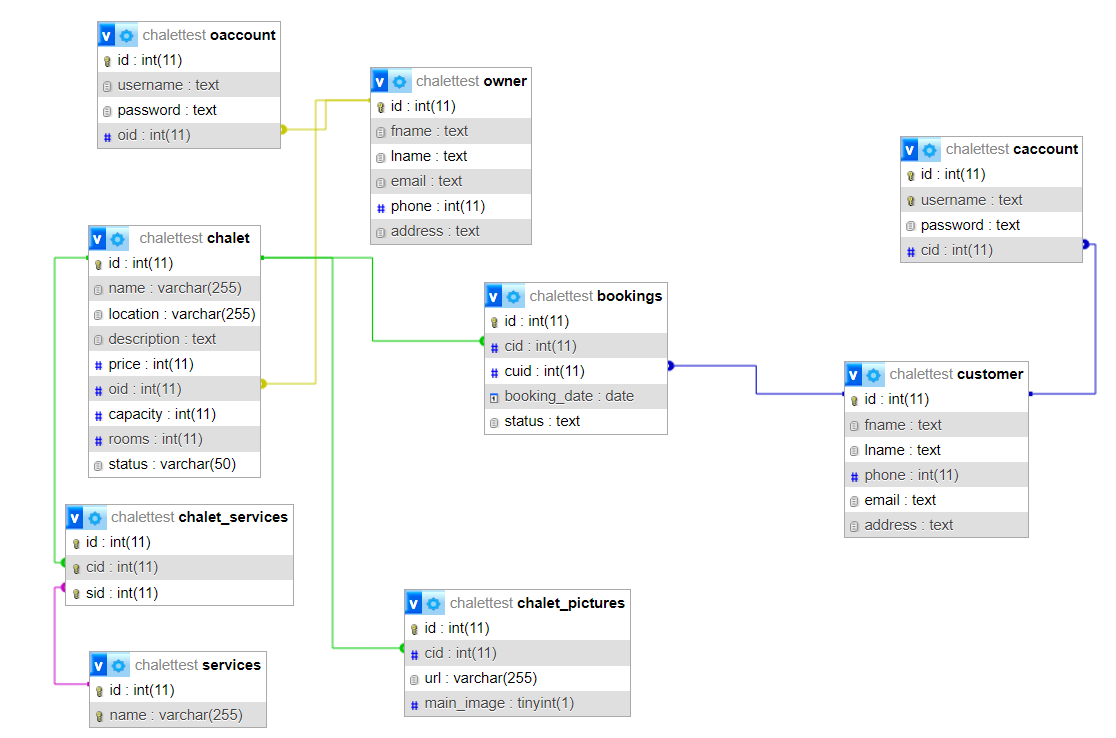
● **Chalet:** id, name, location, description, price, oid, capacity, rooms, status with id as primary key and oid as foreign key.

● **Chalet\_services:** id, cid, sid with id as primary key and cid and sid as foreign keys.

● **Chalet\_pictures:** id, cid, url, main\_image with id as primary key and cid as foreign key.

●**Services:** id, name with id as primary key

* Relational ER Diagram:



Chapter 4

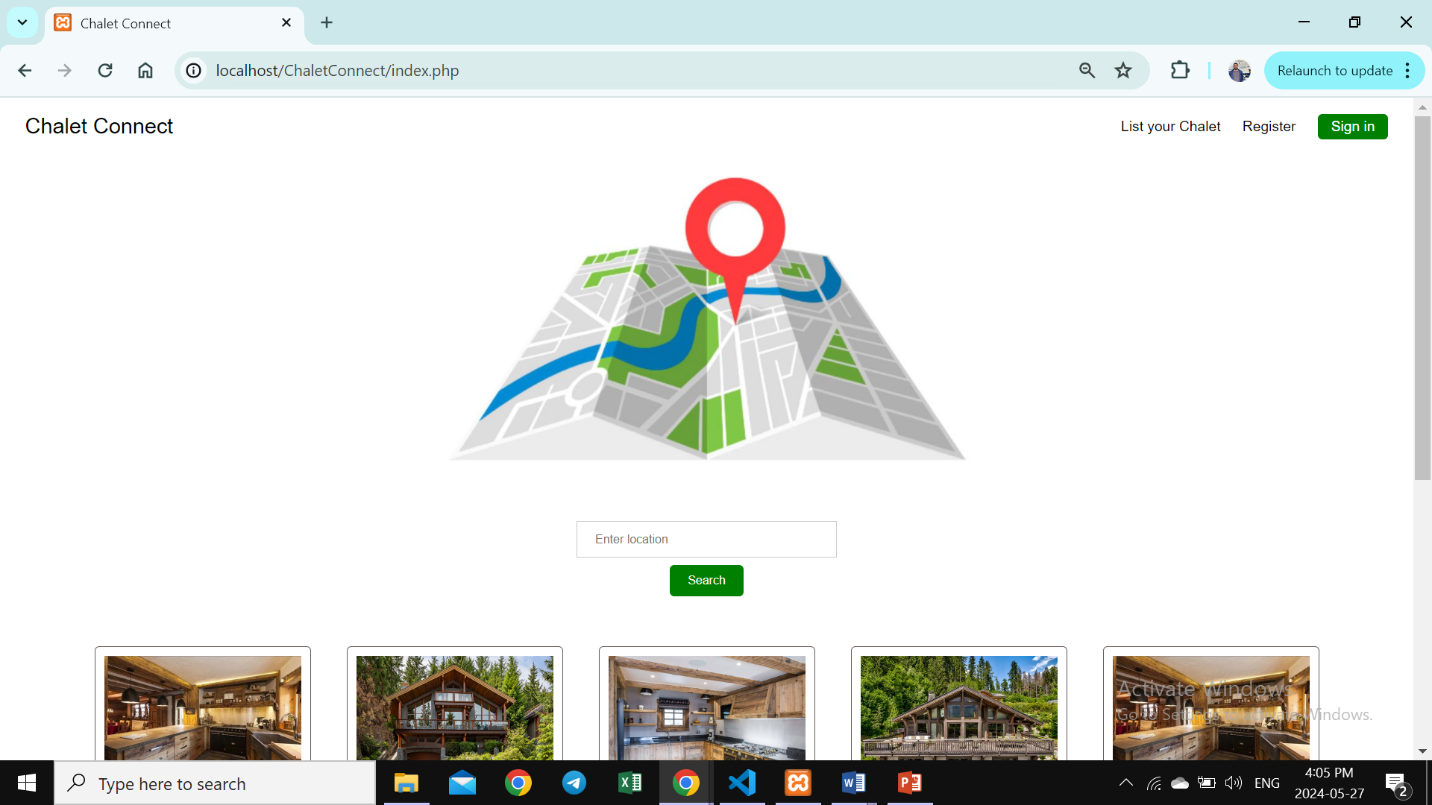
This chapter describes the whole view of our website, it shows the design our projects in addition to stating the main role of each page or button.

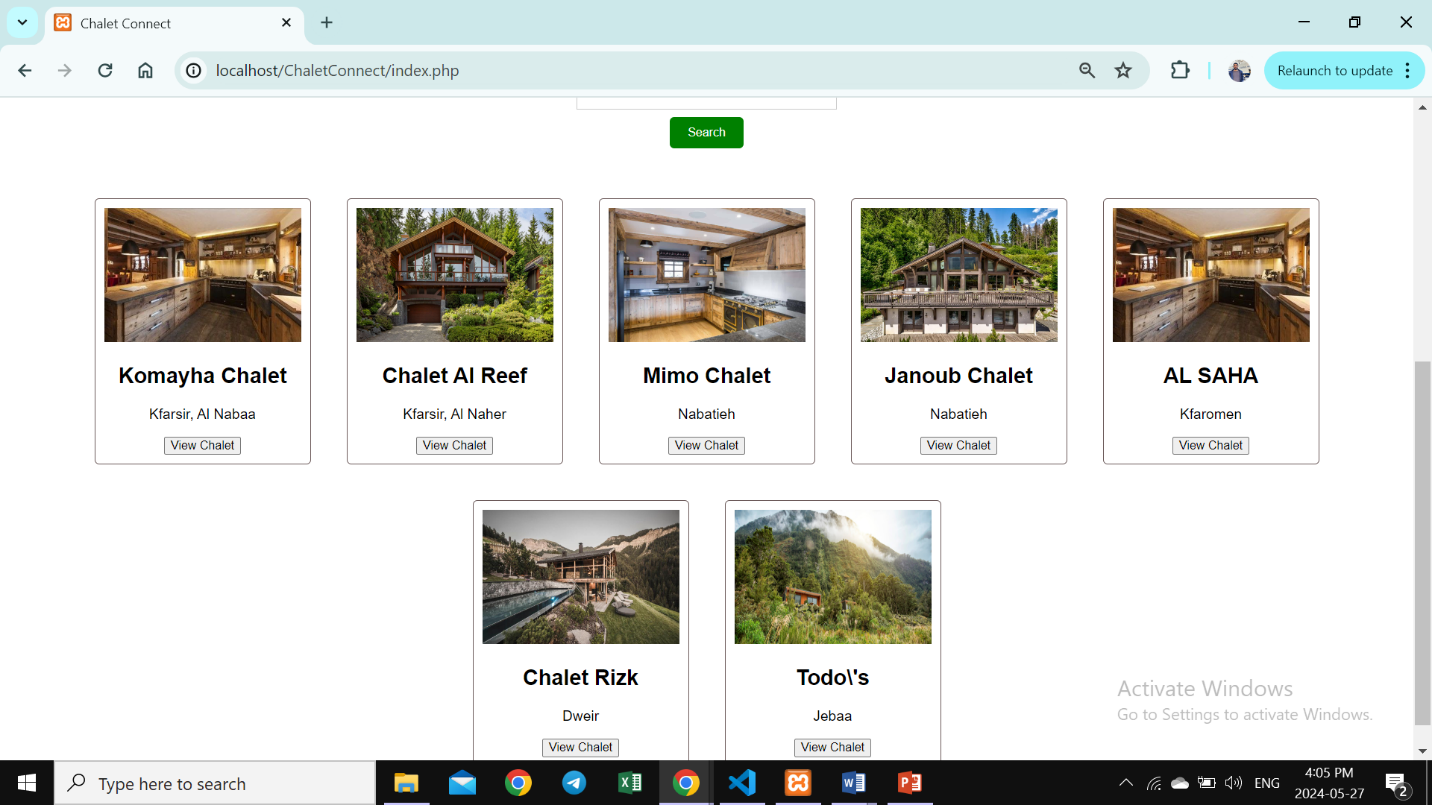
Implementation:

The implementation of our chalet booking system demonstrates the user interface design and functionality to ensure optimal performance and service for the customer. Our goal is to create a user-friendly interface suitable for all ages, emphasizing simplicity and visual comfort. The design allows users to navigate effortlessly through the website without encountering any difficulties or interruptions.

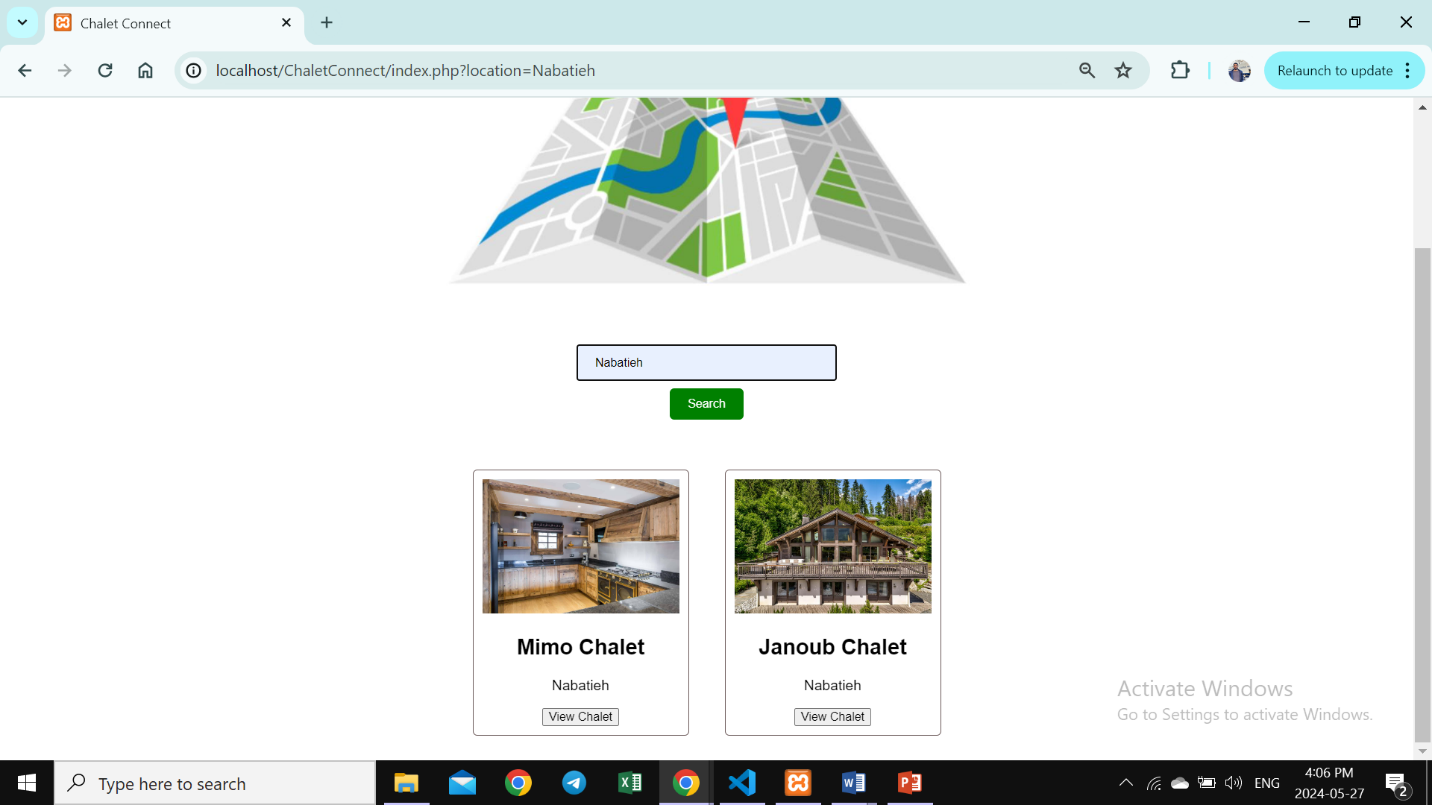
Below, we will detail the different pages of our website and their respective functionalities.

1 – Index page, which can be fully viewed and explored without logging in

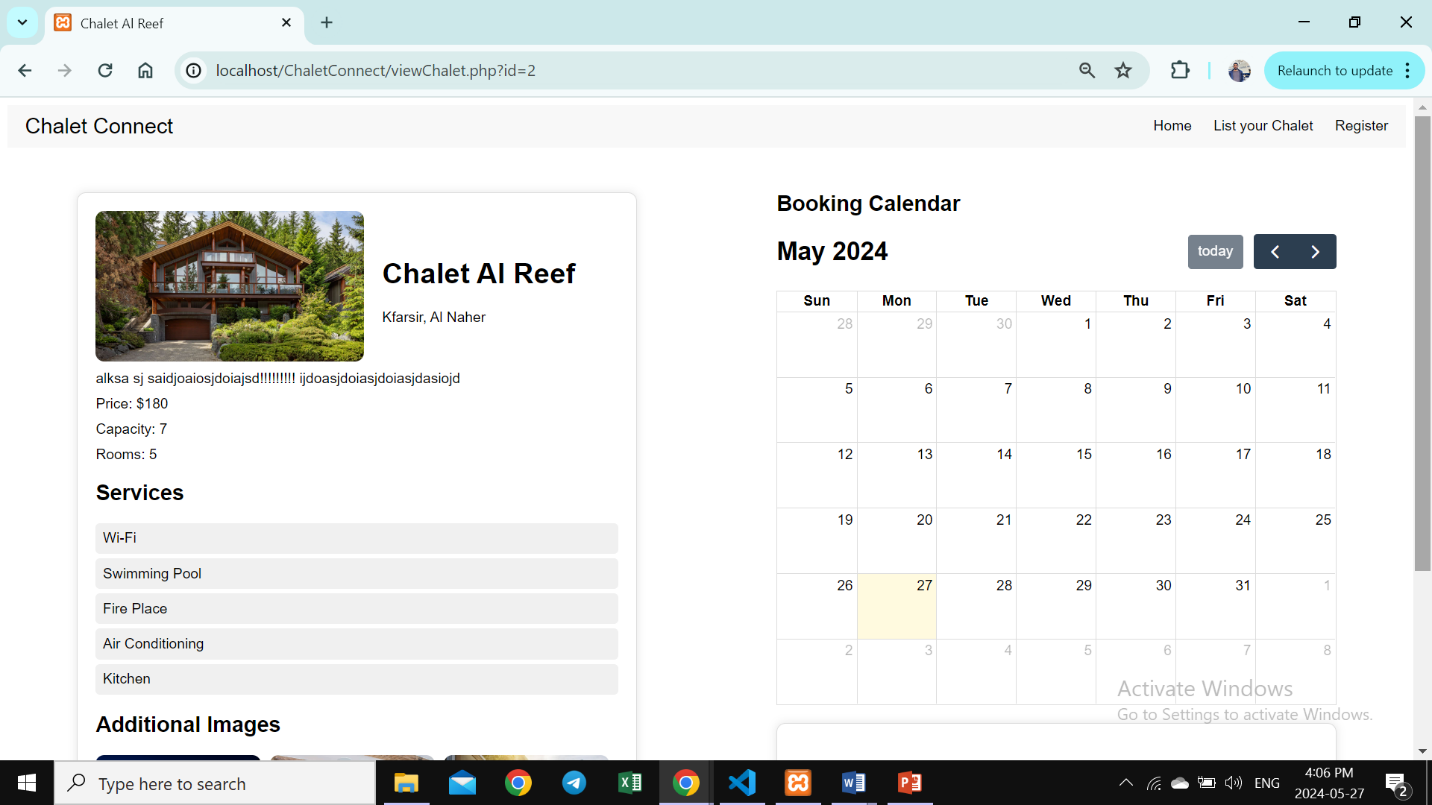




2- Searching for Chalets by location

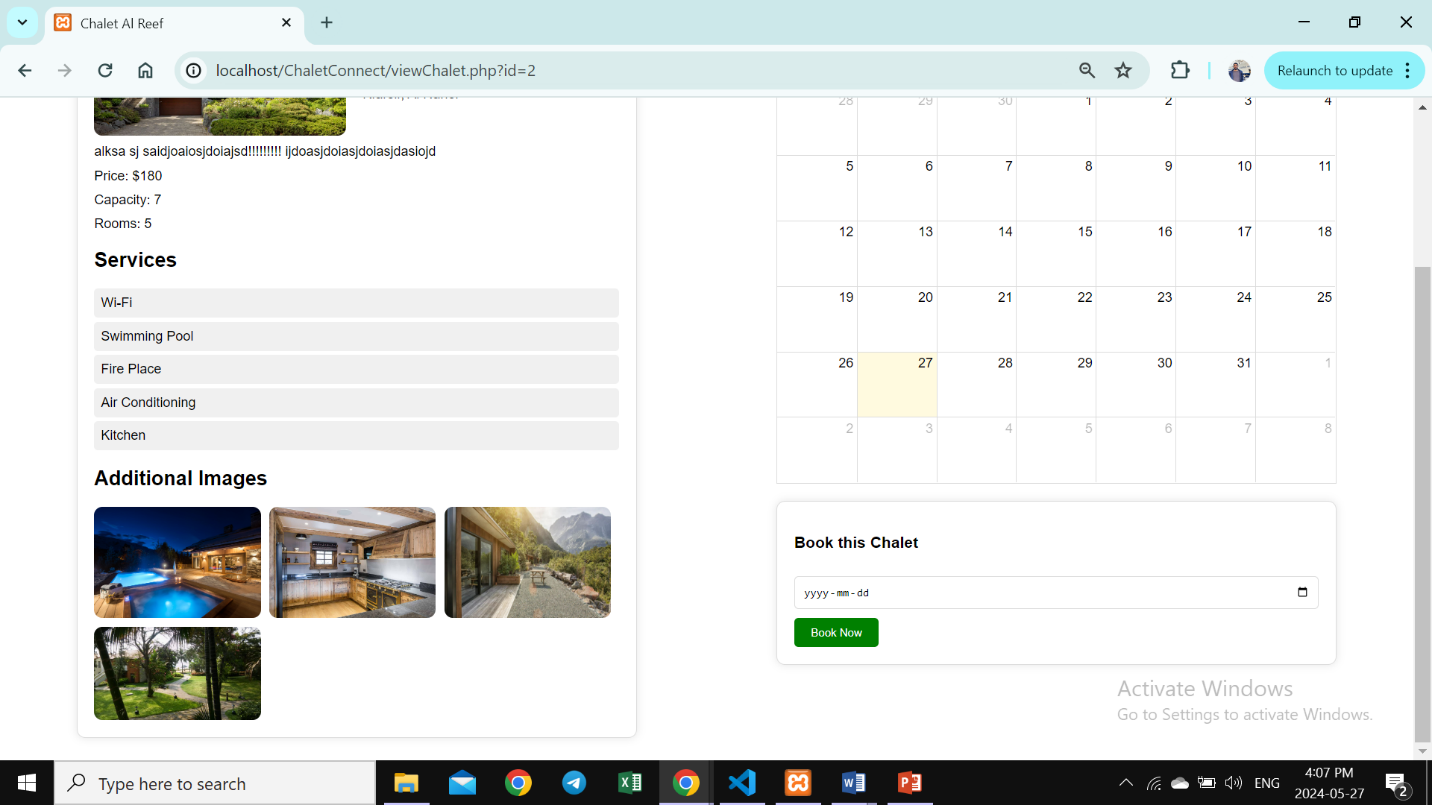


3- Pressing “View Chalet” will take you to specific chalet’s page

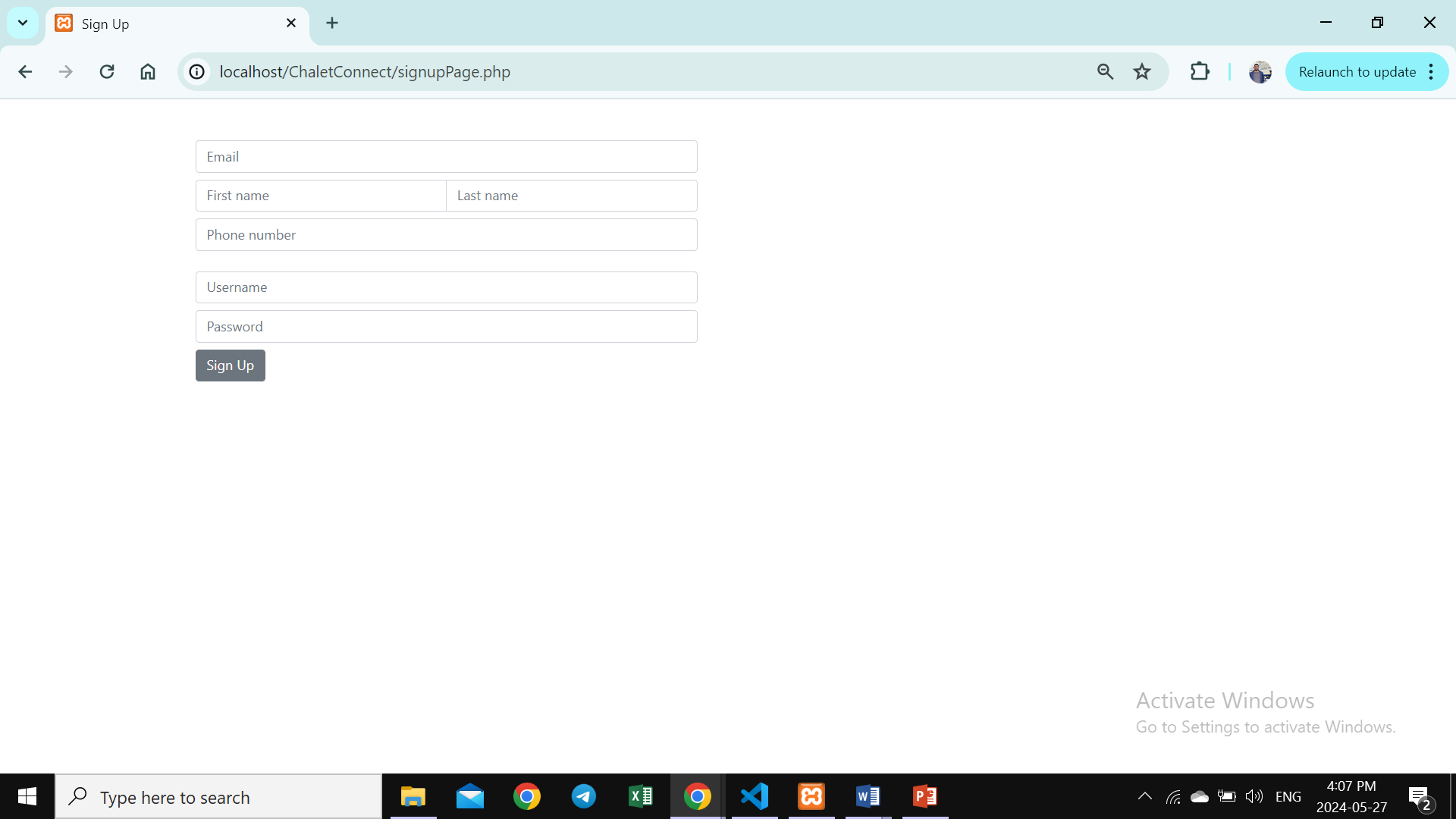


4- The calendar must show the available dates for booking.

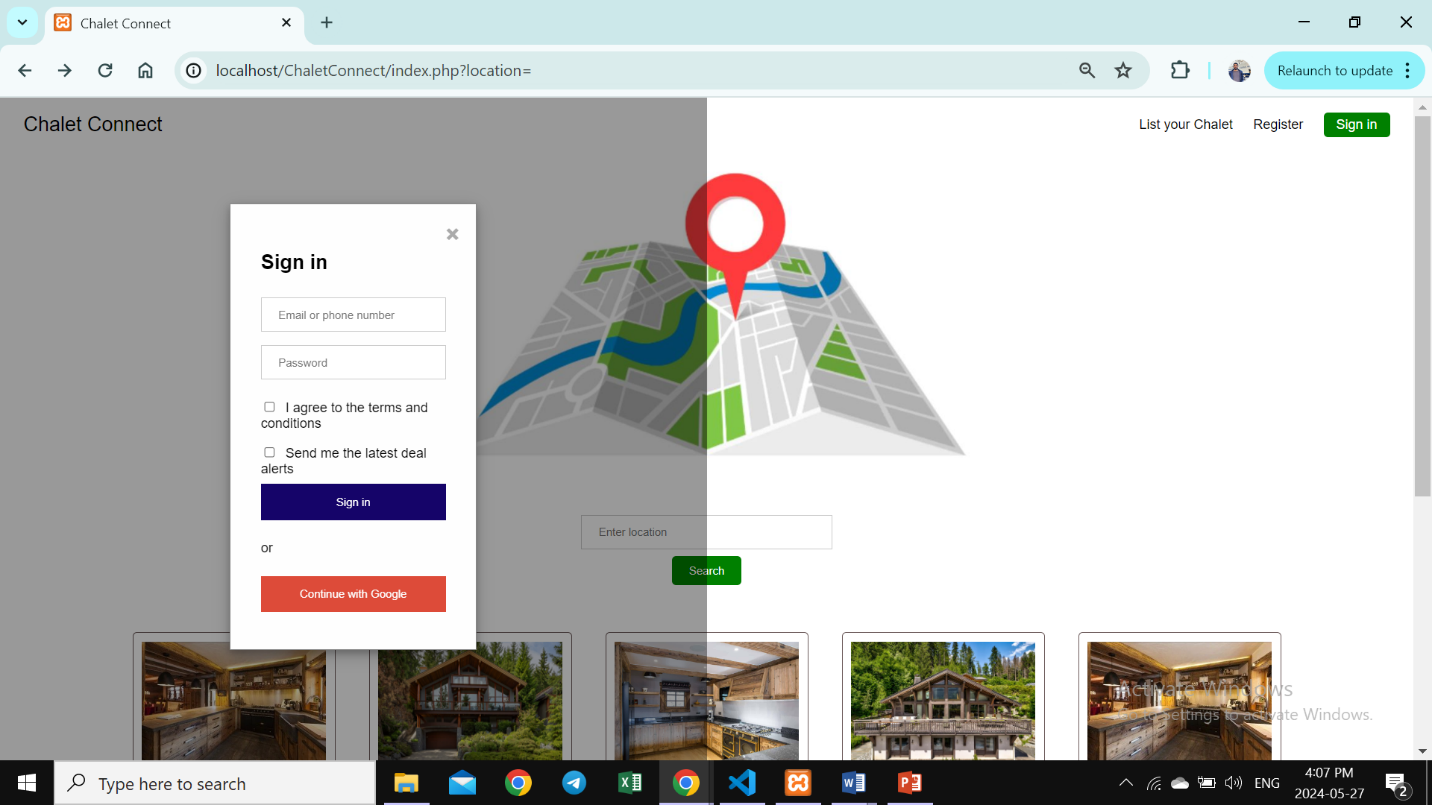
Choosing a date and pressing Boo Now will send a request to chalet’s owner (user must log in).



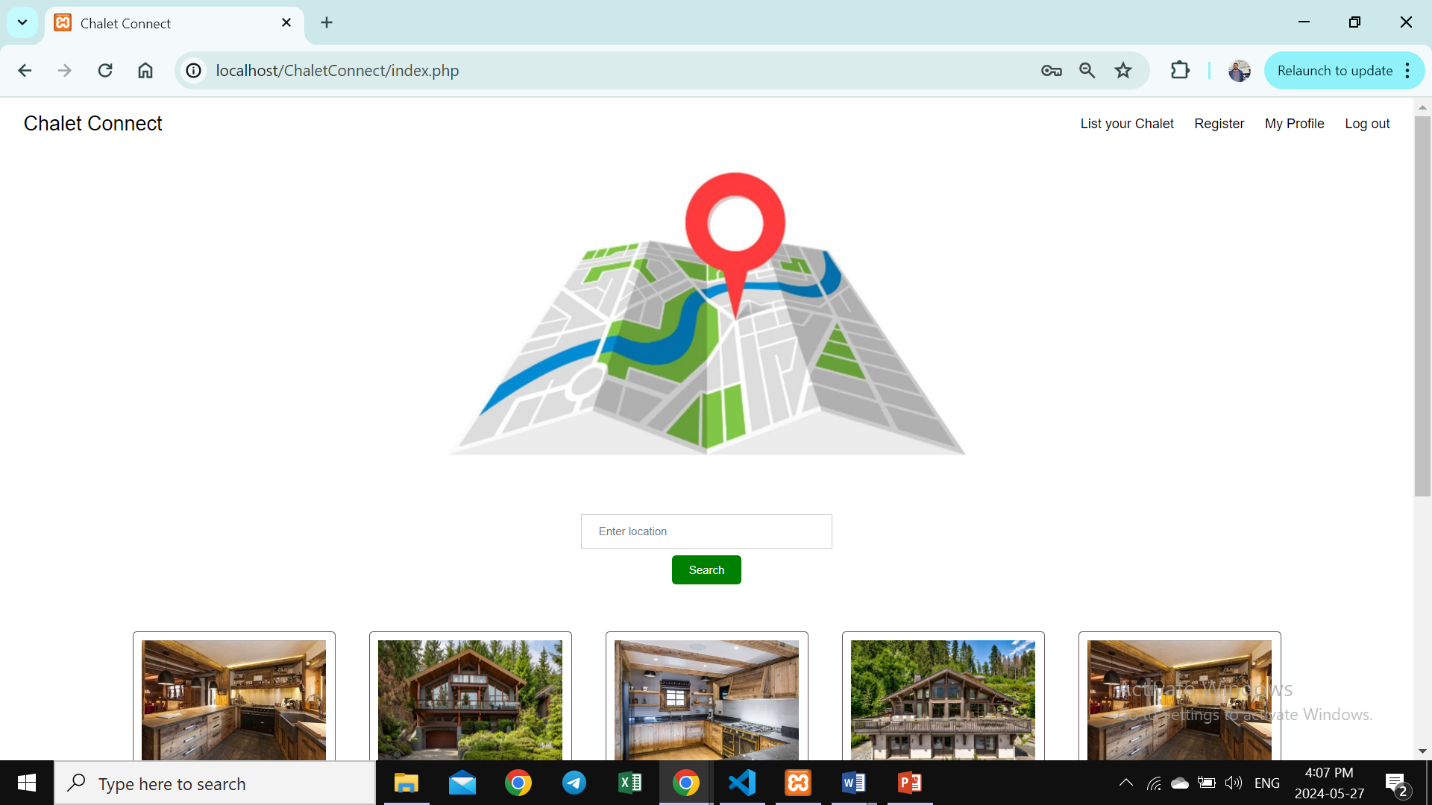
5- Sign Up page after pressing Register



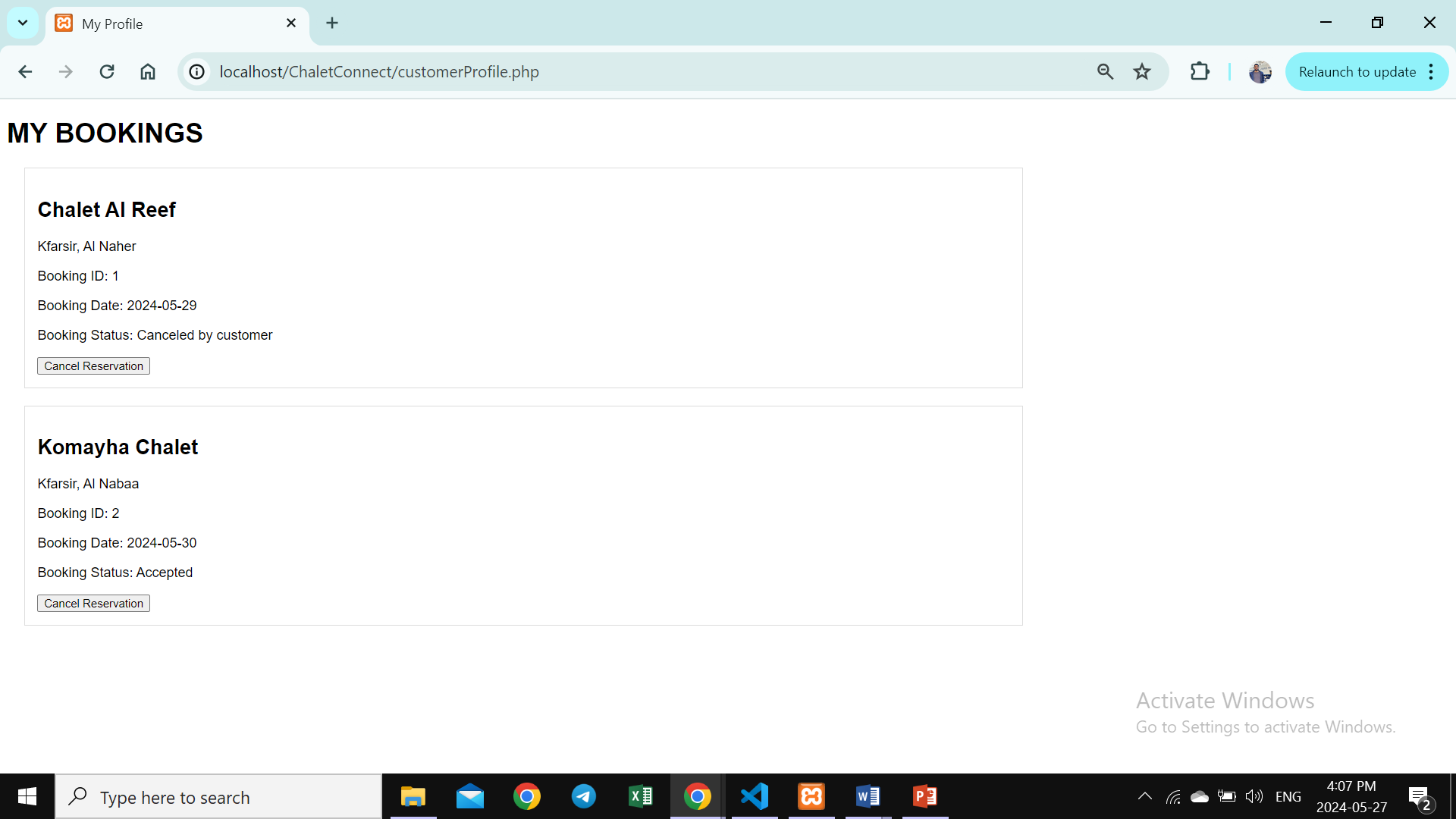
6- Sign in Modal after pressing sign in

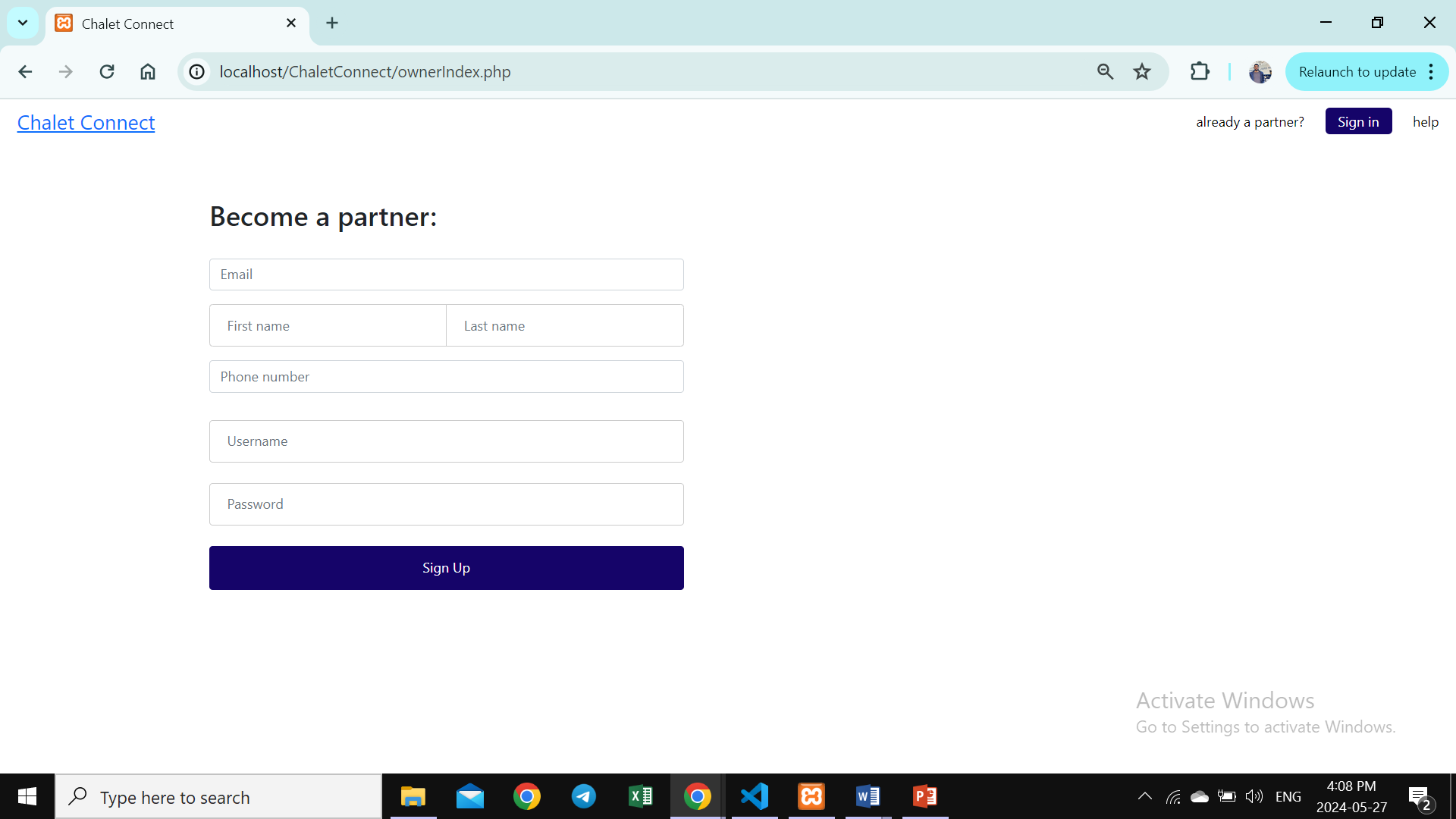
.

After Logging in, My Profile appears

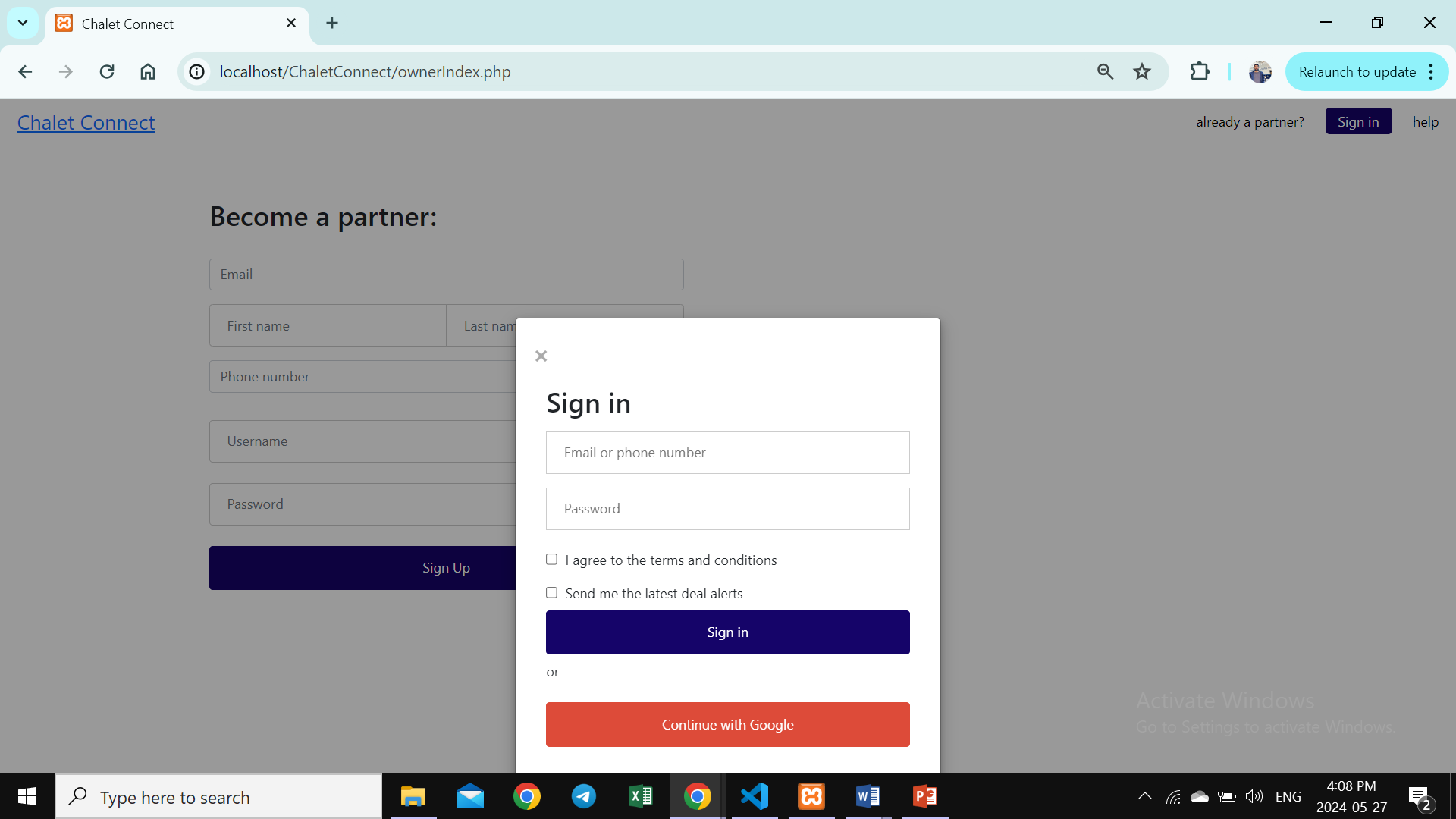


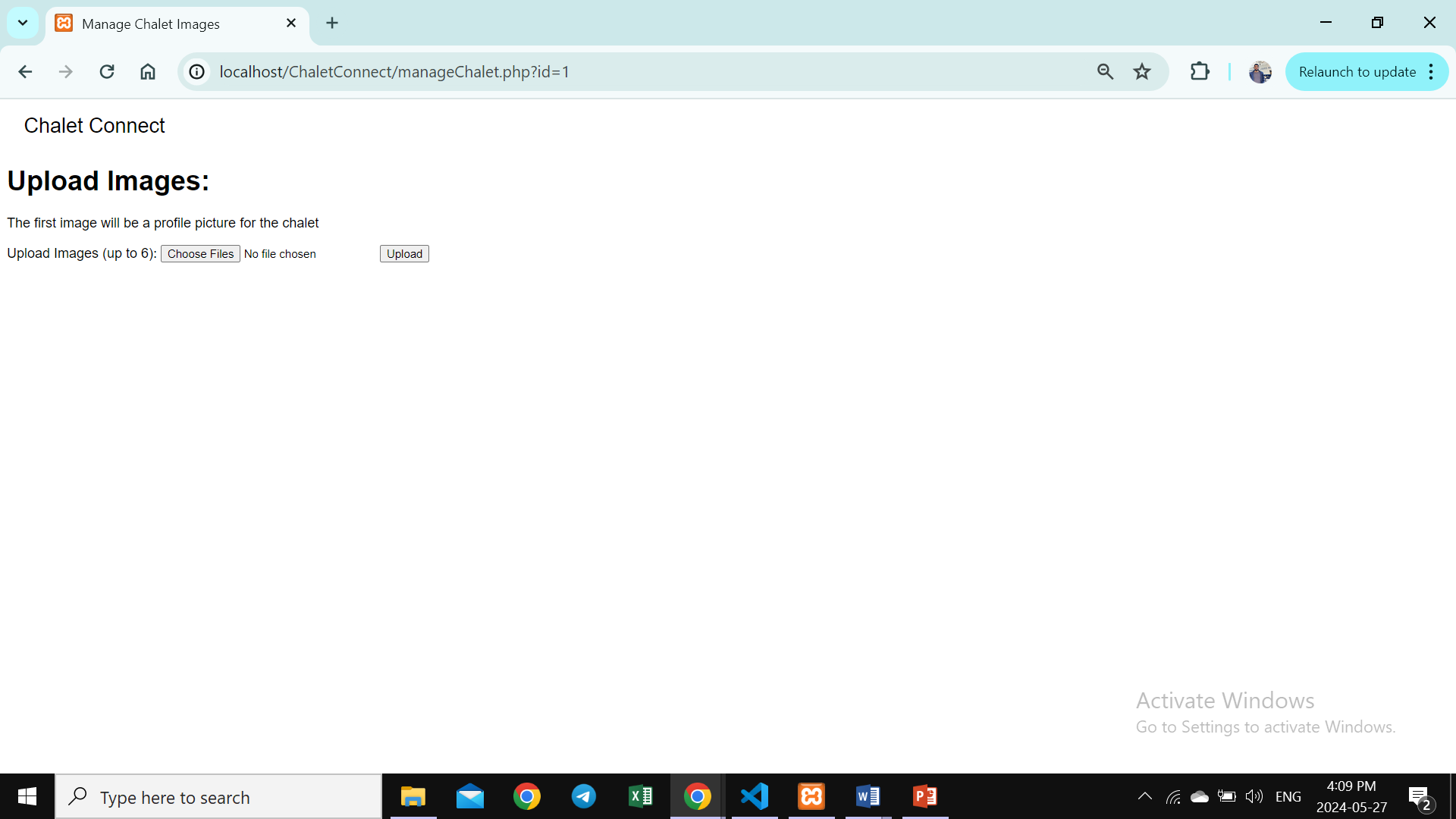
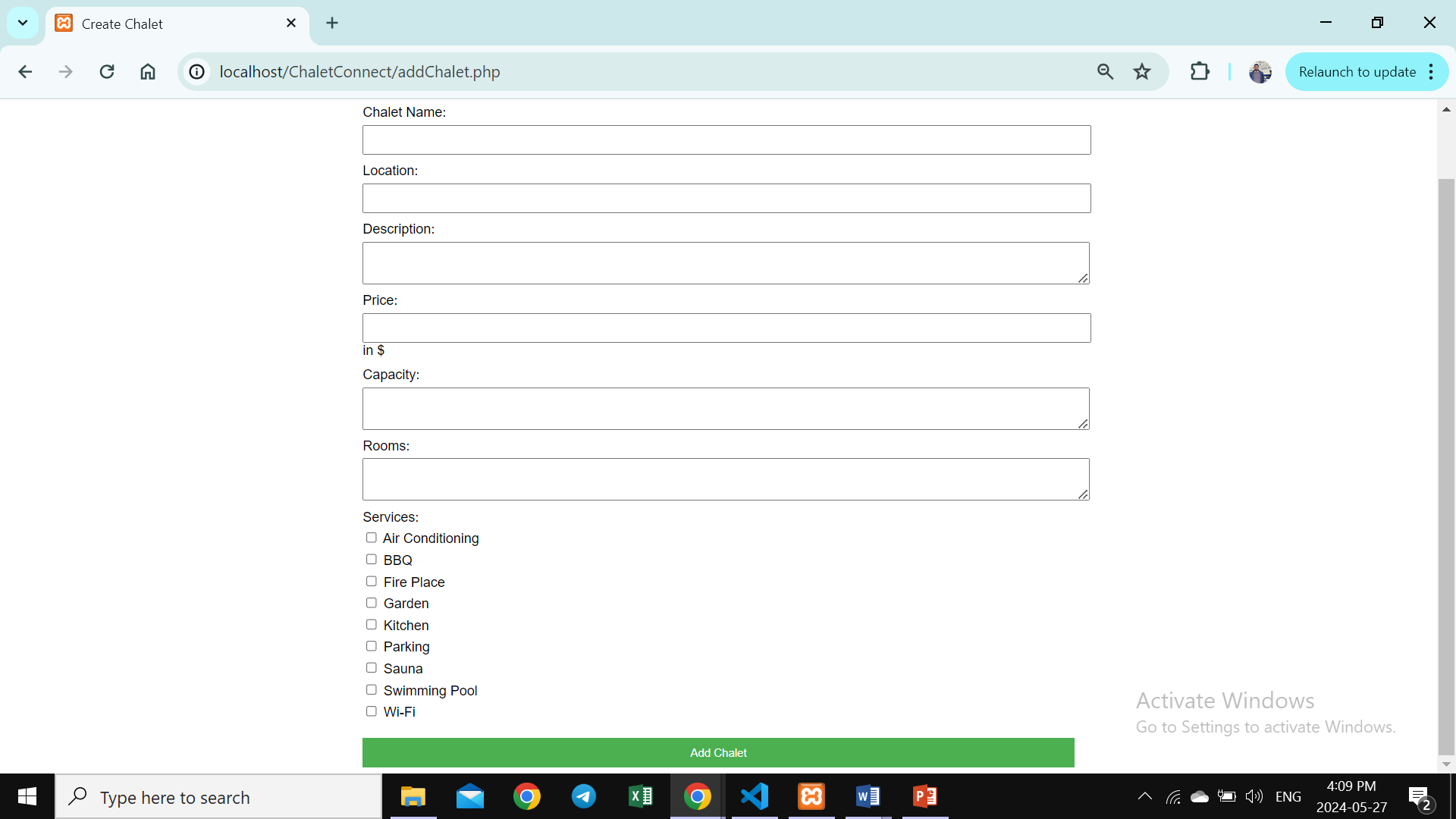
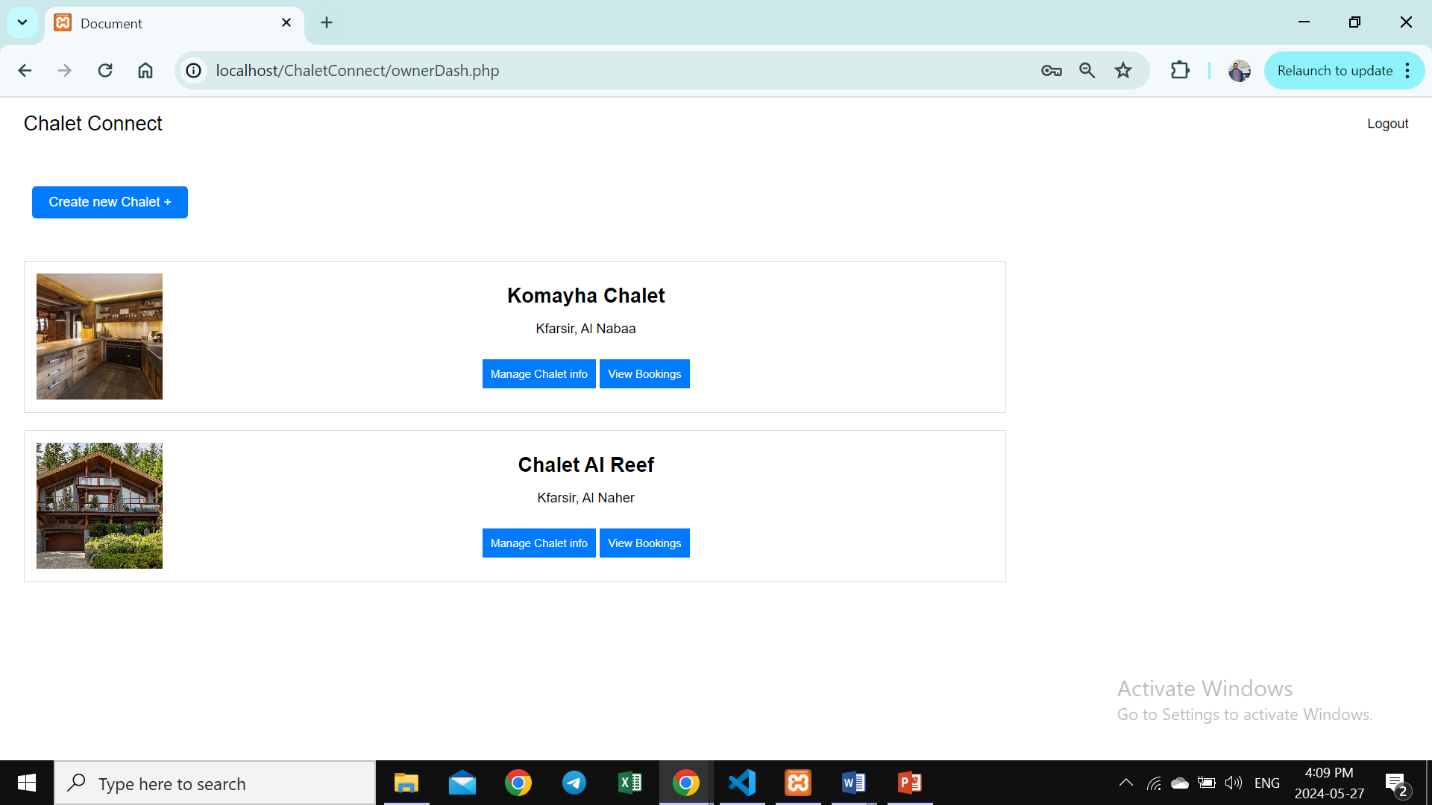
7- Entering My profile will view Customer’s bookings with ability to cancel reservation.



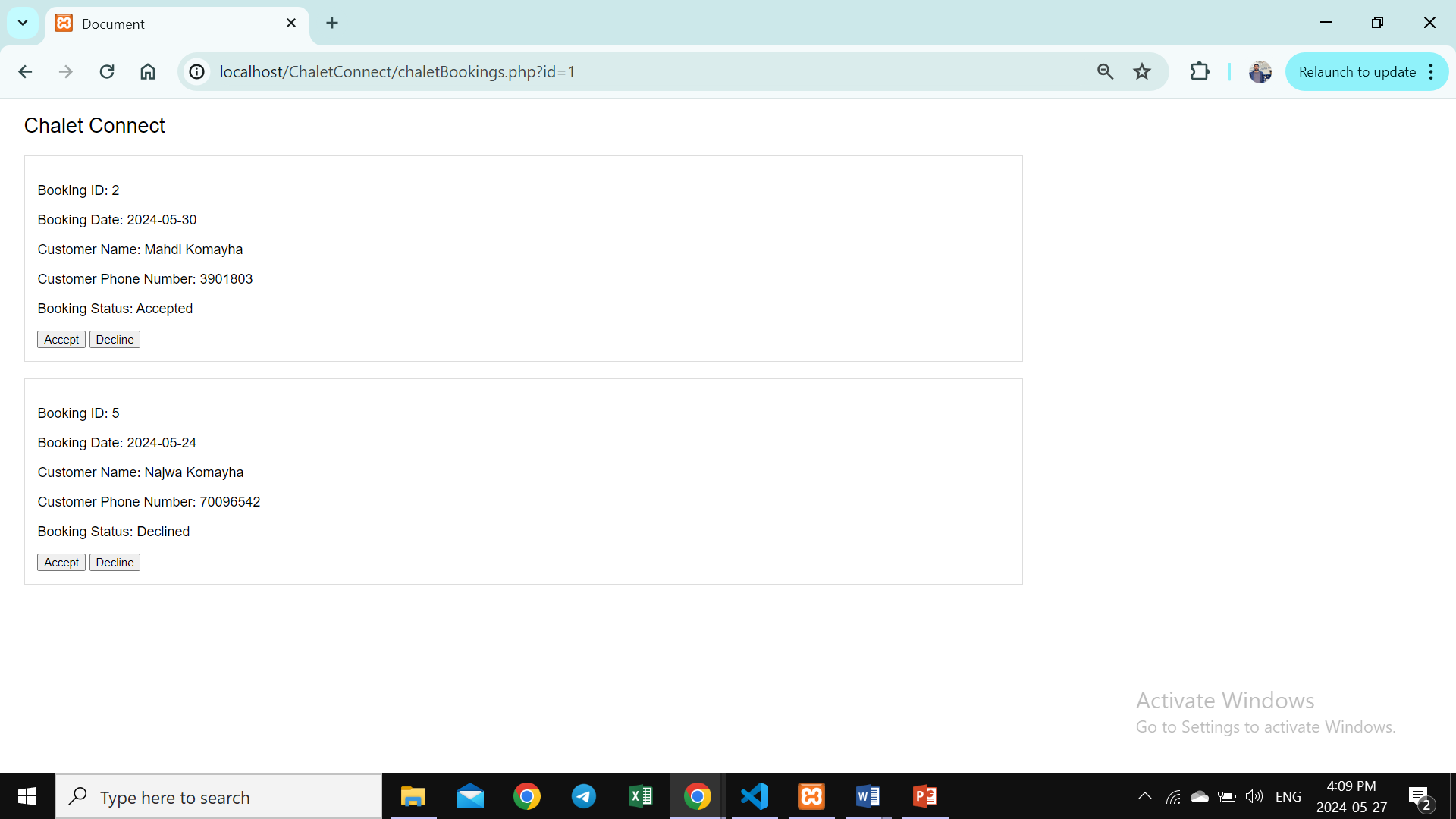
8- Pressing “List Your Chalet” will navigate you to owner’s index page which contains an owner sign up form.

9-Owner Sign in modal



10-Owner Dashboard shows all your listed chalets

11- View All Bookings for a certain account



Chapter 5

5.1- Conclusion

Our aim was to design and implement a chalet booking system. Through our research, we found that an increasing number of people prefer booking accommodations online rather than through traditional methods. This website allows chalet owners to keep up with current trends and technology, offering users a convenient and efficient way to book chalets.

5.2- Future Work

In the future, we want to improve the booking experience and gain a deeper understanding of customer preferences by incorporating advanced analytics and user feedback mechanisms into the system. Furthermore, our goal is to create a mobile application that will improve accessibility and convenience for people who are always on the go. Implementing AI-driven user suggestions and providing promotional offers to boost engagement and reservations are two more possible improvements. These upgrades will contribute to the robustness and ease of use of our chalet reservation system.

Top of Form