**PROJECT TECHNICAL REPORT-WEB DESIGN 2023.**

 University of Rwanda

College of Science and Technology

Computer Engineering

Web Design.

MUGABO Kevia : 223007592

Year of study; Year 1

Date: 20th Nov,2023

**PROJECT TECHNICAL REPORT.**

**EAT WELL WEB APPLICATION.**

**Table of contents.**

[**CHAPTER 1: INTRODUCTION 3**](#_yfocshw36qmn)

[1.1 BACKGROUND CASE STUDY 3](#_gl4bcw4o2gx)

[1.2 PROBLEM STATEMENT. 3](#_kxrvjq687xz0)

[1.3 OBJECTIVE OF MY PROJECT. 4](#_vqt5mkltodri)

[1.4 PROJECT RATIONALE. 4](#_8eo2wi1ynh5e)

[1.5 PROJECT LIMITATIONS. 5](#_c0fvzacvyc1y)

[1.6 IMPLEMENTATION PLAN. 5](#_av5zw9x7aqu4)

[**CHAPTER 2: SYSTEM ANALYSIS AND DESIGN. 7**](#_taon0nf1p693)

[2.1 SYSTEM ANALYSIS 7](#_qgogi6swexrr)

[2.1.1 Introduction to system analysis and design. 7](#_d9f8we43nc0y)

[2.1.2 System actors(internal and external) 8](#_jqxpixg6byqi)

[2.1.2 Functional Requirements Of Project. 9](#_flocyzsau45j)

[2.1.3 Non-functional Requirements 11](#_s7ofwi91xms3)

[**Chapter 3: Introduction 12**](#_r0tha6yzrvo3)

[3.1 Tools and technology. 12](#_gme7bcrhasx7)

[3.2. Screenshots 13](#_efez49vd9khg)

[**Appendix. 20**](#_lreaf4qfgrv)

# **CHAPTER 1: INTRODUCTION**

## **1.1 BACKGROUND CASE STUDY**

The “EAT WELL” platform is a web application designed to revolutionise the way the college community interacts with on-campus dining services. The environment of a college campus is not only a hub of academic activities but also a melting pot of diverse culinary preferences. With a multitude of students, faculty, and staff coming together, there is a need for an effective and user-friendly platform that can serve a smooth dining experience on campus. Recognizing that college dining experience involves long queues, paper-based menu systems, and limited visibility into daily specials and nutritional information, the lack of a centralised platform often leads to frustration among students and staff, resulting in valuable time wasted and a substandard dining experience.

**Key elements of the background include:**

**Reduced queues and wait times:** To put into consideration the time saving principle, “EAT WELL” can enable pre-ordering and reservation systems, reducing wait times and allowing users to schedule their meals efficiently**.**

**Accessibility to Menu Information:** The platform can provide a digital menu with real-time updates, nutritional information, and filters for dietary preferences or restrictions providing nutritional details to college stakeholders.

**Feedback Mechanism:** By incorporatinga feedback system within the application, allowing users to rate dishes, provide comments, and suggest improvements. This can help enhance the overall dining experience.

## **1.2 PROBLEM STATEMENT.**

In college, dining experience involves long queues, paper-based menu systems, and limited visibility into daily specials and nutritional information, the lack of a centralised platform often leading to frustration among students and staff, resulting in valuable time wasted and a substandard dining experience.

**The specific challenges addressed by “EAT WELL” include:**

**Long queues and wait hours**: Traditional ordering systems may lead to long queues during peak dining hours, causing frustration and delays.

**Limited accessibility to menu information**: Students and staff may not have easy access to up-to-date menu information, including specials and nutritional details and changes in offerings.

**Ineffective order processing**: Manual order processing and payment systems can be time-consuming and more prone to errors.

**Diverse dietary need**: Meeting the diverse dietary preferences, including allergies, cultural preferences, and specific dietary restrictions.

## **1.3 OBJECTIVE OF MY PROJECT.**

* **GENERAL OBJECTIVE.**

The general objective of the “EAT WELL” platform is to raise the overall dining experience for the college community by manipulating technology to address various challenges like long queues and inadequate information.

* **SPECIFIC OBJECTIVES.**
* Real-time menu updates.
* Online ordering.
* Reservation systems.
* Mobile-friendly interface.
* Integration with campus ID cards.

## **1.4 PROJECT RATIONALE.**

The development and implementation of “EAT WELL” are a rational need for an innovative and inclusive solution to address the dining challenges faced by the college community.

1. **Meeting user expectations**: With users expecting seamless and high-tech solutions, the “EAT WELL” platform aligns with these expectations, providing a modernised and convenient dining experience for the college community.
2. **Improving operational efficiency**:Traditional restaurant operations often face inefficiencies, leading to long queues and delays. “EAT WELL” platform aims to simplify operations, reduce wait times, and improve efficiency in food services.
3. **Addressing diverse dietary needs**: The college community consists of individuals with diverse dietary preferences and restrictions. The application’s customization features and detailed nutritional information cater to these diverse needs, promoting inclusivity.
4. **Optimising resource utilisation**: With traditional order and payment systems may lead to resource wastage. Online ordering, reservation system, and secure online payments optimise resource utilisation, reducing manual processes and potential errors.

## **1.5 PROJECT LIMITATIONS.**

As “EAT WELL” aims to address key challenges in the college's dining services, it is crucial to manage expectations and ensure that the college community understands certain constraints that may influence the project’s scope and implementation. These limitations include:

**Limited internet connectivity**: For clear operation of the application there is need for reliable internet access which may sometimes not be accessed by all college community members.

**Integration challenges**: There might be technical hurdles in coherent integration with existing campus systems such as ID cards and payment systems.

**Adoption rates**: The success of the application depends on user adoption which is challenging .

**Financial constraints**: Budget constraints may limit the scope and features of the application.

**Sustainability limitations**: Achieving sustainable goals may have limitations based on available resources such as packaging waste among others.

## **1.6 IMPLEMENTATION PLAN.**

The successful implementation of the "EAT WELL" platform requires a systematic and well-coordinated plan. The following steps outline a comprehensive implementation strategy:

**1. Initiation :**

* Define project scope, objectives, and success criteria.
* Form a project team, including developers, designers, and key stakeholders.
* Conduct a kick-off meeting to align everyone on project goals and expectations.

**2. Requirements Gathering and Analysis:**

* Collaborate with key stakeholders to gather detailed requirements.
* Conduct user surveys and interviews to understand user preferences and expectations.
* Create user personas and scenarios to inform design and development.

**3. Design Phase :**

* Develop wireframes and prototypes based on gathered requirements.
* Conduct usability testing with a focus group to refine the user interface.
* Finalise the visual design, including branding elements and user interface components.

**4. Development :**

* Start front-end development based on the approved design.
* Simultaneously, initiate back-end development for order processing, menu management, and database integration.
* Implement security measures and payment gateway integration.

**5. Testing :**

* Conduct comprehensive testing, including functional, usability, security, and performance testing.
* Address any bugs or issues identified during testing.
* Collaborate with a small group of beta users for further testing and feedback.

**6. Integration with Campus Systems :**

* Work closely with campus IT to integrate the web application with existing systems such as ID cards and payment systems.
* Test and ensure seamless integration, resolving any compatibility issues.

**7. User Training :**

* Develop training materials for both end-users and restaurant staff.
* Conduct training sessions to familiarise users with the application features and processes.
* Provide ongoing support and resources for questions or issues.

**8. Soft Launch :**

* Conduct a soft launch on a limited scale to test real-world usage.
* Gather feedback from users and address any issues or concerns.
* Monitor application performance and user engagement during the soft launch.

**9. Refinement and Optimization :**

* Analyse data from the soft launch to identify areas for improvement.
* Implement necessary refinements and optimizations based on user feedback.
* Conduct additional testing to ensure the effectiveness of changes.

**10. Full Deployment :**

* Conduct a full-scale deployment of the web application across the entire college community.
* Monitor system performance, user adoption rates, and any issues that may arise during the initial days of full deployment.

**11. Marketing and Promotion :**

* Launch a marketing campaign to promote the web application to the entire college community.
* Utilise various channels, such as posters, social media, and campus events, to raise awareness and encourage adoption.

**12. Ongoing Support and Maintenance :**

* Establish a support system for users to report issues or seek assistance.
* Implement a schedule for regular maintenance, updates, and feature enhancements based on user feedback and emerging needs.

**13. Data Monitoring and Analysis :**

* Continuously monitor user behaviour, preferences, and system performance.
* Use analytics tools to derive insights for future improvements and strategic decision-making.

# 

# CHAPTER 2: SYSTEM ANALYSIS AND DESIGN.

## 2.1 SYSTEM ANALYSIS

### 2.1.1 Introduction to system analysis and design.

System Analysis and Design (SAD) is a crucial phase in the development of information systems that involves understanding, defining, and designing a solution to meet specific business needs. It is a structured, problem-solving approach that aims to improve the efficiency and effectiveness of a business or organisation by designing and implementing computer-based information systems.

**Key concepts**

⦁ **System**: In the context of SAD, a system is a set of interrelated components working together to achieve a common goal. It could be a manual or automated system.

⦁ **System analysis**: This is the process of studying a system to understand its components, their interactions, and how they fulfil organisational objectives. The goal is to identify the problems and opportunities within the existing system.

⦁ **System design**: Building on the findings of system analysis, system design involves creating a blueprint for the new system. This includes defining the structure, components, modules, interfaces, and data for a system to satisfy specified requirements.

⦁ **Requirements analysis**: Identifying and documenting the specific needs or improvements required by the business. This phase sets the foundation for the development process.

⦁ **Feasibility study**: Before committing resources to a project, a feasibility study is conducted to assess the practicality and viability of implementing the proposed system.

⦁ **Prototyping**: Creating a simplified model of the system to visualise and test key features before the full-scale implementation. Prototyping helps in gathering feedback and refining requirements.

⦁ **Data modelling**: Representing the data involved in the system and the relationships between data entities. Techniques such as Entity-Relationship Diagrams (ERD) are commonly used.

⦁ **System implementation**: Putting the design into action by coding, testing, installing, and supporting the new system. It involves converting data, training users, and ensuring a smooth transition.

⦁ **System maintenance**: After implementation, the system requires ongoing support and maintenance. This includes fixing bugs, making updates, and adapting the system to changing business requirements.

⦁ **User involvemen**t: Throughout the SAD process, user involvement is critical. Understanding user needs, expectations, and feedback helps in creating a system that aligns with the actual requirements and ensures user acceptance.

### 2.1.2 **System actors(internal and external)**

**INTERNAL SYSTEM ACTORS**

**Administrator:** Manages the overall system; manages user accounts, updates menu items, oversees orders, and monitors system performance.

**Chef/Cook:** Responsible for preparing and updating the menu, marks items as available/unavailable, and manages inventory.

**Waitstaff:** Interacts directly with customers by taking and managing orders, communicating with the kitchen, and updating order status.

**Manager:** Oversees day-to-day operations, monitors overall restaurant performance, handles customer complaints, and ensures smooth operation.

**EXTERNAL SYSTEM ACTORS.**

**Customer:** Interacts with the system as a user,

browses the menu, places orders, and provides feedback.

**Supplier:** Provides the restaurant with necessary ingredients, manages inventory, receives and processes orders from the restaurant.

**Visitor/Guest:** Accesses information without registering, browses general information about the restaurant, such as the menu and opening hours.

**Delivery Service:** Handles the delivery of orders, receives order details, delivers orders to customers, and updates order status.

### **2.1.2 Functional Requirements Of Project.**

**LOGIN FUNCTIONALITIES**

| **Features** | **Functionality** |
| --- | --- |
| **User authentication** | Create a user authentication system to securely store and manage user credentials. |
| **Login form** | Design a login form with fields for the username/email and password |
| **Server- side validation** | Validate the entered credentials against the stored user data in the server's database. |
| **Session management** | Create a session for the authenticated user after successful login. |
| **Security measures** | Implement measures like account lockout after multiple failed logins attempts to prevent brute-force attacks. |
| **Password recovery** | Implement a password recovery mechanism, such as sending a reset link to the user's email. |
| **User logout** | Provide a logout option that destroys the user session and redirects to a logged-out state. |

**REGISTRATION FUNCTIONALITIES**

| **Features** | **Functionality** |
| --- | --- |
| **User registration form** | Design a registration form with fields for required information (e.g., username, email, password). |
| **Client- side validation** | Provide feedback to users for any validation errors. |
| **Server- side validation** | Handle errors gracefully and provide meaningful feedback to users. |
| **User authentication database** | Use strong encryption algorithms for password hashing. |
| **Email verification** | Send a verification email with a unique link that users must click to activate their accounts. |

**LOGOUT FUNCTIONALITIES.**

| **Features** | **Functionality** |
| --- | --- |
| **Logout button/ link** | Create a logout button or link, typically placed in the navigation bar or user account section. |
| **Logout endpoint** | Set up a server endpoint or route (e.g., "/logout") to handle logout requests. |
| **Clear session data** | Destroy or invalidate the user's session on the server side to revoke authentication. |
| **Provide confirmation** | Display a confirmation message on the redirected page or in a modal dialog. |

### 

### 

### 2.1.3 Non-functional Requirements

| **Features** | **Description** |
| --- | --- |
| **Performance** | Define the maximum allowable time for the system to respond to a user request. |
| **Reliability** | Specify the percentage of time the system should be available for use. |
| **Security** | Define requirements for user authentication processes. |
| **Usability** | Define principles for the user interface design, ensuring it is intuitive and user-friendly. |
| **Scalability** | Specify how the system should scale by adding more machines or nodes. |
| **Maintainability** | Specify coding standards and practices for easy maintenance. |
| **Interoperability** | Define how the system should integrate with other external systems. |
| **Capacity** | Define the maximum amount of data the system can store. |
| **Compatibility** | Specify which browsers the system should be compatible with. |

# 

# **Chapter 3: Introduction**

## 3.1 Tools and technology.

The various tools used in developing this system are:

**PHP (Hypertext Preprocessor):**

Purpose: Server-side scripting language used for web development.

Function: Executes on the server, generating dynamic content and interacting with databases.

Common Use: Creating server-side logic, handling forms, interacting with databases.

**MySQL:**

Purpose: Relational database management system (RDBMS).

Function: Stores and manages data in a structured manner.

Common Use: Storing and retrieving data for web applications.

**HTML (Hypertext Markup Language):**

Purpose: Standard markup language for creating web pages.

Function: Defines the structure and content of a web page.

Common Use: Creating the basic structure and content of web pages.

**CSS (Cascading Style Sheets):**

Purpose: Style sheet language used for describing the presentation of a document written in HTML.

Function: Controls the layout and appearance of web pages.

Common Use: Styling and formatting the visual presentation of web pages.

**JavaScript:**

Purpose: High-level, interpreted programming language for making web pages interactive.

Function: Runs on the client side, enabling dynamic content and interactivity.

Common Use: Adding behaviors, validating forms, creating interactive features.

**Visual Studio Code:**

Purpose: Source code editor developed by Microsoft.

Function: Supports various programming languages, offers features like debugging, syntax highlighting, and extensions.

Common Use: Writing, editing, and managing source code for web development.

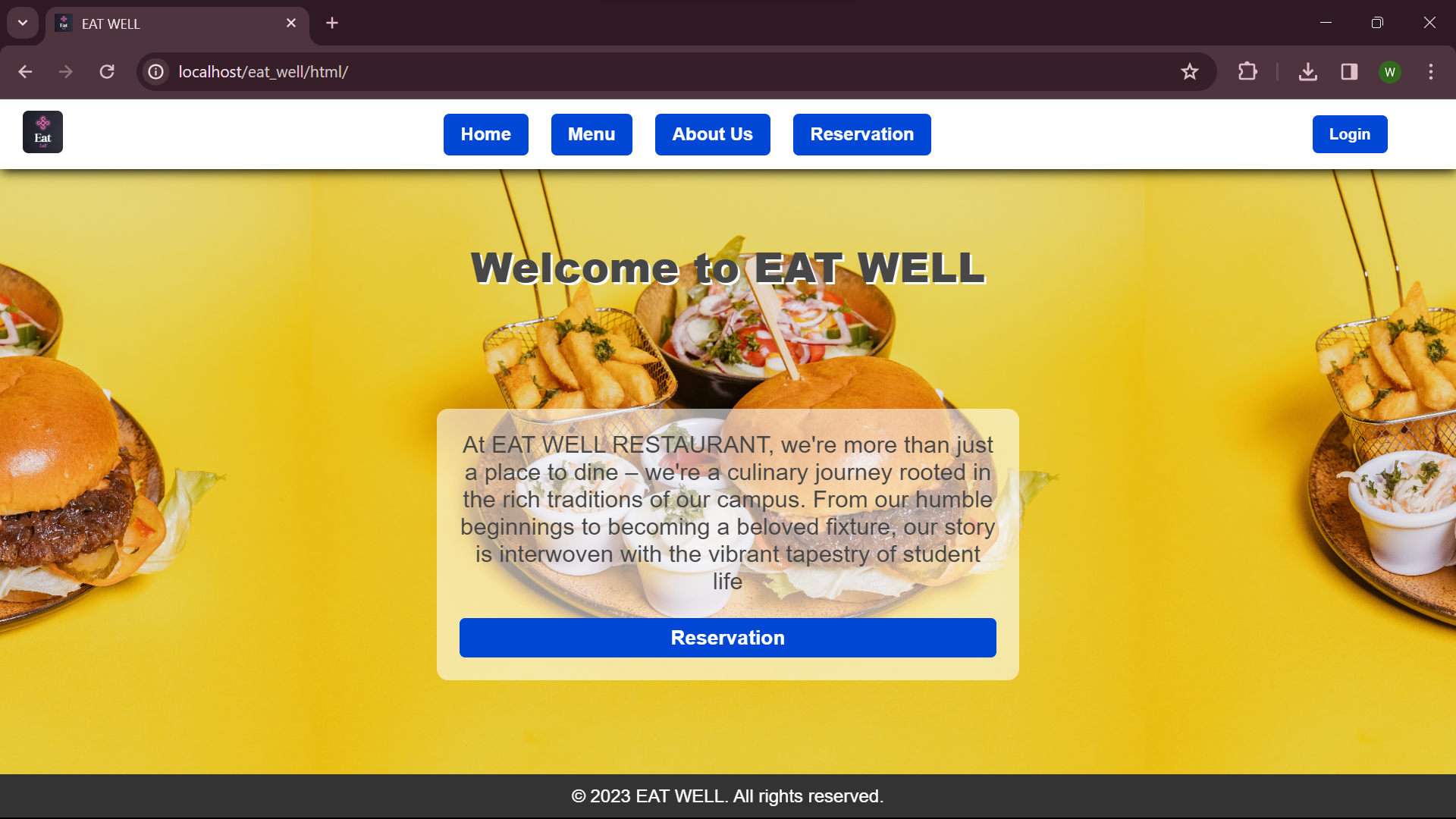
## 3.2. Screenshots

Interfaces are grouped based on the user's classification:

1. **CLIENT.**

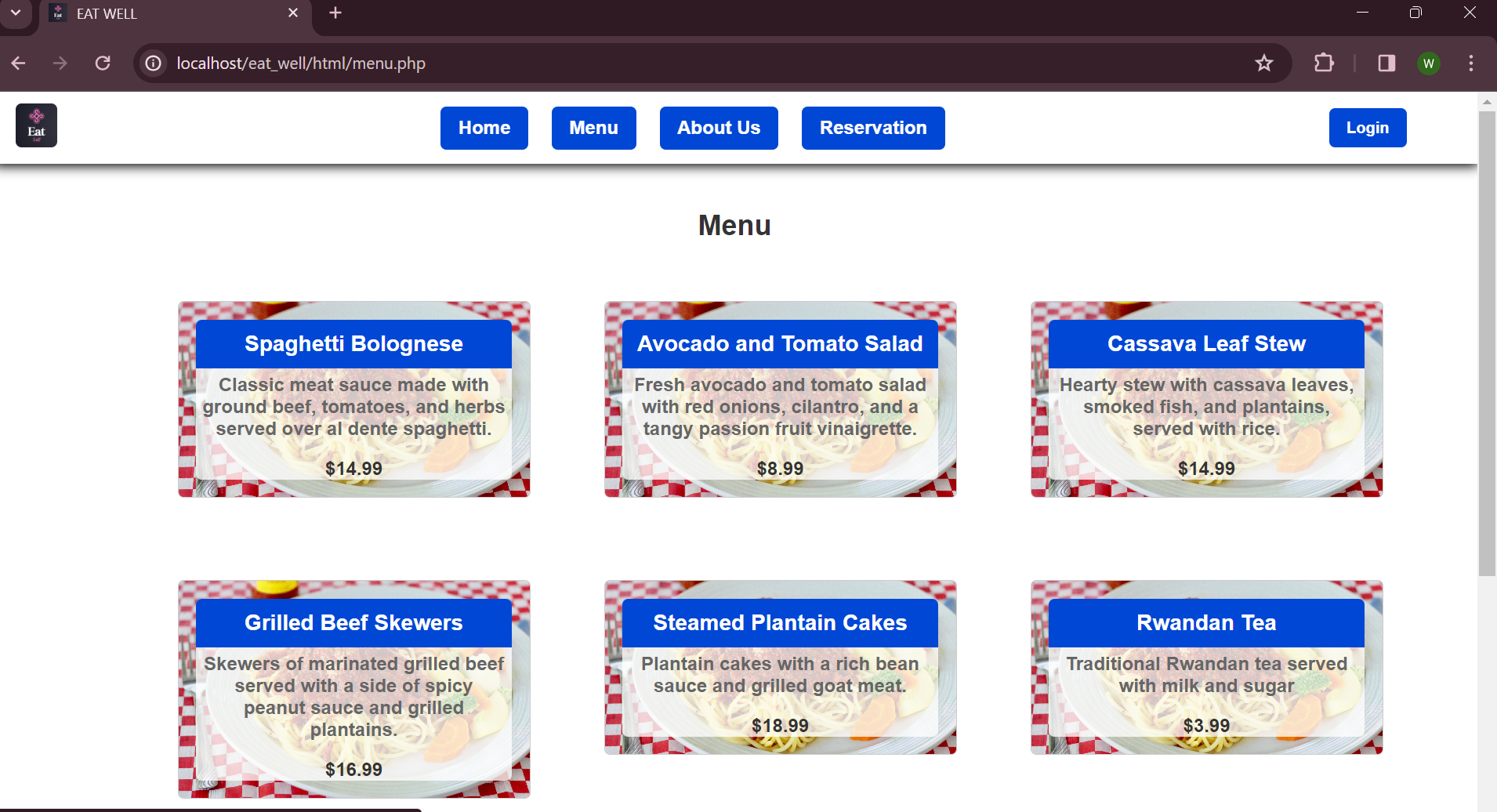
**Home page.**

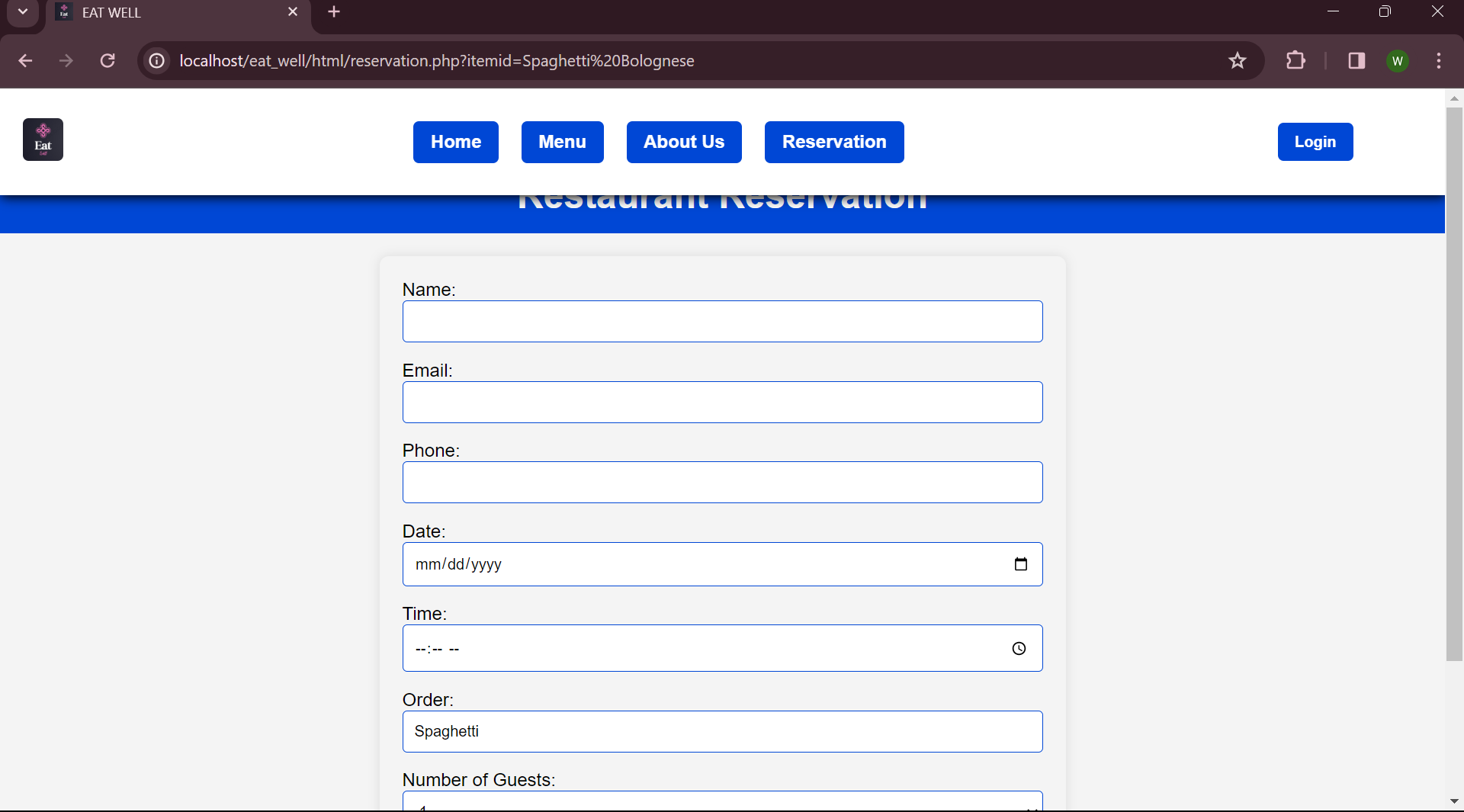
Eat well homepage features a sleek top navigation bar, prominently displaying the brand logo alongside essential options like home, menu, about us, reservation and login. This user-friendly design ensures easy access to key sections, allowing clients to efficiently navigate the site and engage with the restaurant's offerings.



**Menu.**

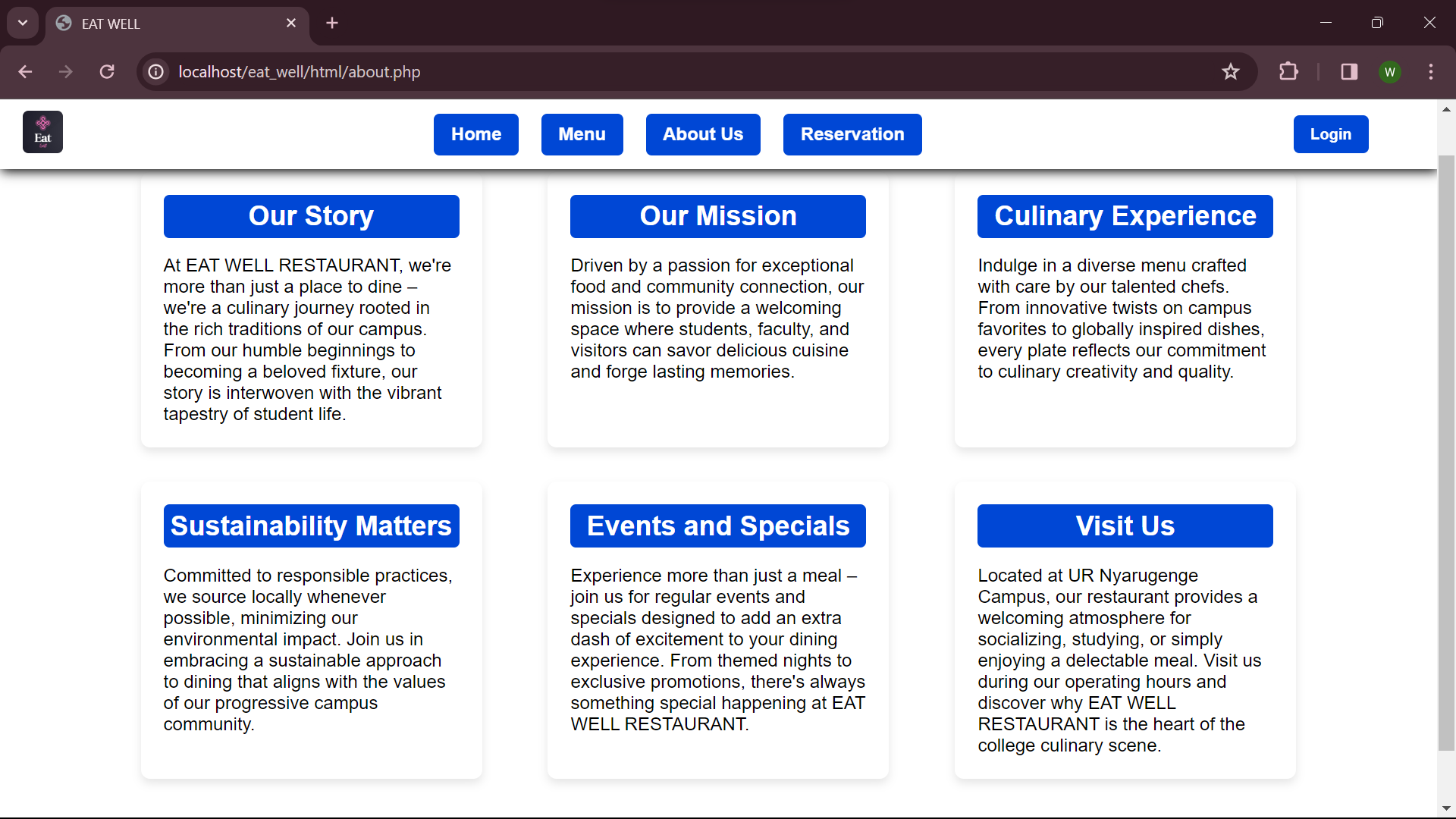
The menu offers a spacious layout with an organised list for easy browsing that goes along with favourable pricing and description which makes it easy to browse and select items based on their preferences and budget and make reservation, ensuring a user-friendly experience . Notably, all products are accessible without the need for logging in, ensuring a hassle-free experience for every visitor.





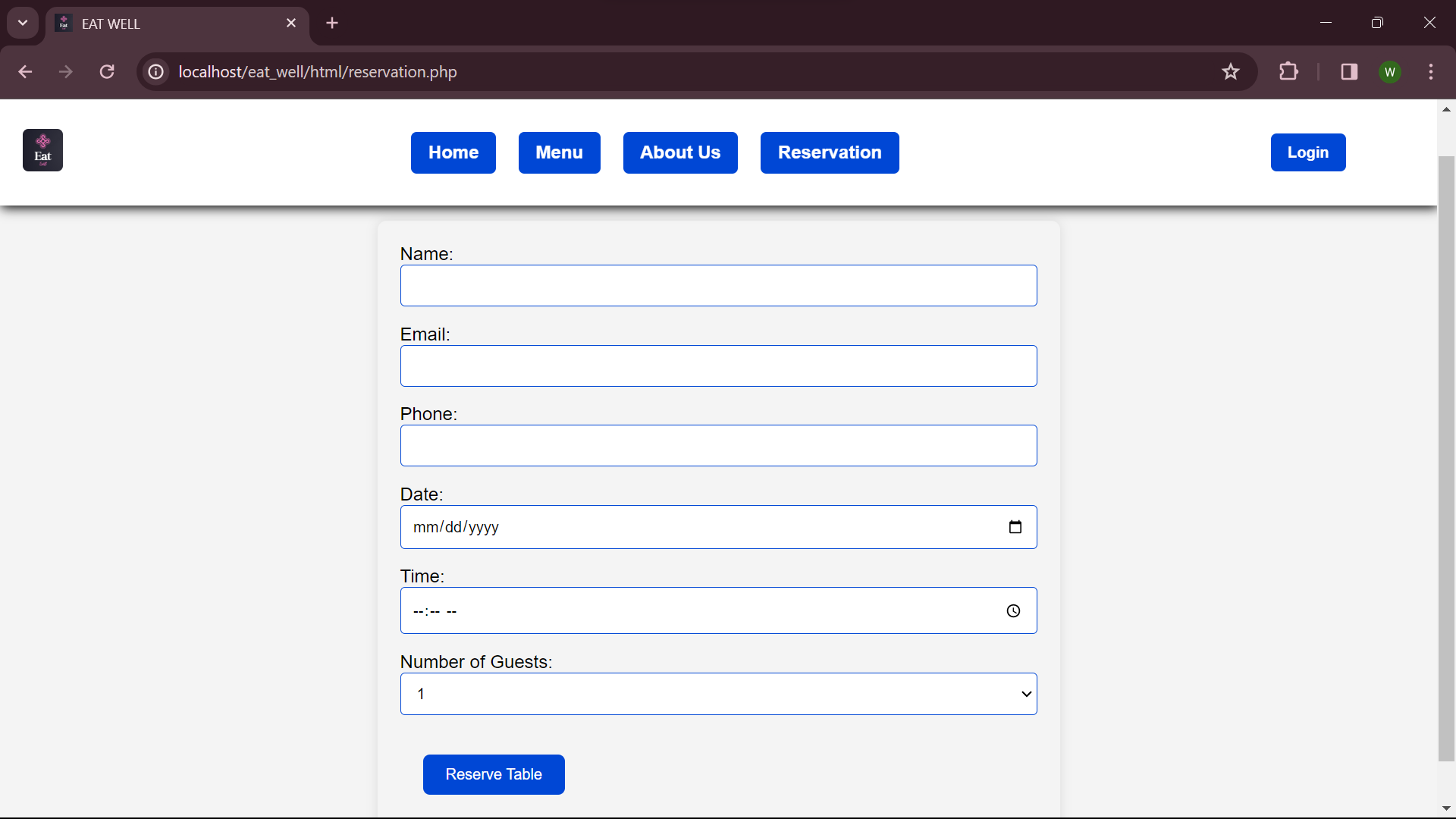
**About us.**

About us gives a clear notion on Eat well restaurant’s story, mission, culinary experience, sustainability matters, events and specials and where to find us.This content is created by system administrator.



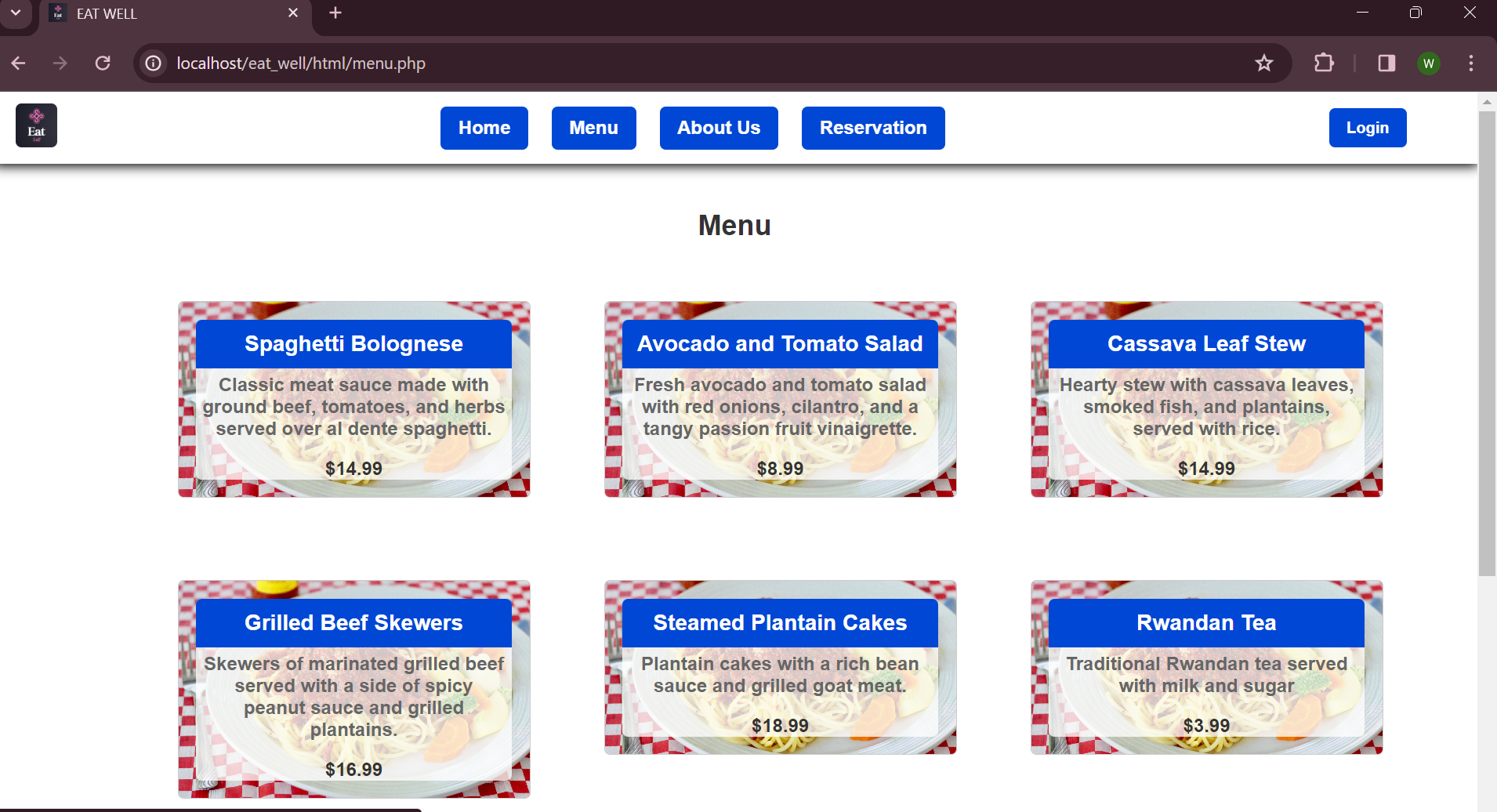
**Reservation.**

This helps a client make a reservation of his/her choice from the list of products on the menu and specify the number of people to seat on the table. But while using this button to make a reservation, unfortunately you don’t specify your choice .

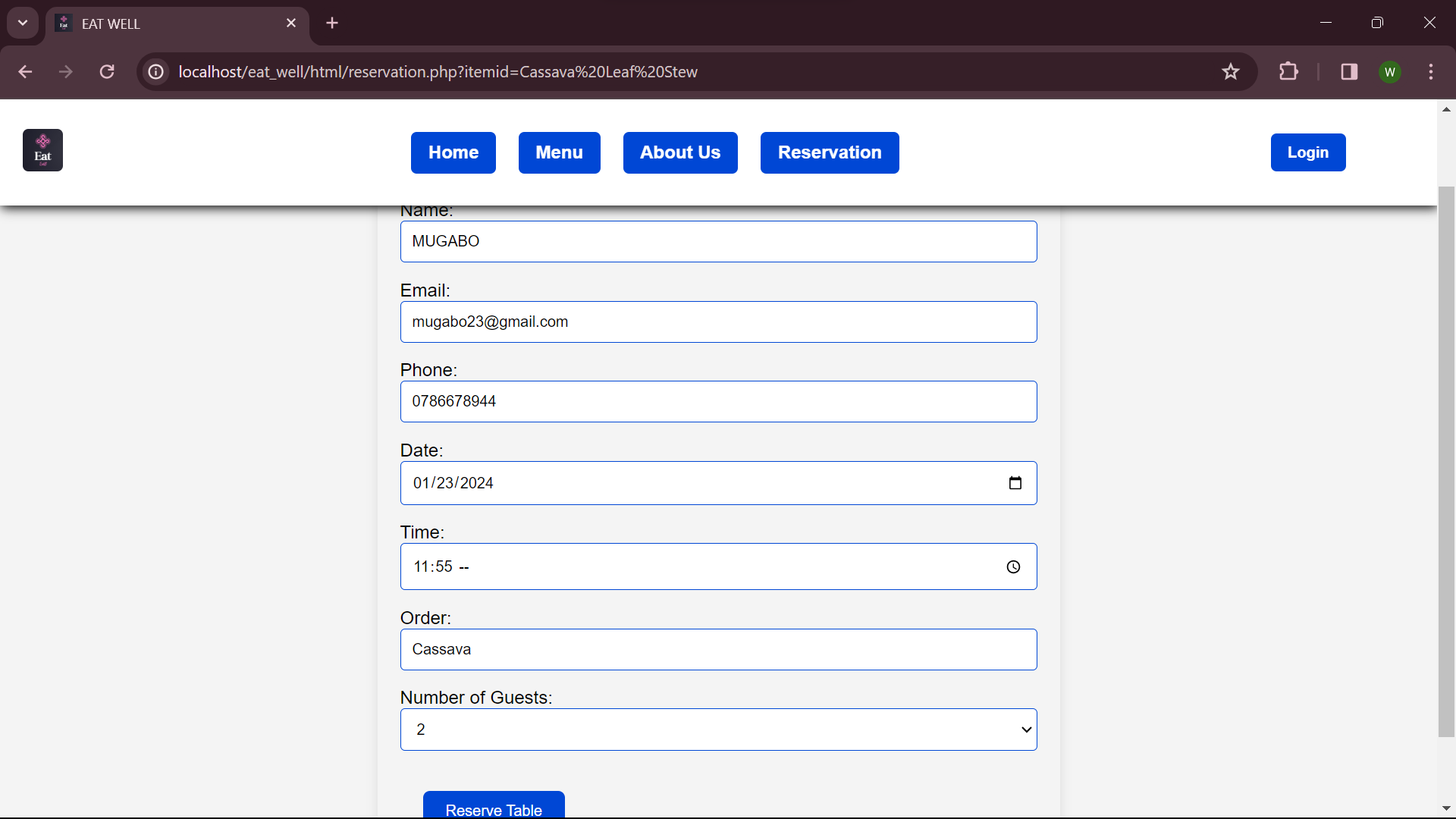


**Reservation process.**

To make a reservation, start by visiting the menu.

.

Then, click on the desired order to direct you to the reservation page.

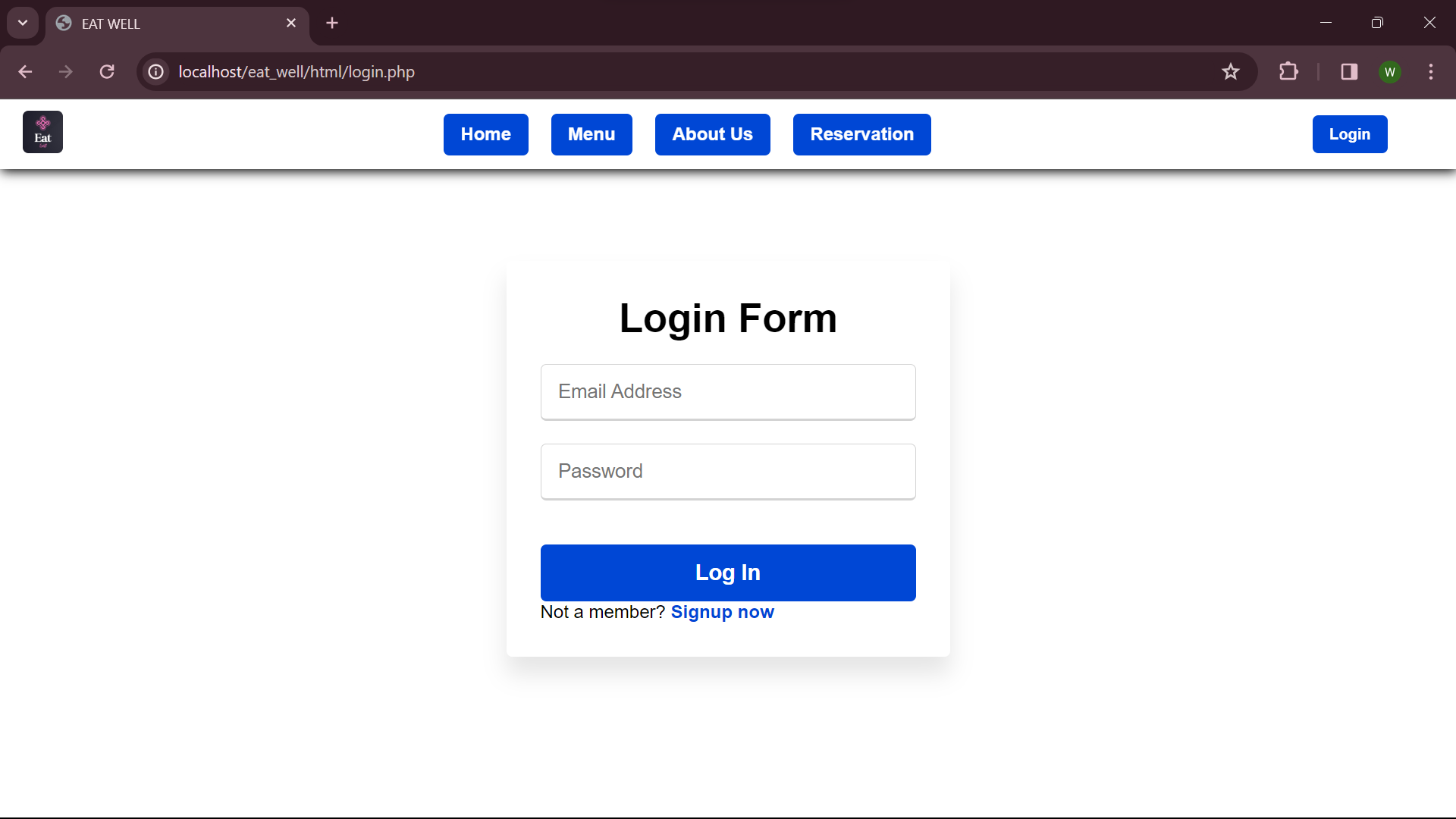


And finally your reservation is done successfully!!



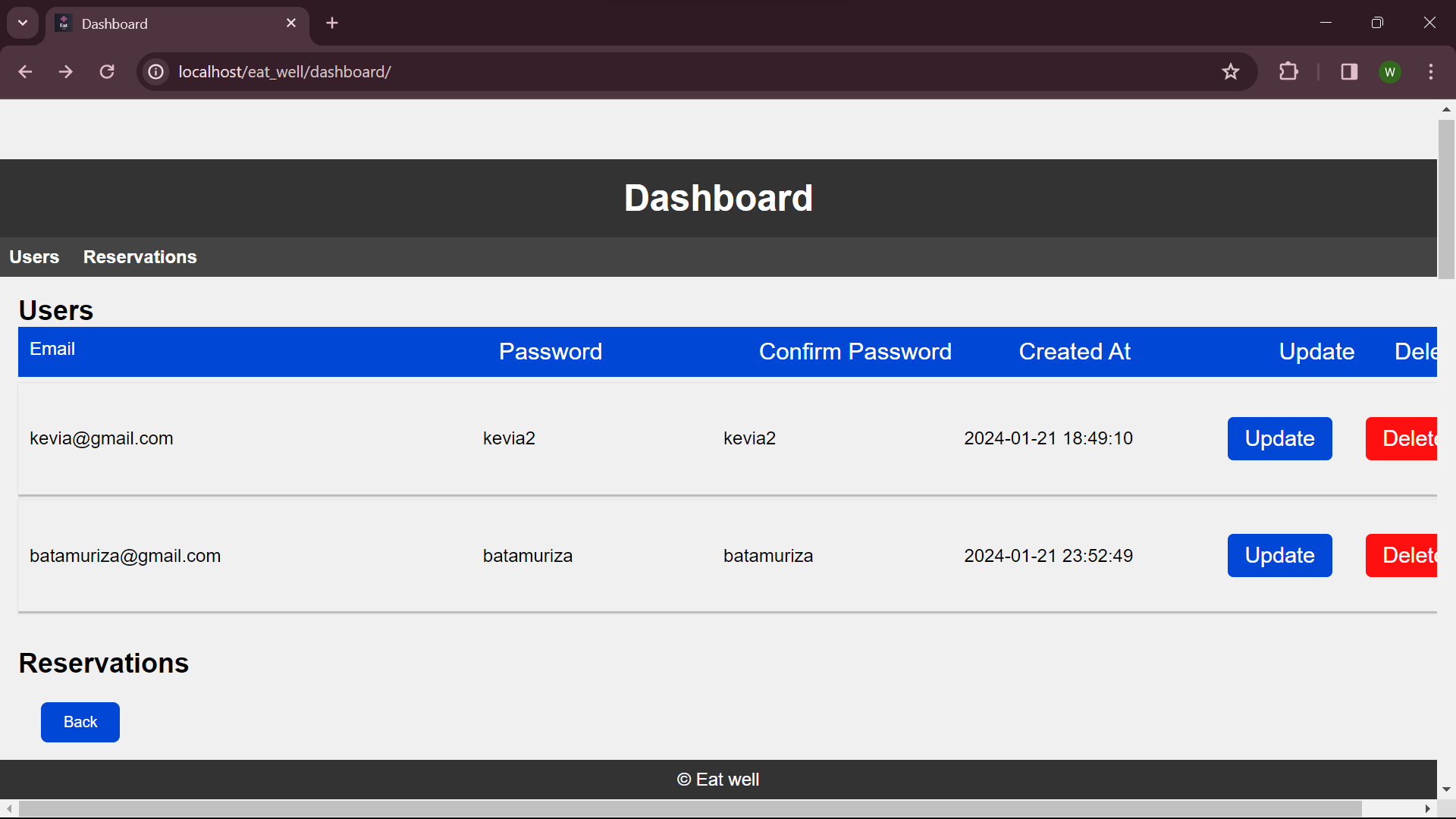
2. **System admin.**

To access the admin panel, login is required first. The figure below describes a login page.



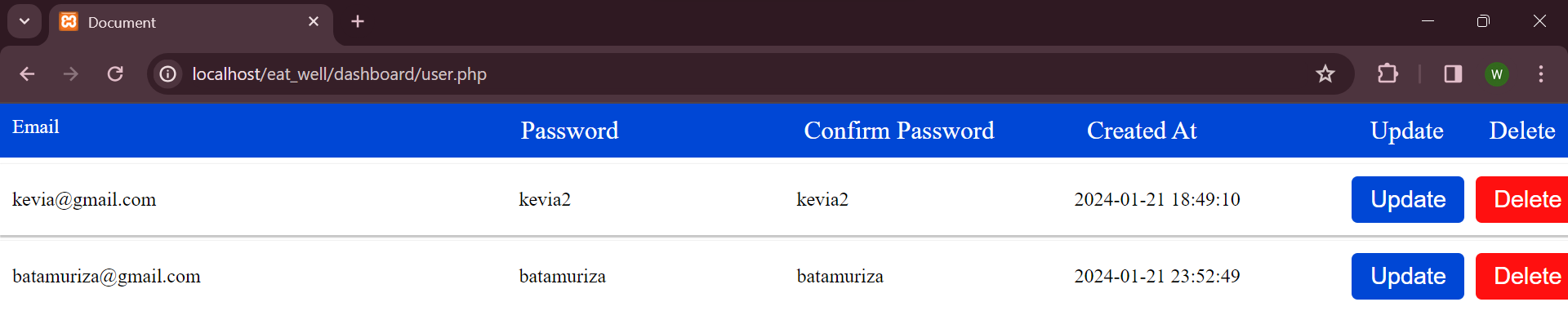
When the form is filled and the information match these in the database, the redirect is done to dashboard page.

**Admin’s dashboard.**

****

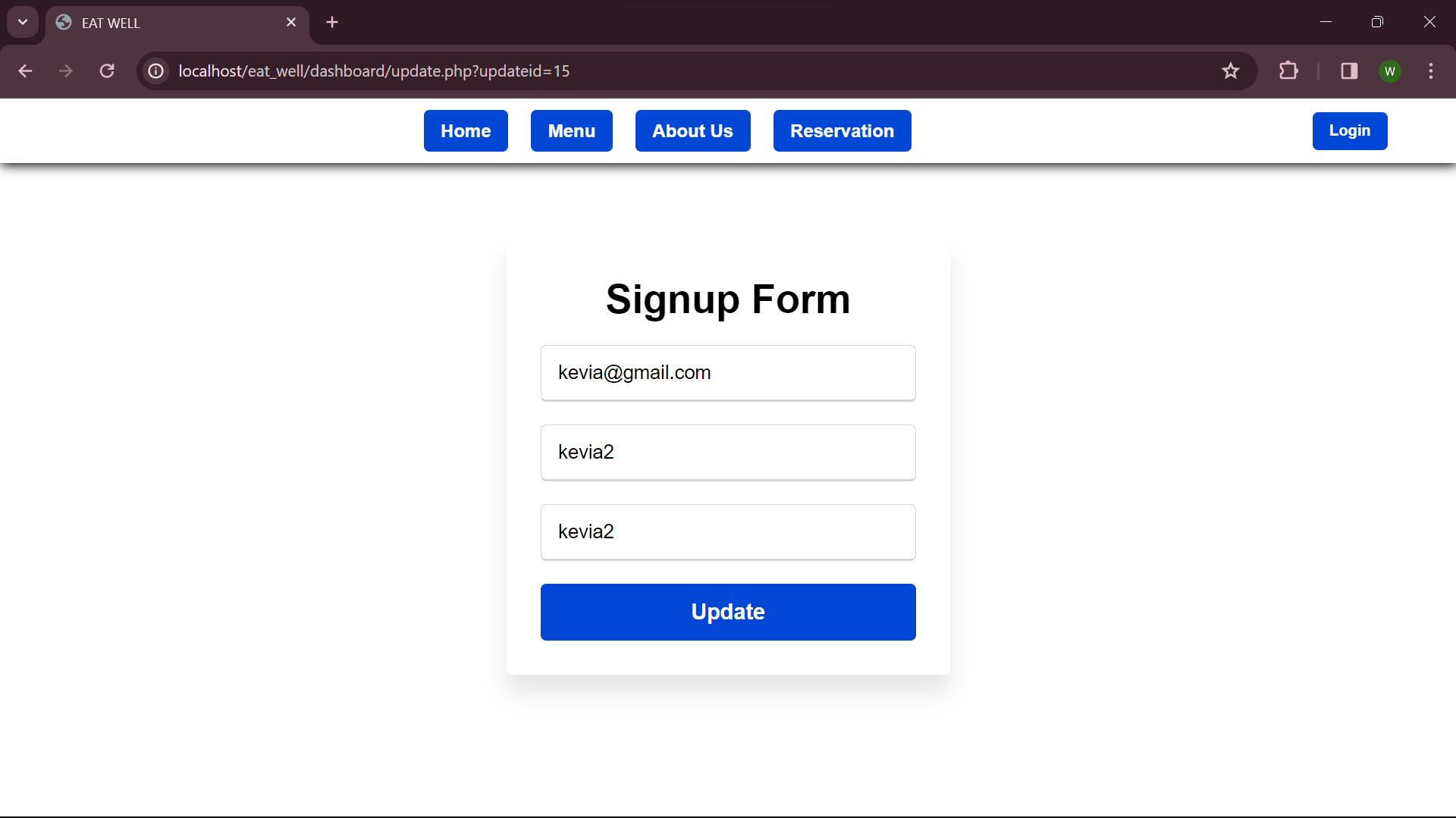
1. **Users.**

This shows persons who made login as admin.



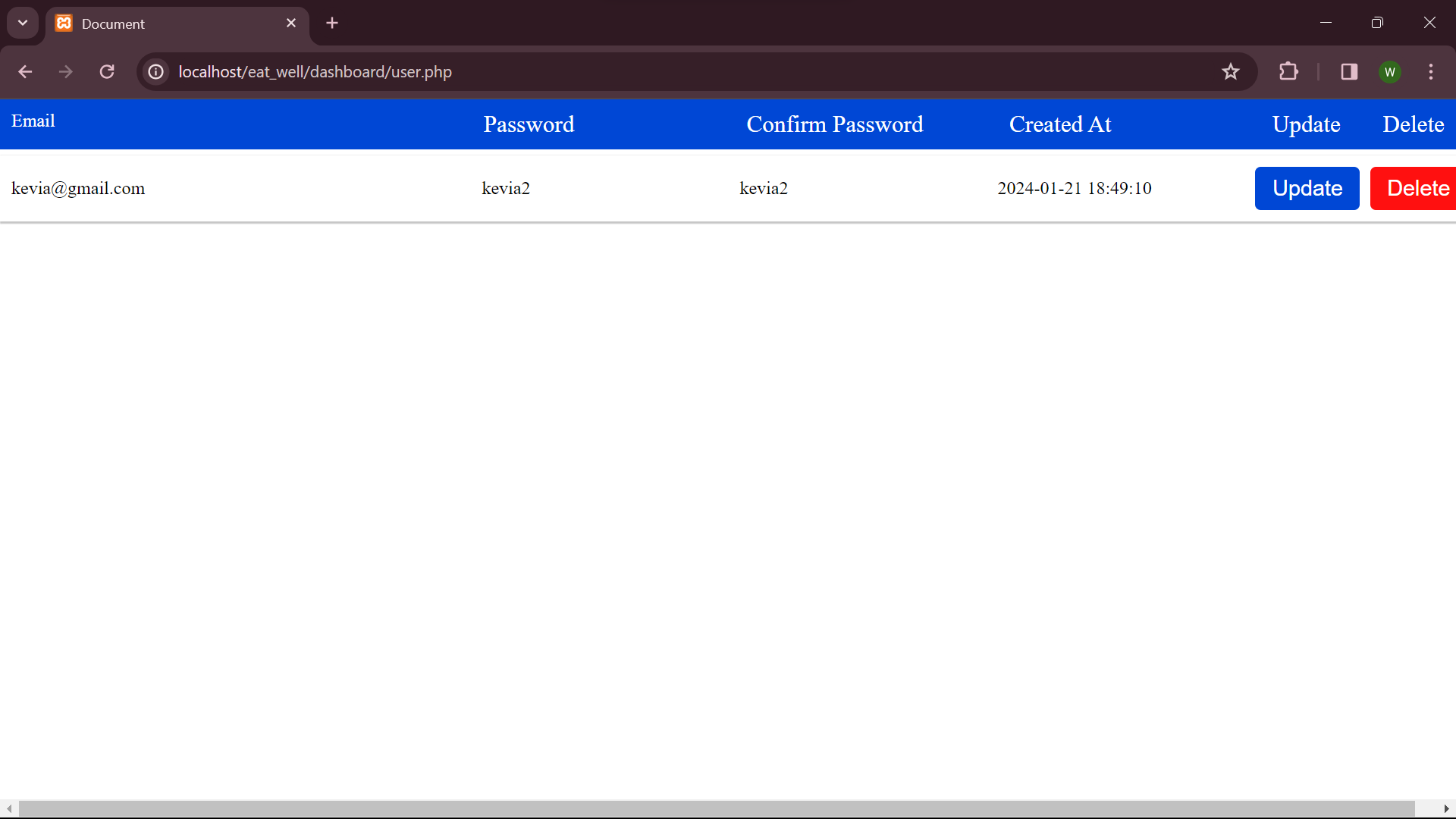
1. **update.**

This enables the admin to make changes on his/her credentials; either email or password.



**b**. **Delete.**

Delete button allows admin to remove one of the admins from the dashboard.



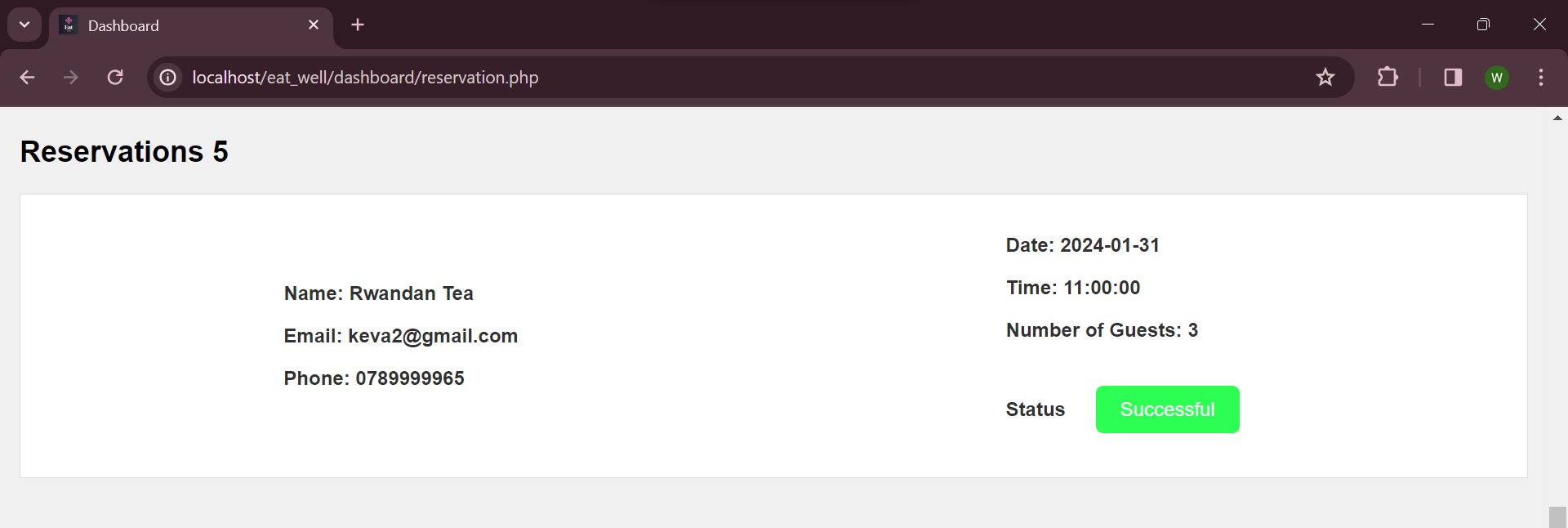
As from this picture, the admin with the email of batamuriza was deleted.

**2. Reservations.**

This shows the admin the list of clients who placed their orders.



After a client is served, the admin approves this by clicking the “served” button which changes the status of the reservation from “pending” to “successful”.



As seen in the above picture,the involved client was approved to have been served.

# Appendix.

* GitHub link: <https://github.com/Keva24/Eat-well.git>
* Username:Keva24
* Password:kevia@123