People's Democratic Republic Of Algeria Ministry Of Higher Education
University August 20th, 1955- skikda Faculty Of Science Department Of Computer Science

THESIS

Digitization of Hotel Systems in Algeria

Author:
BOUMELHA Nour El Houda

Supervisors:
Mr.LAROUM Toufik

Submitted on
July 2, 2025

2024 - 2025

Abstract

Digitization of hotel systems is being explored in Algeria to improve accessibility and efficiency, and the aim of this research is to reduce bureaucracy, simplify services and facilitate access to hotels for citizens and visitors. The study reduces the field visits needed to obtain information by utilising digital means (thus saving time). The paper highlights the benefits , challenges and opportunities of digital transformation in hotel management and is part of a master's thesis.

Résumé

La numérisation des systèmes hôteliers est à l'étude en Algérie pour améliorer l'accessibilité et l'efficacité, et le but de cette recherche est de réduire la bureaucratie, de simplifier les services et de faciliter l'accès aux hôtels pour les citoyens et les visiteurs. L'étude réduit les visites sur le terrain nécessaires pour obtenir des informations en utilisant des moyens numériques (gain de temps). L'article met en évidence les avantages, les défis et les opportunités de la transformation numérique dans la gestion hôtelière et fait partie d'un mémoire de master.

Acknowledgements

First of all, I want to say biggest thanks to Allah whose guidance, strength and support have been my most important resource while preparing this Master Thesis. This work would not have been possible without His blessings.

Mr. LAROUM Toufik is greatly thanked for his dedicated help, patience and ongoing support in the guidance of this research. I have benefitted from his guidance and encouragement in the execution of this work. I thank all the faculty members, colleagues and my academic family in the university as well for their help, guidance and support during my academic journey on ways to better myself.

A huge thanks to my ever loving parents and sisters for the never giving up, encouragement I needed. Their faith in me propelled me to stand firm and finish my work. Please may Allah give them good health, joy and protection from every harm.

Contents

List of Figures

0.1 General Introduction

Technology has bridged and made many sections interactive via communication, information management and service effectiveness in the digital age today. This is true for the hospitality industry as well, increasing hotel management, simplifying operations and enhancing guest's experiences largely depend on digitization. Hotels can deliver faster, faster and cost-effective services through digital solutions and less paperwork than paper-based operations.

This dissertation, accomplished within a master's degree in information Systems is carried out with the aim of dealing with the topic that consists of what is the digitalization of hotels systems in Algeria. The purpose is to provide simple access to hotel offerings and services for the tourist and citizen through effective data retrieval without the need for physical visits.

Providing digital solutions, this work intends to expedite time management, enhance the quality of service and move towards the modernization of hospitality.

This thesis is organized as follows towards the objective of this thesis:

- Chapter 01 introduces the daily life and applications of web applications in different industries
- Chapter 2 illustrates the analysis and design of hotel management system using UML diagrams
- This third chapter describes the digital application development process, the end product, and functions.

Results of this research are expected to increase hotel management in digitation advantages and obstacles, prospects of development sectoral hospitality in Algeria.

Chapter 1

Study of the existing

1.1 Introduction

The World Wide Web has permeated our everyday life altering the way we learn things, interact and run a business. This chapter digs into different dimensions of the web, the relevance of web in contemporary times. Furthermore, we shall drill a little deeper on the digitization of hotel business and then convey how web has spoilt hotel operations as well guest experience.

1.2 Computer Networks

1.2.1 Network

Computer Network is a system that connects more than one independent computers and devices so that they can share data and resources. It simplifies user to user communication by connecting computers together via either wired (cable) or wireless. Both hardware and software equipment work the network. It usually runs in client-server model, client will request servers to utilizing those resources. Networks come in different sizes, ranging from a basic between two computers up to large scale systems which need other devices like routers, switches.

[1, 2]

1.2.2 Categorize

- Controller Area Network (CAN): personal network for systems electronics (buses), especially used in the automotive sector;
- Local Area Network (LAN);
- Metropolitan Area Network (MAN): Metropolitan Network;
- Wide Area Network (WAN);
- Personal Area Network (PAN).

[1, 3]

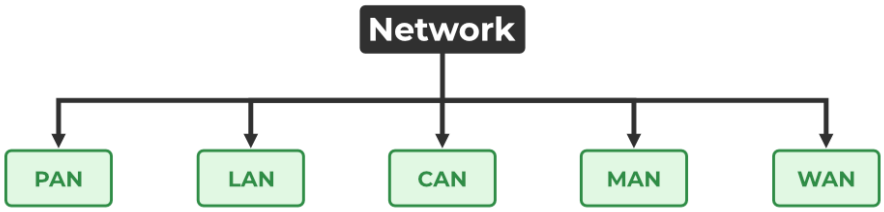

FIGURE 1.1: Types of Computer Networks

1.3 Internet

Internet the global system of interconnected computer networks that use Internet Protocol suite for communication with each other. Connects local areas to statewide networks by using various digital, wireless and networking technologies Internet Proffers many services like forums, databases , email , hypertext which facilitate World wide internet use and intermingling of information. [4]

1.3.1 Internet Protocols

Internet protocols are the rules used for the purpose of connecting computers and devices to the Internet. This keeps the data correctly transmitted between systems. Protocols do different things some are for file transfer, others are for e-mail or data security etc. Familiarizing one with these protocols becomes instrumental in enabling seamless internet experience.

- The "TCP" protocol:(Transmission Control Protocol/ Internet Protocol) consists of a set of protocols: IP that gives each computer unique address and assigns TCP to manage the exchange of data packets that will be correctly sent/received in the right order. A Connection-oriented protocol that provides reliable service;

- The "SMTP" protocol: (Simple Mail Transfer Protocol) to send and push outgoing mails into queue with email ID of the recipient in header It is delivered there, will then take it off the outgoing mail list and can be of various message types (text, video, or images) ;

- The "FTP" protocol: (File Transfer Protocol). FTP is used to transfer files between systems, client-server model for doing so. This connects, (machine to machine) using ID and password for both hosts and files will be transferred;

- The "HTTP" protocol: (HyperText Transport Protocol).HTTP is used for the communication and transfer of hypertexts across the internet; it tells how information is formatted and sent;
- The "HTTPS" protocol (with S for secured, or "secured") HTTPS is an extension of HTTP to secure communication using SSL/TLS encryption and ethical functionality (originally introduced in 1999). If a website deals with sensitive information (credit card numbers) a SSL certificate is a must, and users should confirm that a site is serving over HTTPS before submitting any personal data;
- The "UDP" protocol: (User Datagram Protocol) is a connectionless unreliable transport protocol that sends packets which may not be in order or delivery attempted. UDP is better than TCP, and is to be used in real time applications i.e. Video Streaming, Online gaming and VoIPs where low latency matters and some percentage of loss is not a big deal;

[5]

1.4 The Web

The World Wide Web (WWW), also known as the Web or W3, is a system of interconnected public webpages accessible through the Internet. There is a big difference between Web and Internet but the Web is only one application on top of Internet Tim Berners-Lee proposed the design of World Wide Web, he wrote first web server, web browser and webpage on his computer at physics research lab of CERN 1990 .[6, 7]

1.4.1 How The Web Works

The Client-Server works which is used to run various websites using browsers on Internet. It gives the basic level of knowledge about web technology.

Client : In virtual world, a "client" is computer (or host) that asks and receives service or data from server.

Server : a Server is the computer where the data or information is stored, the clients request it when they need.

In short, Request is sent by the Client and if available the Server provides requested information, or service from its SQL database.[8]

Advantages :

- Centralized Management : Simplifies data administration and security protocols ;
- Scalability : Allows independent upgrading of servers and clients to meet evolving demands ;

Disadvantages :

- Security threats : Both Clients can be infected by malware and servers running the rest of your infrastructure could get DoS ;
- The risk of data transmission vulnerabilities: the possibility for data packets to be sniffed or tampered ;

[8]

1.4.2 Definition Of Website

Website is a set of web pages that are connected together and have some kind of domain name shared across interconnected webpages on the Internet. The majority of these pages are written in HTML (HyperText Markup Language) containing Text, Images, Videos and other types of multimedia. Websites are served on web server which can be accessed through Chrome or Firefox or Safari etc. [9]

1.4.3 Definition Of Webpage

A webpage is an existing web site, in other words a single HTML file viewable on World Wide Web using web browser and scripted with Hypertext Markup Language (HTML). They have a unique Uniform Resource Locator (URL) which enables users to access it independently, one page per post. [10]

1.4.4 Difference Between Webpage And Website

Webpage

- A single document available through a web browser ;
- Usually composed in HTML (HyperText Markup Language) ;
- May contain text, images, videos, and other multimedia ;
- Designated by a special URL (Uniform Resource Locator) ;
- Example: A story posted on a news website ;

Website

- A collection of linked web pages grouped together under a common domain. name ;
- The content is published utilizing the internet by being placed on a web server ;
- Typically has a homepage that provides links to other pages within the site. The whole news website is composed of diverse articles, segments, and multimedia content ; [11]

1.4.5 Types of Web Pages

The fundamental building blocks of the internet, web pages are what give content and features to users across the world. Static and dynamic web pages are the two major classes that they can be sorted.

1. Static Web Pages

Static web pages are built out of code that never changes, mainly HTML and CSS, so always the same content for any visitor. Every page is baked on a server and served to the client with zero modification. Simplicity means faster load times and better security because there are just fewer moving parts that someone can hack. And Updates on the Content need to be done manually, If you want to modify anything in the code.

2. Dynamic Web Pages

In contrast, Dynamic web pages are those pages which show the content according to user activity or on other parameter. They employ server-side programming like PHP, ASP.NET or Node.js fetches information from database and generates personalized content. This is applicable for businesses, where the data is updated quite frequently or user interface is required to be used like booking systems, user profiles, e-commerce etc. [12] [13]

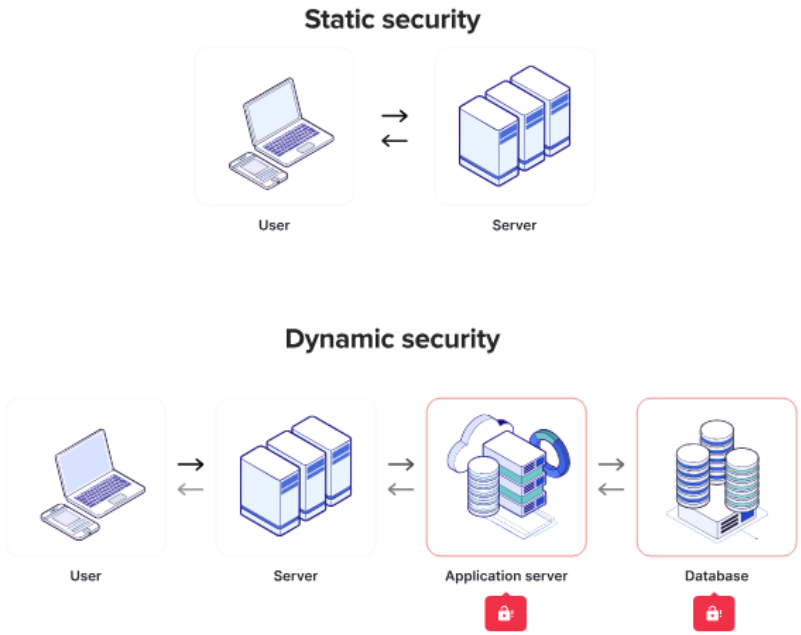

FIGURE 1.2: Static And Dynamic Web Pages Request

1.5 Conclusion

To conclude this chapter, we looked at the basics of computer networks, internet, world wide web. We created some basics, what sites and pages are not the same, and understanding the difference between static vs dynamic

content And these principles lay the foundation of what you need to know to understand the digital terrain and how it is changing. It builds well as a foundation to evaluate contemporary web technologies. Next chapter will cover the conception phase where these theories are used to generate any novel solution.

Chapter 2

Analysis and Conception

2.1 Introduction

Building a website requires a structured methodology of analysis and design to organize ideas, document requirements, and prevent issues during development. In this chapter, we will explore the modeling approach that underpins our design process, specifically employing the Unified Modeling Language (UML) to systematically represent and plan the website's architecture.

2.2 Why Modeling Matters in System Development

Modeling is crucial in system development to comprehend the function of the system in dealing with complexity and maintenance. It is universal, which stakeholders communicate through, getting everyone to think the same about requirement. Creating prototypes or models helps teams to see the essence of main features, having tasks to work on and collaborate better, so it makes the development process more streamlined and organized. [14]

2.3 what's Sahla DZ

In this era of digital, the hotel bookings need to be managed properly for everyone (travelers and owner of Hotel) Sahla DZ is a hotel system manager which want to reduce overall time and efforts in searching and compare booking of a room. Being an echo to platforms gisted around like Trivago, it provides a smooth and logical experience where select what you want because you got just the balance level of deals and services catered to your requirement.

Some features of Sahla DZ :

- Search and compare hotel prices on different booking platforms (like Trivago) ;
- Show extra hotel details like room amenity, rating and user review ;

- Filter search results by price, location (centre), star rating and what you can do ;
- Direct bookings through integrated partner websites reservation, cancellations and modifications of hotel reservations;
- Get personalized suggestions with user preferences;
- Open an account to save your favorite hotels and follow your bookings.

All These features are instantly accessible thereby offering a simplified and hassle-free hotel booking process

2.4 Uml

Unified Modeling Language (UML) A visual language in Software engineering to model structure and behavior of the system. It offers graphical notations for teams to specify, visualize and document software components that can help in communication and design. UML is non-methodology and can be used in different development process to align people together with system as a common target. [15]

2.4.1 Characteristics Of UML

The UML has the following features:

- It is a generalized modeling language;
- Maintained by OMG (the Organization for the Advancement of Simulation);
- Used for software, business and system modeling;
- Agnostic- Works with all the methodologies;
- UML saves money.

[15]

2.4.2 Advantages Of UML

- Standardized notation for consistent communication;
- Facilitates visualization of system architecture;
- Enhances collaboration among stakeholders;
- Aids in early detection of potential issues;
- Improves planning and design efficiency;
- It reduces costs and time to market.

[16]

2.4.3 Disadvantages Of UML

- Time-consuming to create and maintain diagrams;
- Steep learning curve for beginners;
- Potential for ambiguity in system requirements;
- Complex diagrams can be hard to interpret;
- May not capture dynamic aspects of systems effectivel.

[16]

2.4.4 Uml Views

Unified Modeling Language (UML) uses various perspectives to capture diverse stances of a software system viewed. Basically :

1. View from user: It shows system functionalities in end-user view, what the system do but not internally how it works.

2. Structural View – representing the static aspects of system, like classes, objects and their relationships that show how the system is constructed.

3. Dynamic view (Behavioral view): Shows the dynamic of the system, which are the interactions among instances/object and their change over time, how the system reacts to different input.

4. Implementation View: Displays the modular structure and how software components are related, It helps in knowing how the system is divided into parts.

5. Deployment View: How the system components are deployed on top of hardware environments describes where in the physical world software elements are installed.

This set of perspectives combined into make a holistic view on the system to be able for a better design and stakeholder communication. [17]

2.4.5 Types Of UML Diagrams

Unified Modeling Language (UML) offers a variety of diagrams to represent different aspects of software systems. These diagrams are categorized into two main types: [18]

- Structural UML diagrams :
  - Class diagram;
  - Package diagram;

  - Object diagram;
  - Component diagram;
  - Composite structure diagram;
  - Deployment diagram.

- Behavioral UML diagrams :
  - Activity diagram
  - Sequence diagram;
  - Use case diagram;
  - State diagram;
  - Communication diagram;
  - Interaction overview diagram;
  - Timing diagram. [19]

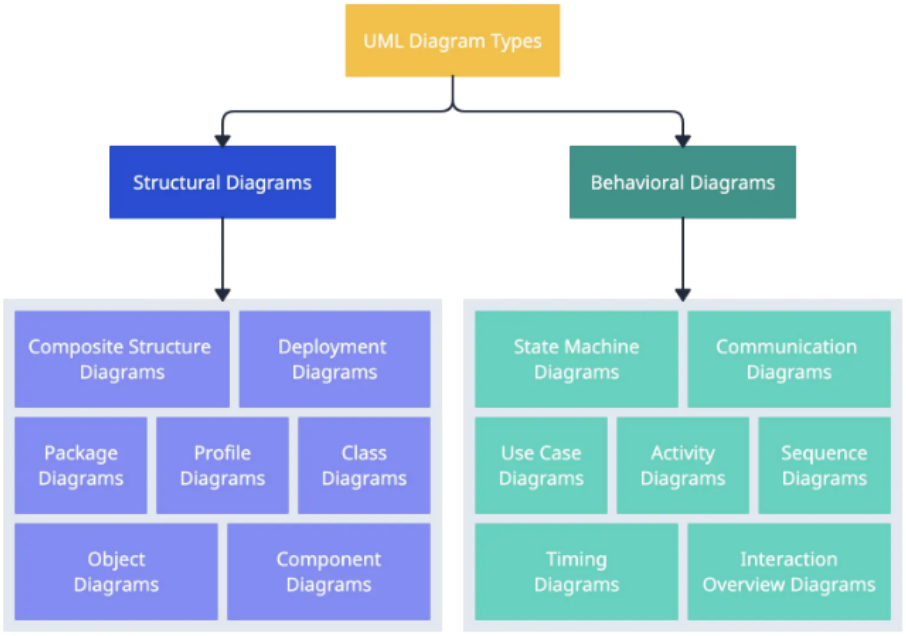

FIGURE 2.1: diagrams categorized

2.4.6 The definitions Of The diagrams Used

- What Is A Use Case Diagram ?

  Use Case Diagram High level visual representation of user(actors) and system interactions that is used to illustrate the functional requirements of a component or the whole Application. This gives you an overview

of how other users interact with features to help with consumption and structuring of system requirements. [18]

- What Is A Class Diagram?

  A Class Diagram is a Structural Diagram kind of the Unified Modeling Language (UML), which emphasizes the static structure of a system's representation by displaying classes, attributes, methods and relationships between objects. It is a high level architecture of the system, showing how classes are connected to each other which helps the understanding as well designing the structure of the system. [20]

- What Is A Sequence Diagram?

  A Sequence Diagram is one of the Unified Modeling Language (UML) diagrams that depicts objects and the flow of control between them in a specific scenario of system showing messages that occurs over time among these objects. [21]

2.5 Conception

In a project context, conception refers to the initial stage of planning and designing a project. It involves identifying the project goals, objectives, scope, and requirements, as well as determining the resources, timeline, and budget needed for the project.

During the conception phase, project stakeholders, such as the project sponsor, project manager, and team members, collaborate to develop a comprehensive project plan that outlines the project's purpose, deliverables, and strategies for execution.

The conception phase is critical because it sets the foundation for the entire project, and any missteps or oversights during this stage can have significant impacts on the project's success. Therefore, it is essential to ensure that all relevant information is gathered and considered during the conception phase to ensure a solid project plan that can be executed effectively.

2.5.1 Use Case Diagram

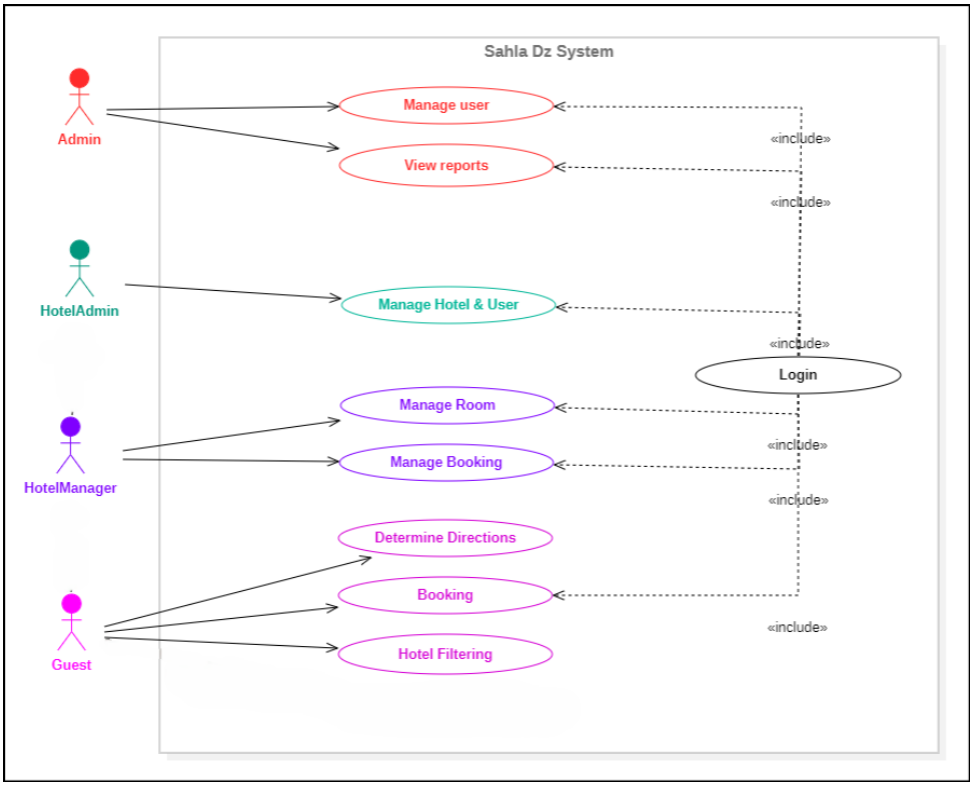

FIGURE 2.2: Sahla DZ use case Diagram

2.5.2 Class Diagram

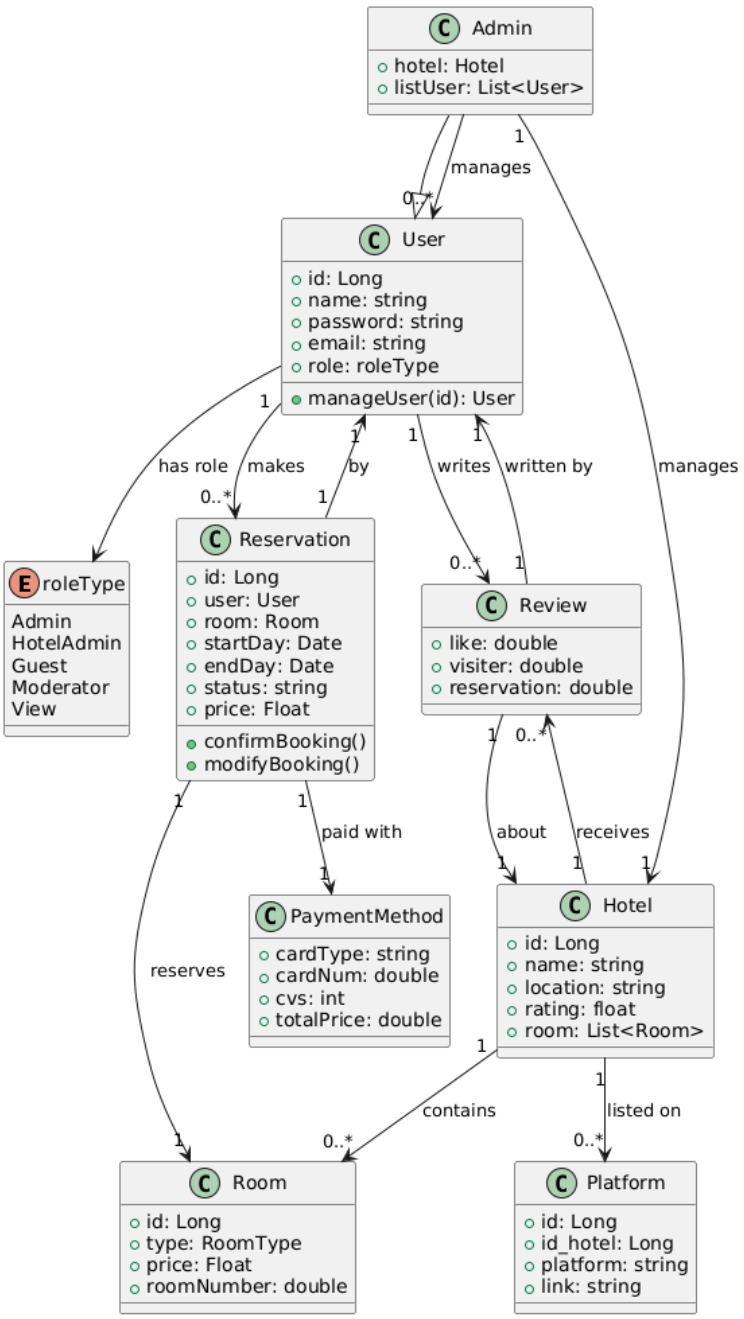

FIGURE 2.3: website class Diagram

2.5.3 Sequence Diagram

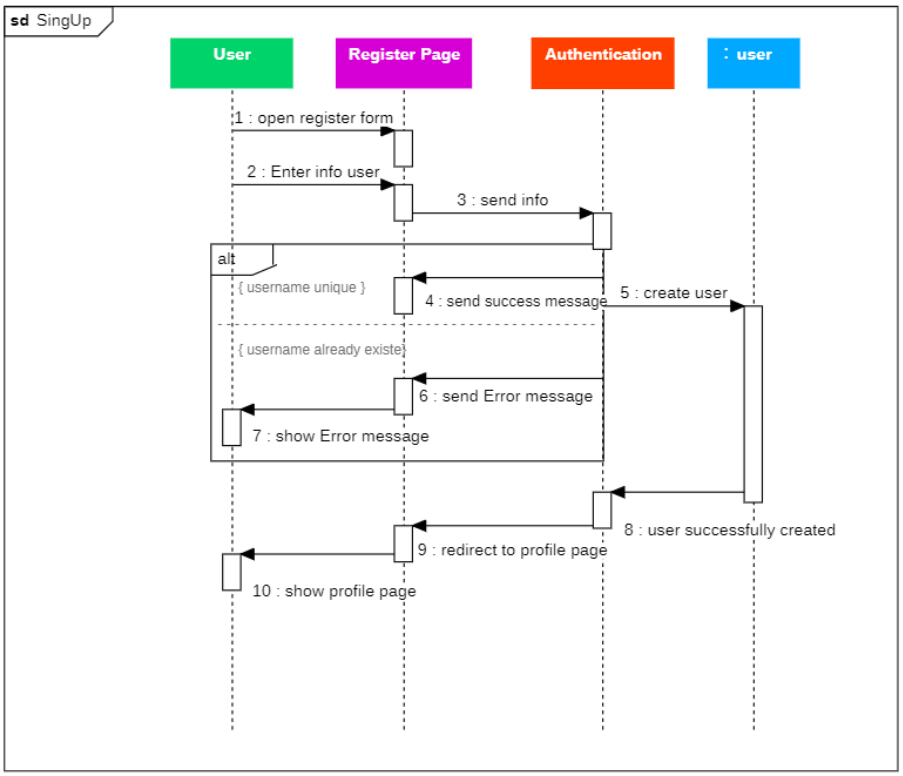

FIGURE 2.4: website sequence Diagram sing up

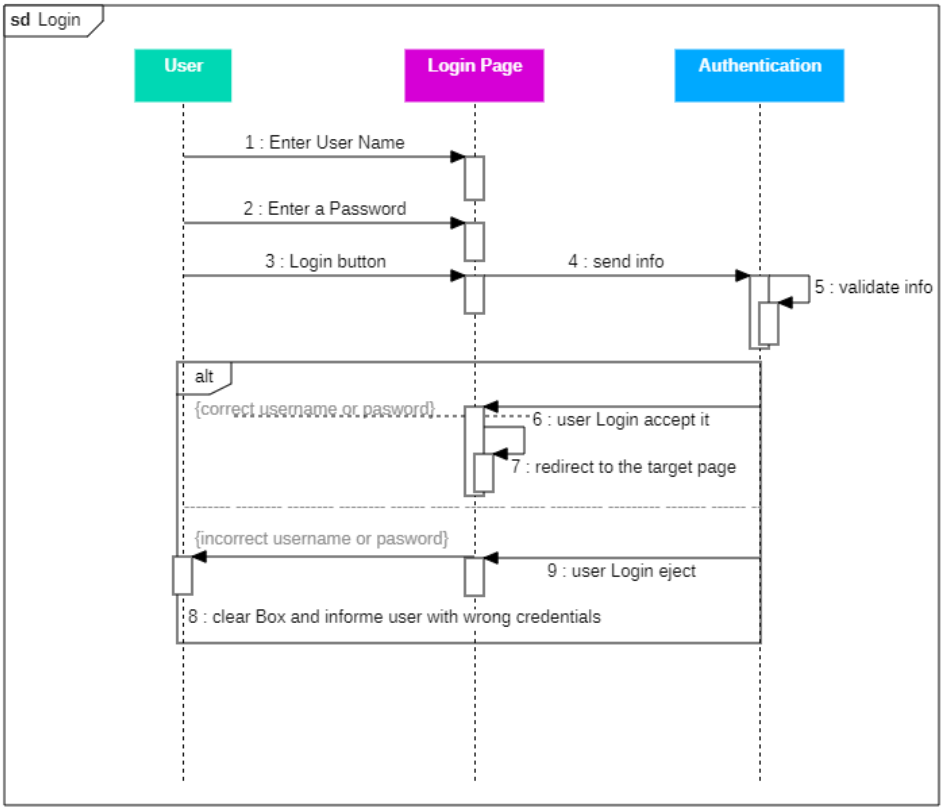

FIGURE 2.5: website sequence Diagram log in

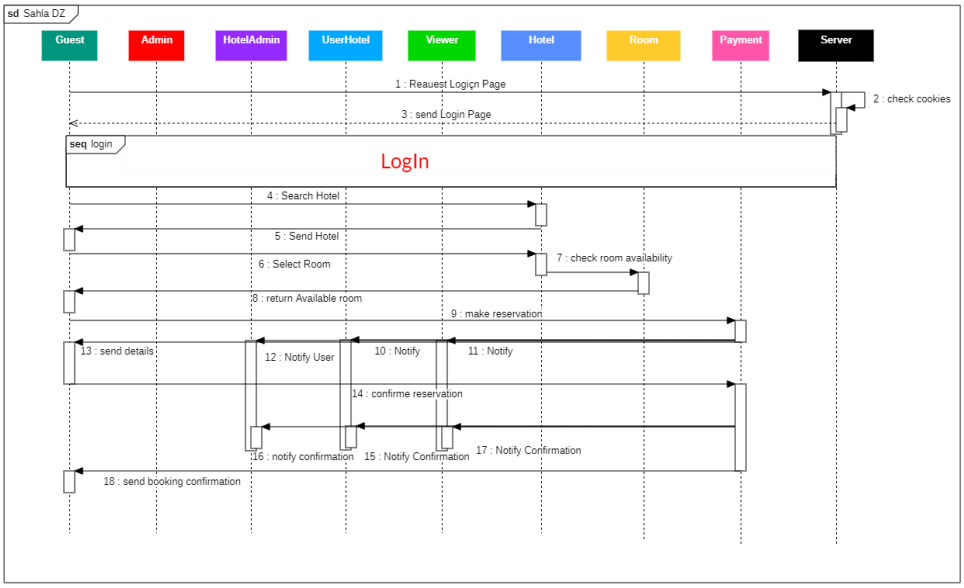

FIGURE 2.6: website sequence Diagram interaction

2.6 Conceptual Data Model

Conceptual Data Model (CDM) represents a sensible high level abstraction of entities, attributes and relationship in a system Attention to business needs rather than technicalities, aiming for a unified understanding among customers. The CDM is not technology bound and is a stepping stone to various levels of designs

Commonly (Entity-Relationship Diagrams ERDs) or UML diagrams for visualization of the relationships between the data.

It allows for specifying key entities (e.g Hotel, Room, Reservation) and their associations in order to logically structure the data.

CDM is important for synchronizing business requirements with system design, to keep data manageability in check and to further facilitate smooth implementation later.

[22]

2.6.1 What Is An ER Diagram?

A ER Diagram (ER Diagram/ Entity- Relationship Diagram) represent entities, their attributes in database design and relationships visually. (rectangle) entities and e.g. Customer, Order; ovals(attribute) diamond or line for relationships.

Identity of entities (e.g. Customer, Order) are rectangular, attributes ovals and relationships are diamonds or line labelled.

These help with data modeling, which is important to have a clean and structured design in place before you code anything. They are used to bridge the communication gap between the technical and non-technical stakeholders as well they are documentation for system operability.

Introducing ER : ER Diagrams, A Key to Clear Architecture provides Design Redundancies clear understanding of data model and necessary to give databases business backing

[23] [24]

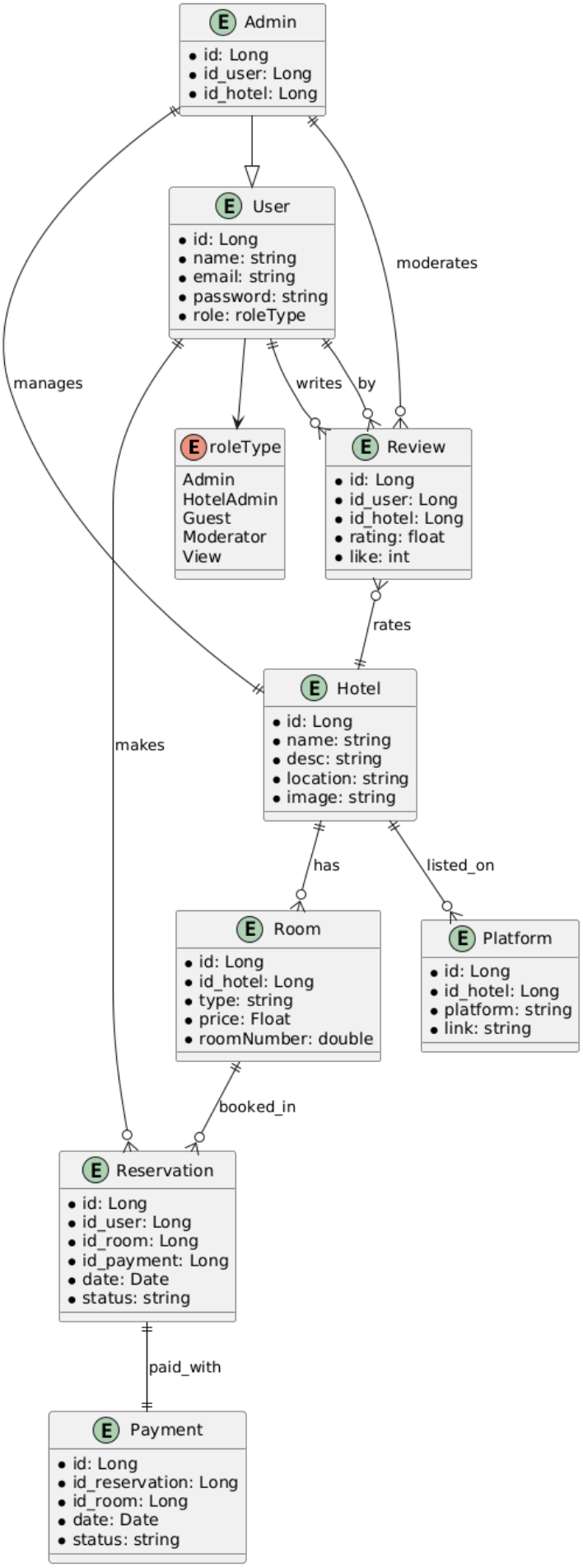

FIGURE 2.7: website ER diagram

2.6.2 The relational model

Organizing data into tables (relations), that are relations organized in rows (tuples) and columns (attributes) to allow for both data integrity, and flexibility with database management The way it organizes these relations, keys (primary and foreign) for relationships and querying and manipulative operations such as selection, projection and joins The structured paradigm allows it be useful, reproducible and scalable in the modern database systems. [25]

2.6.3 Passing Rule

- User (id_user, name, email, password, roleType).
- Admin (id_user, id_hotel).
- Hotel (id_hotel, name ,location , raiting , image).
- room (id_room,id_hotel, type , price ,roomNumber).
- review (id_review,id_hotel , id_user, like).
- reservation (id_reservation, id_user, id_room,id_payment, date , status).
- Payment (id_reservation,id_room, type, cardNum, cvs, totalCard,date).
- platform (id_platform,id_hotel , platform, link).
- RoleType (Enumeration) (Admin, HotelAdmin, Guest, Moderator, View).

2.7 Conclusion

In this chapter we discussed analysis/Conception of Sahla DZ analytical approach Modeling is used as a backbone in system development. UML (Unified Modeling Language) diagrams (use case, class and sequence) to organize system interactions,flows. The Conceptual Data Model was also walked through ER diagrams and relational , to give the best database design We employed these methodologies to create an hotel booking system which is scalable and optimized and prepare the platform for the next level system implementation.

Chapter 3

Implementation

3.1 Introduction

In this chapter the implementation of our hotel digitalization system is shown we describe the approach and technologies used for development. This chapter will discuss the project architecture, tools and frameworks employed for its development as well as essential system capabilities. Further we will discuss the difficulties faced in development and how it was resolved. Ultimately, last is to discuss the result of implementation and success in completing those indicative objectives of the project.

3.2 the developement environment

3.2.1 hardwares

- Processor : Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50 GHz;
- RAM : 8 GB;
- Hard drive : 128 GB;
- Operating System: Windows 10 Pro version 21H2;

3.2.2 software tools and technologies

- HTML (Hypertext Markup Language);
- CSS (Cascading Style Sheets);
- React.js (JavaScript);
- spring (java framework) + spring boot;
- MySQL (Relational Database Management System);
- Tomcat (Web Server Software);
- Browser (Web Browser);

- VS Code (Code Editor);
- Google Font (Online Font Library);

3.3 Project Architecture

Our hotel digitization system is based on a client-server architecture, through which centralized resources and services are achieved to improve control, security(since it is able to isolate the servers and databases used), as well as scalability. Whereas, the Central hub or the server handles storage of data, business logic and processes in this model; clients (hotel staff and guest using network to request services from or accessing resources over).

3.3.1 Client Server Architeture advantages

- **Centralized Control:** Resources and services are centrally stored on the server, so updates as well as maintenance and security to them are easier;
- **Scalable:** Upgrade servers to better serve client increased client load, which guarantees performance as traffic increases the system;
- **more secure:** with Data on the server it can be used for strong and elaborate security implementing so you can keep sensitive information off stolen devices and from unauthorized eyes;
- **Optimized Servers** maximizing performance and reliability to efficiently provide resourceful services to multiple Clients;

3.3.2 System Components:

- **Frontend** : (Client-Side)Developed using HTML, CSS, JavaScript and React the frontend offers a easy to use and responsive UI to system users;
- **Backend(Serverside):** Developed in Java spring boot with logic, request management from client to server and handling all data transactions;
- **Database:** Data is stored in the MySQL database, making data retrieval and management fast and efficient;
- **Authentication:** JSON Web Tokens (JWT) are utilized for user authentication on how users will authenticated inside the system;

Through the adoption of client-server architecture, our hotel digitization system creates a secure, efficient structure within that allows for an effective operation and a smooth experience for both staff and guest using the platform. [26, 27]

3.4 Frontend Implementation

3.4.1 Html

FIGURE 3.1: Html Icon

HTML (HyperText Markup Language), in our hotel digitization system is the basic structury language of the web that we can build content on. Allows the construction of structured documents using markup to state explicitly the semantics of text elements like headings, paragraphs, lists, links, quotes, and so on.[28]

```
<!DOCTYPE html>
<html lang="en">
  <head> </head>
  <body>
    <h1>Hello From HTML</h1>
  </body>
</html>
```

3.4.2 Css

In the digitizastion system of our hotel, CSS is used to improve the visual presentation and headings lay-out of the web pages set at CS (Cascading Style Sheets).

CSS is the separation of format from content, to produce more flexible and sustainable code.

FIGURE 3.2: Css Icon

Writing stylesheets in css that governs HTML elements from:color, font to space and position, CSS enhances the whole user experience. [29]

```
body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
    margin: 0;
    padding: 0;
}
```

3.4.3 JavaScript

JavaScript is a see-soon crucial component

towards digitizing our hotel, that will increase interactivity and creation of great user experience.

This enables dynamic page updates, making

our web application behave in a very responsive

FIGURE 3.3: js icon

manner by updating on user actions.

E.g., JavaScript allows instant form validation that give immediate user feedback and check the quality of data gathered.

And also it allows to fetch data asynchronously (AJAX, Fetch API) for real-time updates, like live notifications or search results. [30]

```
<script>
    alert('Hello World !');
</script>
```

3.4.4 React

React in our digitized hotel system builds dynamic, responsive and instantaneous user interfaces.

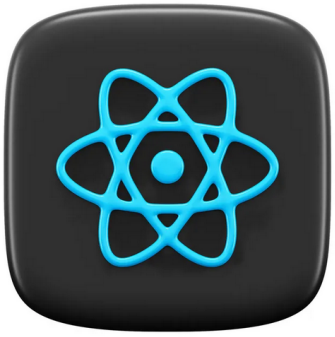

FIGURE 3.4: react icon

Component-based architecture that helps build reusable UI components leading to better maintainability and scalability

React makes it easy for the declarative design of reactive UIs, which makes state management predictable and gives an eventual predictable application.

```
import React from 'react';

const MyComp = (props) => {
    return <div>Hello, {props.name}!</div>;
};

export default MyComp;
```

All in all these capabilities helps to make a rich, guest experience oriented platform for hotel digital services.[31]

3.5 Backend Implementation

3.5.1 MySQL

FIGURE 3.5: Mysql Icon

MySQL: A highly used open source relational database management system(RDBMS) that uses Structured Query Language(SQL) to provide a way of manipulating data in the database.

```sql
-- Create a table named 'Guests'
CREATE TABLE Guests (
    GuestID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    CheckInDate DATE
);
```

A reliable, high performance and easy to use database perfectly for small to large scale applications MySQL It is multi platform, that is to say, it may be used by developers in many different programming languages. [32, 33]

3.5.2 Leaflet Integration

FIGURE 3.6: Leaflet Integration icon

Leaflet is the leading open-source JavaScript library for creating mobile-friendly, interactive maps. Despite weighing only about 42 KB, it supports essential mapping features such as tile layers, markers, popups, vector overlays, and full user interaction across desktop and mobile platforms. It's built with a focus on simplicity, performance, and extensibility via plugins, and offers a clean, easy-to-use API .[34]

3.5.3 Amadeus Api

```

const axios = require('axios');

// Replace with your access token
const accessToken = 'YOUR_ACCESS_TOKEN';

// Set the API endpoint and parameters
const endpoint ='https://test.api.amadeus.com/
    v1/security/oauth2/token';
```

FIGURE 3.7: Amadeus Api icon

Amadeus provides a range of APIs that allow travel businesses to access a rich set of connec tivity to travel content, services (flights, hotels, cars rentals etc.) through Amadeus APIs suite. Using Amadeus APIs developers can add live travel data to their applications and capabilities like flight search, hotel booking and car rental reservation.[35]

3.5.4 Java

Java(popular programming language and extensively used due to it independence from platform) But it does not stop with being popular itself, JVM.

It allows Java developers to write applications which will work on other platforms and therefore avoiding the need of reinventing the wheel. For example the hotel digitization system of our hotel Java is very robust and scalable that it handles all operations such as Reserving rooms, Guest Check-In/Check-Out and also Manage Room Availability .

FIGURE 3.8: Java icon

Object oriented nature of the language help developers create parsable and manageable code, e.g. hotel management solutions which can be efficiently and reliably maintained. [36]

```
1 public class Main {
2   public static void main(String[] args) {
3     System.out.println("Hello World");
4   }
5 }
```

3.5.5 Spring

FIGURE 3.9: spring icon

Introduction The Spring Framework is a Full

Stack, Lightweight Development Framework

for Java to build enterprise applications.

Possibly the most all-encompassing, lightweight and popular Java enterprise applications Framework for building of those application out there in market i.e Spring. It provides rich programming and configuration model to carry out the development of scalable and maintainable application quickly.

Spring Boot is an extension of Spring Framework and it helps us to create a new Spring application easily because of default configurations that come along with less manual boilerplate code for developers to write. Enables developers to build production code in a jiffy functionality-wise, and on almost negligible configuration-level. An article on Spring Boot for Beginners.

Our digitization system in the hotel is built on Spring and Spring Boot capabilities, it forms a modular maintainable architecture that lend itself well for the operational efficiencies and guest satisfaction.[37]

a- Advantages of Spring :

- Light and Modular : developers may use only what is needed and therefore applications are lighted up and maintainable.
- MVC Architecture : Application who adhered with MVC developed better supports separating of concern with the Model-View-Controller
- Security : It provides security with features such as inbuilt security (Authenticated, authorized and protection against risks: CSRF, session fixation etc)
- Less complex Database Access : Spring Data JPA, JDBC template for database interactions are easier.
- Plenty of Integration : Seamless to integrate with other third-party tools and APIs, Cloud Platforms.
- Dependency Injection (DI) : Loose coupling among components, thus makes it flexible.
- Faster development : decreases boilerplate code, development of application is very fast.

- Scalability : Works great to develop microservices or even enterprise applications. [38][39]

b- Key Spring Modules :

- The core of Spring : Inversion of Control (IoC) and Dependency Injection (DI) enables more structured
- Spring AOP : Divides cross-cutting concerns (logging and transactions) from real business logic.
- Spring MVC : Provides structured approach for building web applications based on the Model-View-Controller(MVC) pattern;
- Spring Data : Contains support for JDBC and ORM Frameworks to ease database interactions.
- Spring Security : needs for application authentication and authorization
- Spring Test : Provide test support to ensure the reliability of the application.

  These modules are built to be scalable, maintainable and secure hotel management system. [40, 41]

c - Spring Project Structure :

- **Root Package (com.example.sahladz)** The main Spring Boot application class and all the sub-packages reside in this core package.
- **Project Folder Structure**

```
sahladz/
          src/
                main/
                      java/com/example/sahlaDz/
                            controller/      # Handles API requests
                            service/         # Business logic layer
                            repository/      # Data access layer
                            model/           # Entity classes
                            config/          # Configuration Classes
    (Security, CORS, etc.)
                            exception/       # Global exception
                            util/            # Utility/helper classes
                            sahlaDZ.java     # Main application file
                      resources/
                            static/          # Frontend assets
                            templates/       # Thymeleaf templates
                            application.properties  # Config file
                test/java/com/example/hotelmanagement/   # Unit &
    integration tests
          pom.xml  # Project dependencies (Maven)
          README.md  # Documentation
```

- **Layered Explanation**

  1- Controller Layer (controller/) :

@RestController for Receive HTTP requests Example: BookingController.java, UserController.java

2- Service Layer (service/) :

Business logic (@Service) Example: BookingService.java, UserService.java

3- Repository Layer (repository/) :

@Repository for interfacing the Spring Data JPA to DB Eg: .java user.java

4- Model Layer (model/) :

Entity Classes (@Entity) Eg: Booking.java and User.java

5- Config Layer (config/) :

Includes config files such as security, database and CORS configuration For Example : SecurityConfig.java, SwaggerConfig.java

6-Exception Handling (exception/) :

Includes a Controller Advice for catching any unhandled Errors (@ControllerAdvice) Example: GlobalExceptionHandler.js

- **Additional Considerations** :

  1 - For authentication and authorization Spring Security

  (config/SecurityConfig.java)

  2 - For working with database Spring Data JPA **(repository/)**

  3 - For unit and integration testing in Spring Boot Test **(test/)**

  4 - API Documentation using swagger (config/SwaggerConfig.java).

d- Spring Security

Spring Security is a full-stack security middleware for applications which are based on java. Built on Spring Framework, provides method annotations or XML that can be used to configure the security.

1 Authentication and Authorization: Validates user identities to control their access within the application.

2 Enterprise Java Beans (EE) based, full stack with best of class Security Services providing a comprehensive solution for developing any Java EE application

3 Declarative Security Configuration: Enables developers to place security settings via annotations or XML, thereby making the process of security management easier.

5. Counter Measures Common Exploits: Provides high-level security measures like Stand Again Cross-Site Request Forgery (CSRF).

5 Supports multiple authentications mechanisms: Supports form-based authentication, LDAP authentication, basic Auth, Oauth2, JWT etc.

Using Spring Security you can easily provide an added layer of security against resource dump / unauthorized access via other security threats and improve the security in the domain of your java applications. [42, 43]

d.a - Spring Authentication :

Authentication Spring Security provides a great foundation for authentication that verifies users are who they say they are, before touching the application resources. It is associated with the following:

1. Basic Auth: Every request needs to have a username and password.

2. Form Authentication Redirects the users to a defined login page for taking credentials.

3. Token-Based [authentication]: Stateless authentication using e.g. Tokens (JWT etc) for scaleability.

4. Connects with an external identity provider using OAuth2 + OpenID connect for supporting single sign on

With this in mind, Spring Security positions itself to guarantee only the authenticated user has access to protected resources and hence helps in enhancing application security.[42, 43]

d.b - Spring Authorization :

Authorization is a method of specifying whether an user is allowed or not granted access for some of application resources. In Spring Security, role and permission is basically how this is usually handled. The main thing in this whole section is role-based access control programming, what users have and based on that users can access certain endpoints or resources.

In Spring Security you can use annotations and expressions like hasRole() to configure authorization easily. This is a way to restrict access of certain features or areas of an application using specific user with a role.

d.c - Spring Security Configuration :

Java classes can be used to configure Spring Security, usually with @Configuration and an @EnableWebSecurity annotation. This is a sample of how to configure the authorization by hasRole("Admin"):

```
1
2 @Configuration
3 @EnableWebSecurity
4 public class SecurityConfig extends WebSecurityConfigurerAdapter {
5
```

```
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
            .antMatchers("/admin/**").hasRole("ADMIN")
            // Only ADMIN role users can access URLs starting with /admin
            .antMatchers("/user/**").hasRole("USER")
            // Only USER role users can access URLs starting with /user
            .anyRequest().authenticated()
            // All other requests require authentication
            .and()
            .formLogin()
            // Enabling form login
            .permitAll();
    }
}
```

here is an example :

```
AntMatchers("/admin/**").hasRole("ADMIN")
```

the line verifies that clicks for all URL with /admin string can only be peformed by users in the ADMIN role.

```
antMatchers("/user/**").hasRole("USER")
```

will give access only to users who is having the role of USER for /user endpoints.

```
.anyRequest().authenticated()
```

means any other requests needs to be authenticated (not logged in), but without a specific role Remember that the

```
 .hasRole("ADMIN")
```

method checks if user has role from GrantedAuthority objects in security context. [44]

e- Spring Data JPA :

1- Introduction to Spring Data JPA : Spring Data JPA is a piece of larger Spring Data suite which allows developers to use various awesome persistence/relational patterns, etc in their application with ease. It is based on Java Persistence API (JPA), and this provides a repository abstraction so you do not have to bloat your code with creating the boilerplate code for database deliberations. It fits perfectly with Spring Boot applications, allowing to write CRUD operations without writing raw SQL or HQL (Hibernate Query Language) anymore. [45]

2- Benefits of Using Spring Data JPA :

- Reduces Complexity for data Access: Using Spring Data JPA, one can communicate with the database via plain repository interfaces and cut off the common data access layers hassle.

- Query Methods: Spring Data JPA provides to generate query on method name or annotations so you are no longer writing a custom query from the basic operations.
- Pagination and Sorting support: Spring Data JPA has support for paginated and sorted results, which can be very handy when it comes to dealing with large data sets.

3- Configuring Spring Data JPA for the Hotel System :

To use Spring Data JPA in your hotel system project, you need to:

- Include Dependencies : make sure Spring Data JPA and a JPA implementation (preferably Hibernate) are in your project dependencies automatically Example for pom.xml (for Maven):

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

- Configure DataSource : In application.properties or application.yml, set your database connection properties.

```
spring.datasource.url=jdbc:mysql://localhost:3306/hotel_system
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update
```

- Create Entity Classes : Define your JPA entities that represent the tables in the database. Example:

```
@Entity
public class Hotel {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String location;

    // Getters and setters
}
```

- Create Repository Interfaces : Spring Data JPA uses repositories to provide CRUD functionality. Extend JpaRepository to automatically gain common CRUD operations. Example:

```
public interface HotelRepository extends JpaRepository<Hotel, Long> {
    List<Hotel> findByLocation(String location);
}
```

3.6 Deployment and The Different Interfaces

In these images we can see that the platform give an ability for hotel administrators to handle hotel information like name, location, description, services offered, list of languages supported, Contact information and room Info (photos, type, capacity, price) From the user side it gives they a basic and responsively booking interface where users select destination, check-in and check-out dates and number of guests. The app is fully responsive and runs great on desktop as well as phones. Moreover, It is also installable as Progressive Web App(PWA) so that user can install this app on their mobiles, which can be just used like a native application.

3.6.1 Deployment of the Hotel Reservation System

To make the application accessible to end users, both the frontend (React) and backend (Spring Boot) were deployed to cloud platforms:

1- Frontend: Hosted on Netlify https://sahladz.netlify.app

2- Backend: Deployed on Render https://sahladz.netlify.app

3- Database: Hosted on Railway (private url) https://railway.app

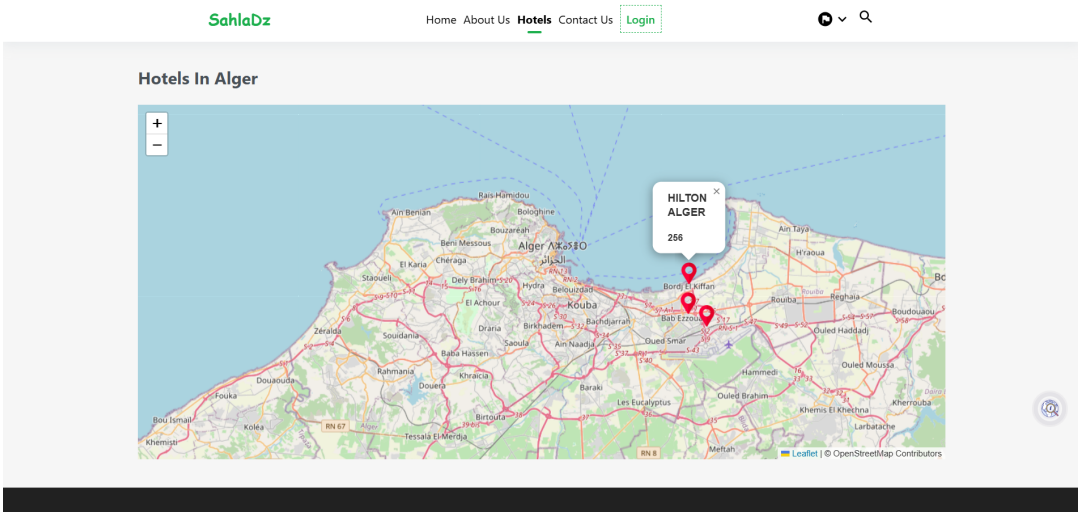

FIGURE 3.10: map page

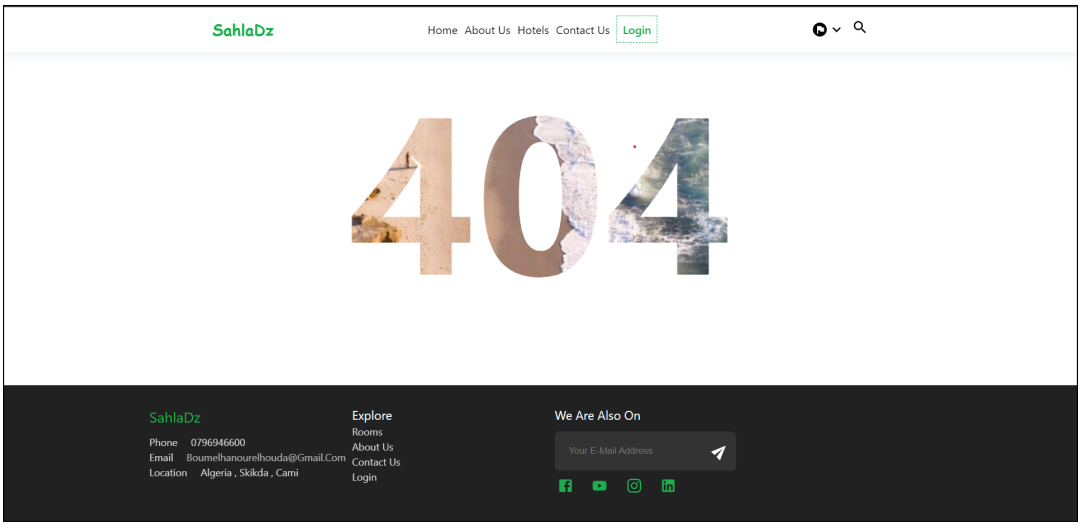

FIGURE 3.11: 404 page

FIGURE 3.12: home page

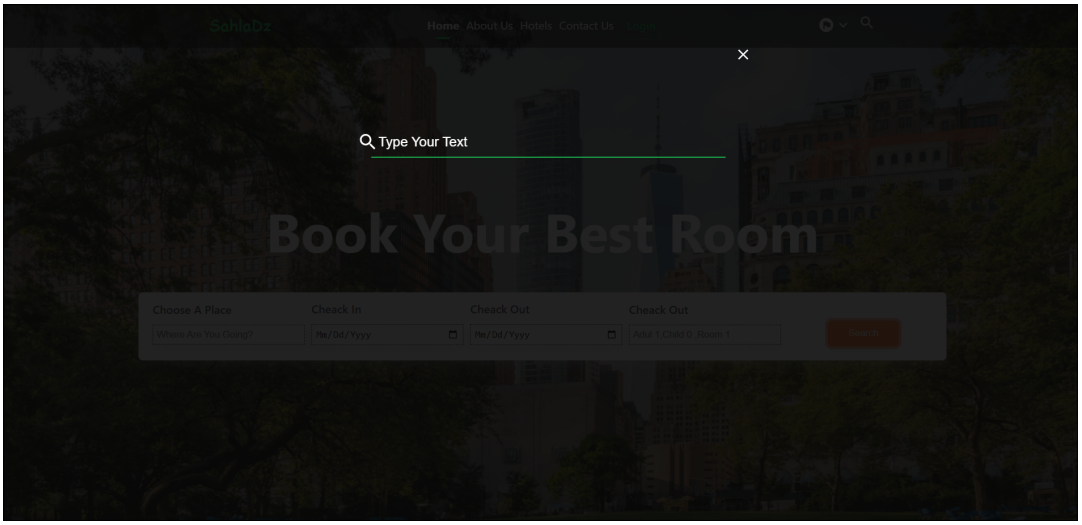

FIGURE 3.13: search bar

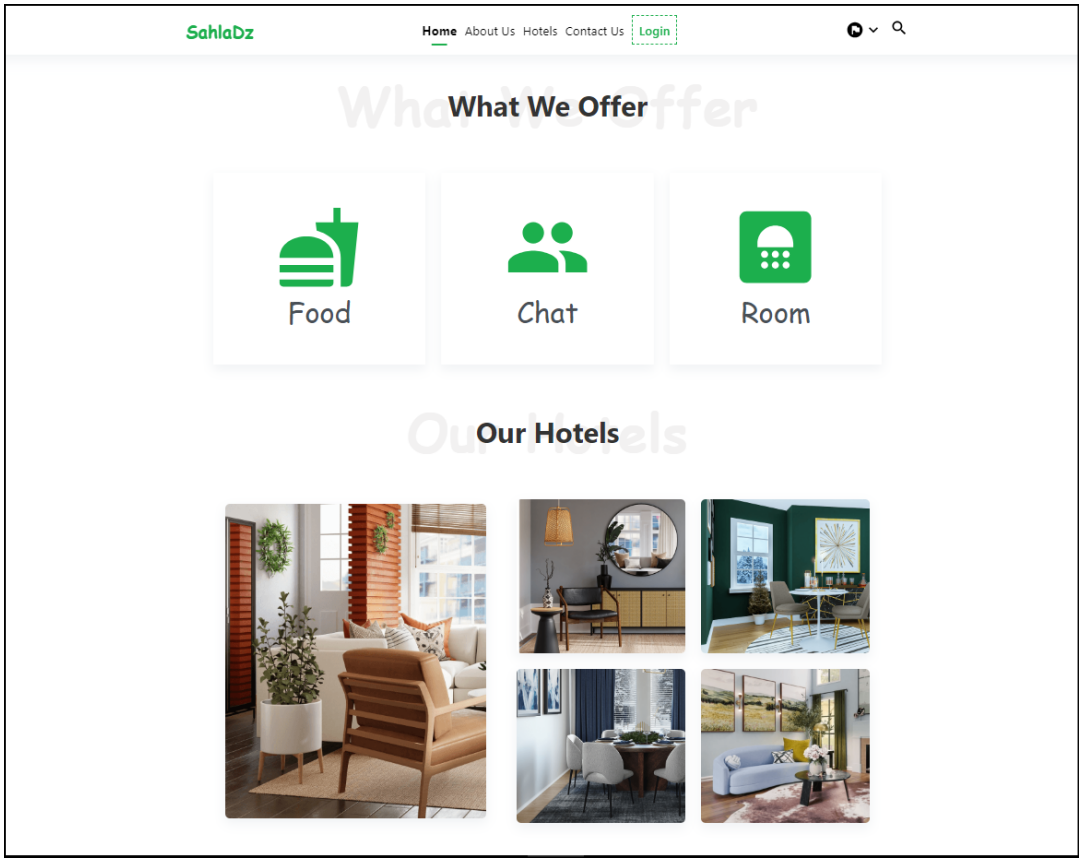

FIGURE 3.14: home section

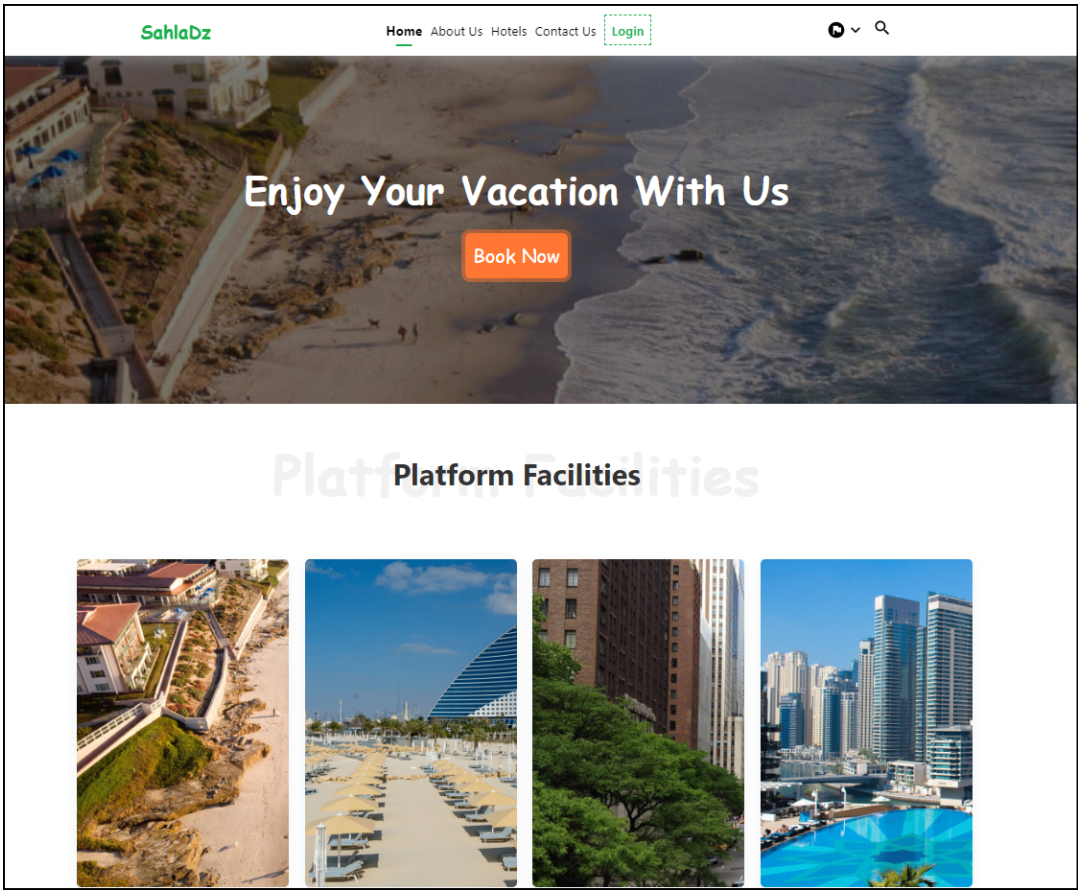

FIGURE 3.15: home paltform section

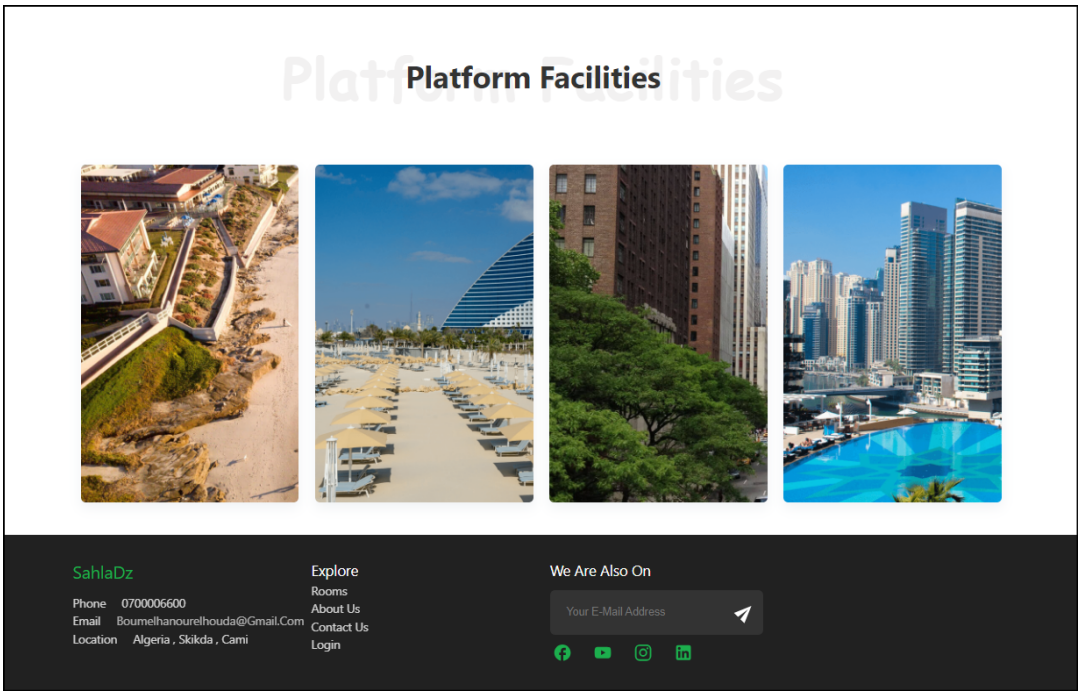

FIGURE 3.16: home paltform section

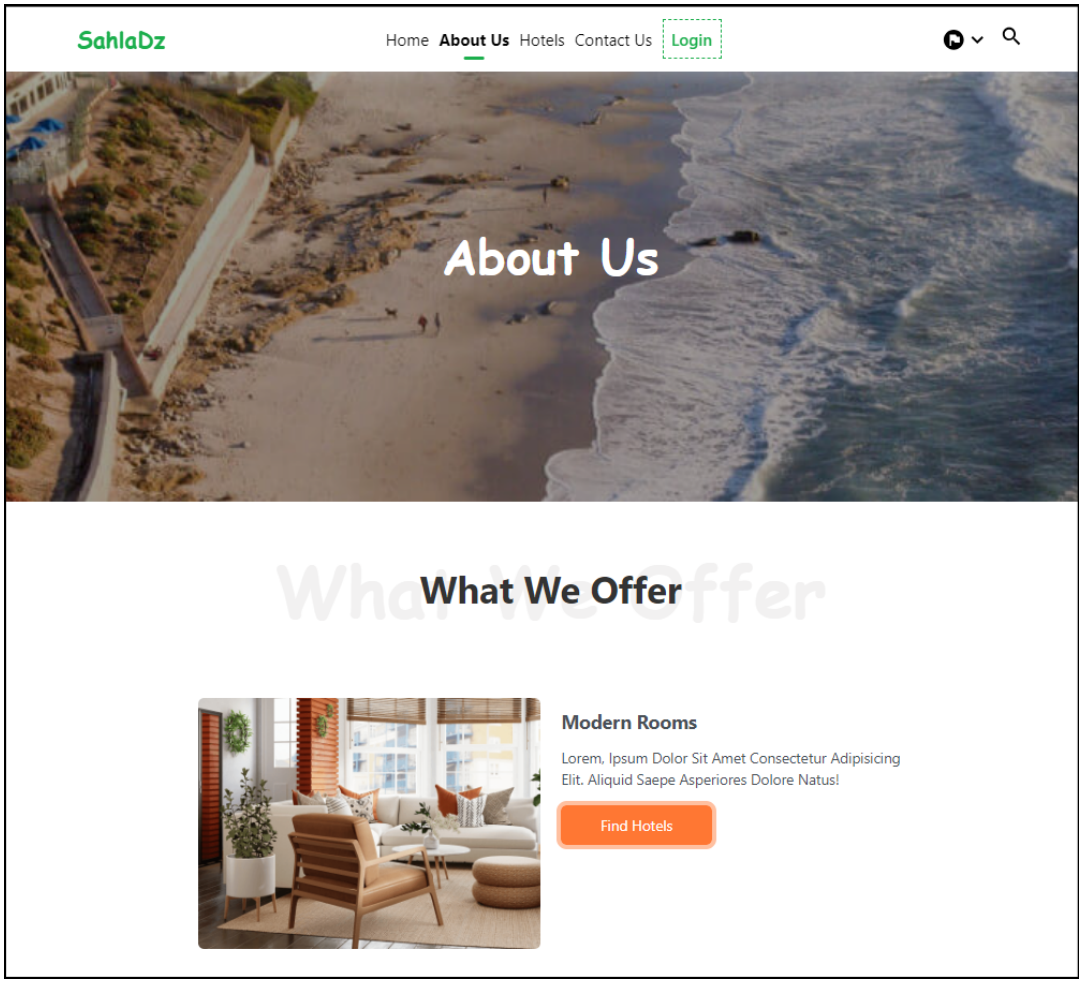

FIGURE 3.17: about us page

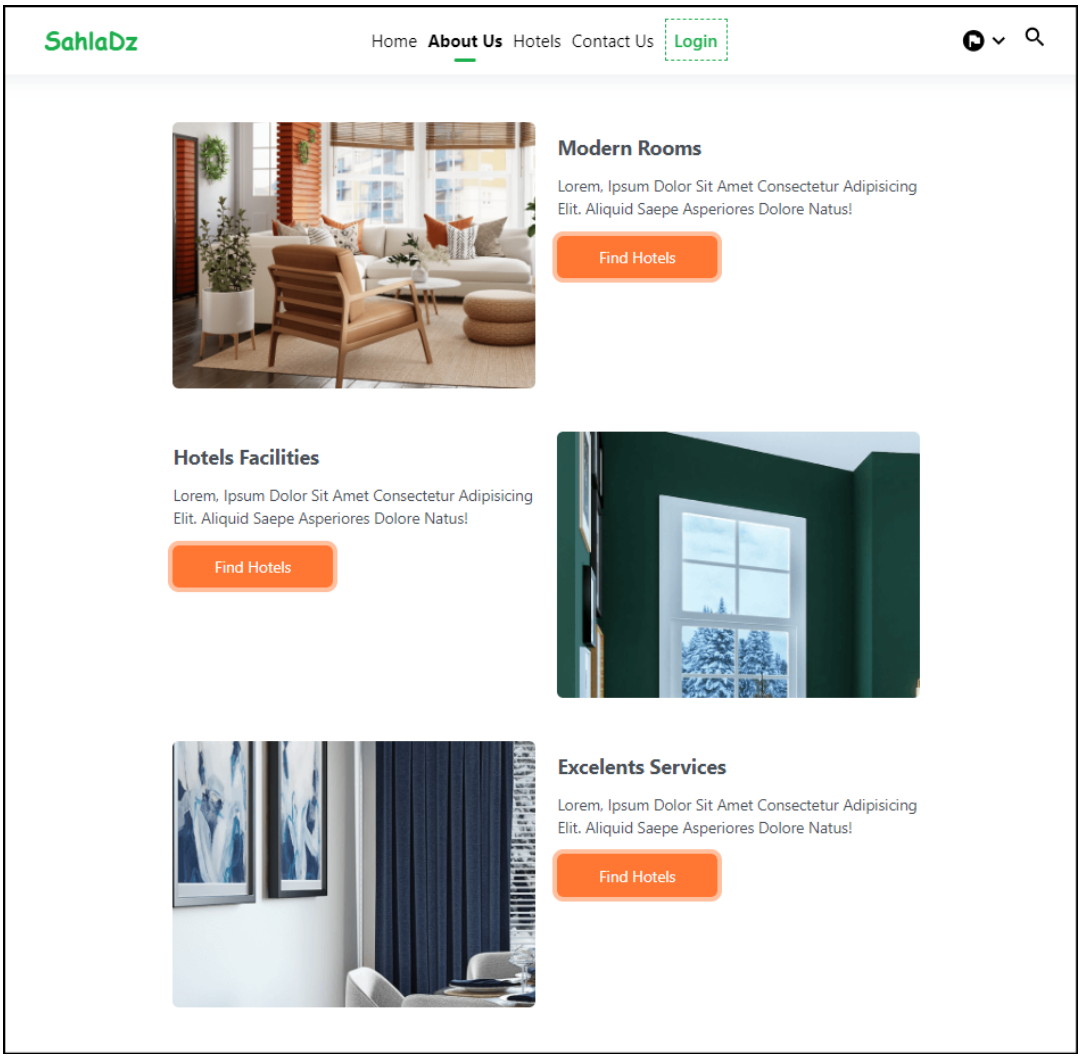

FIGURE 3.18: About us section

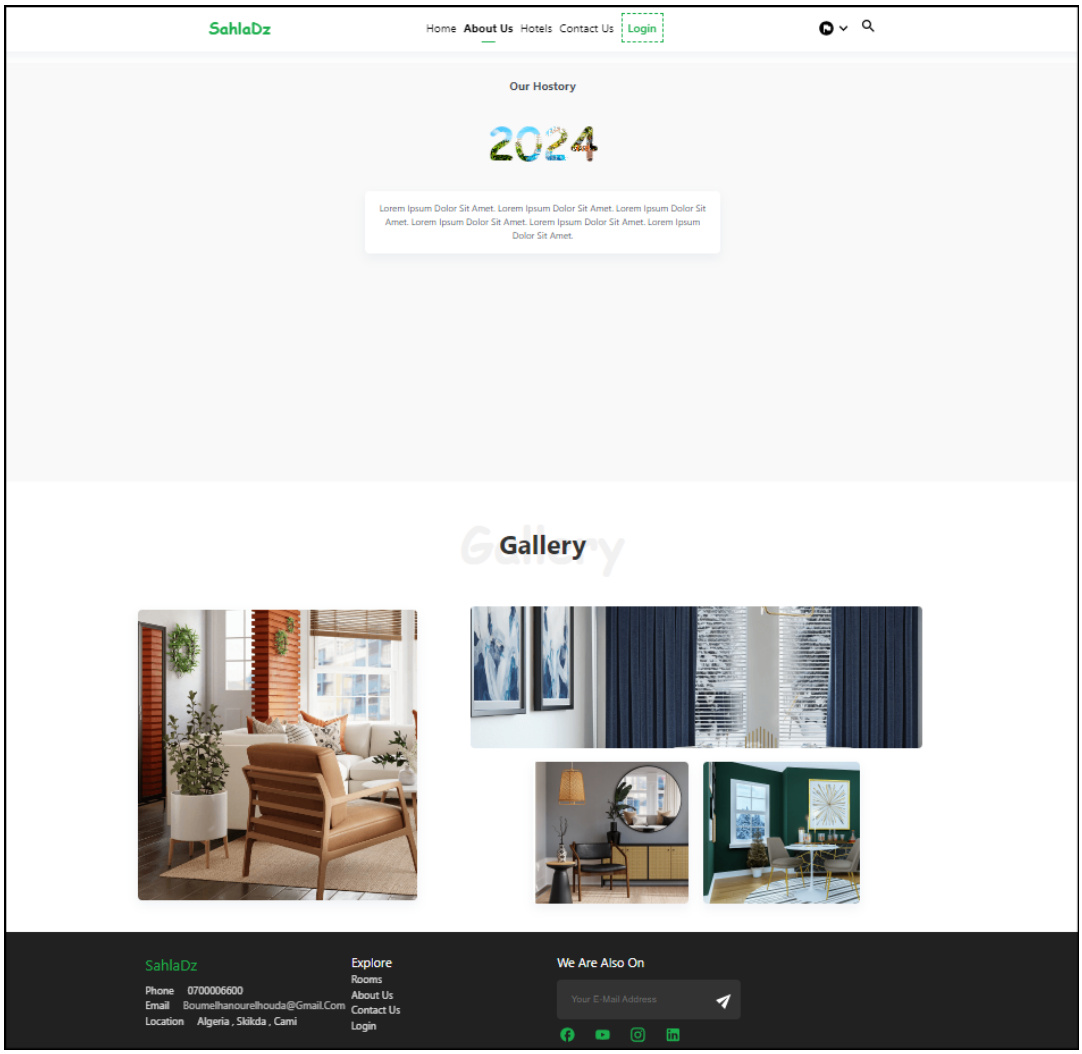

FIGURE 3.19: About us gallery sectione

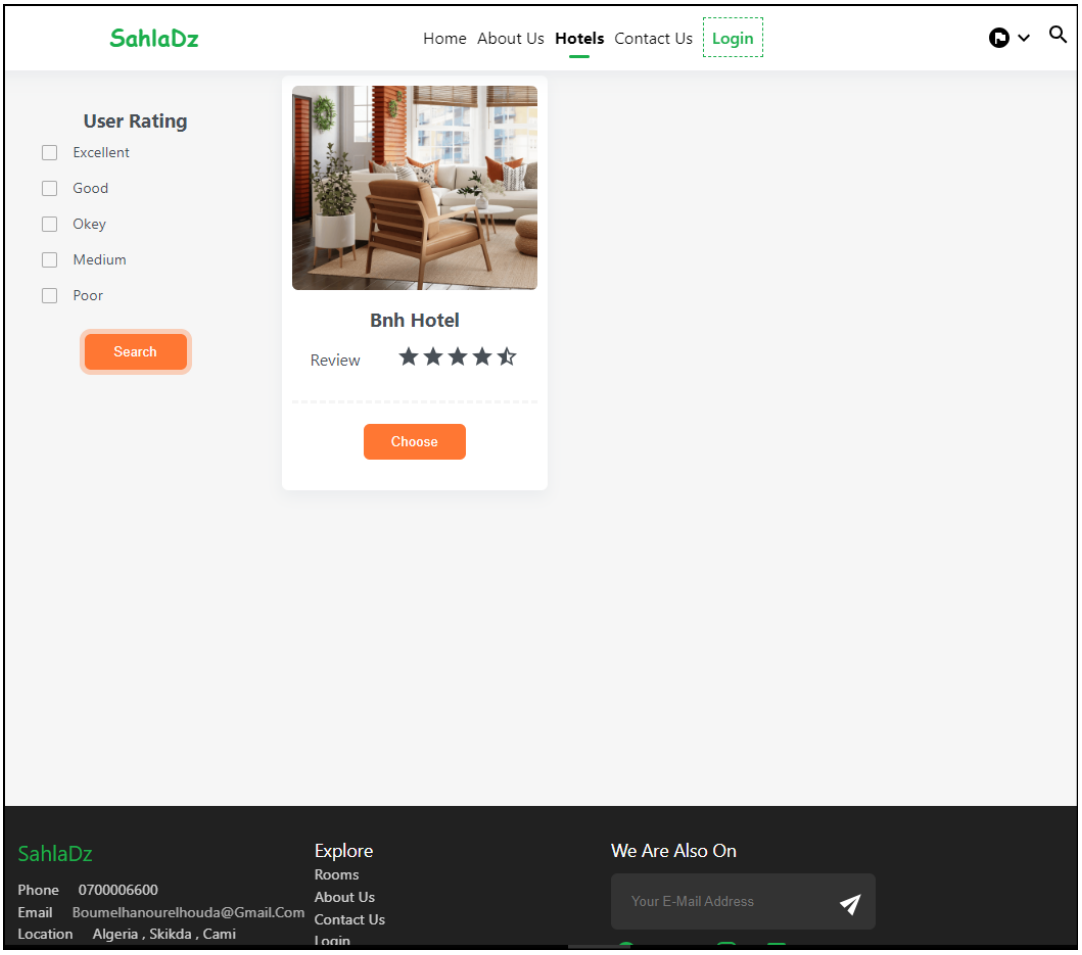

FIGURE 3.20: hotel page

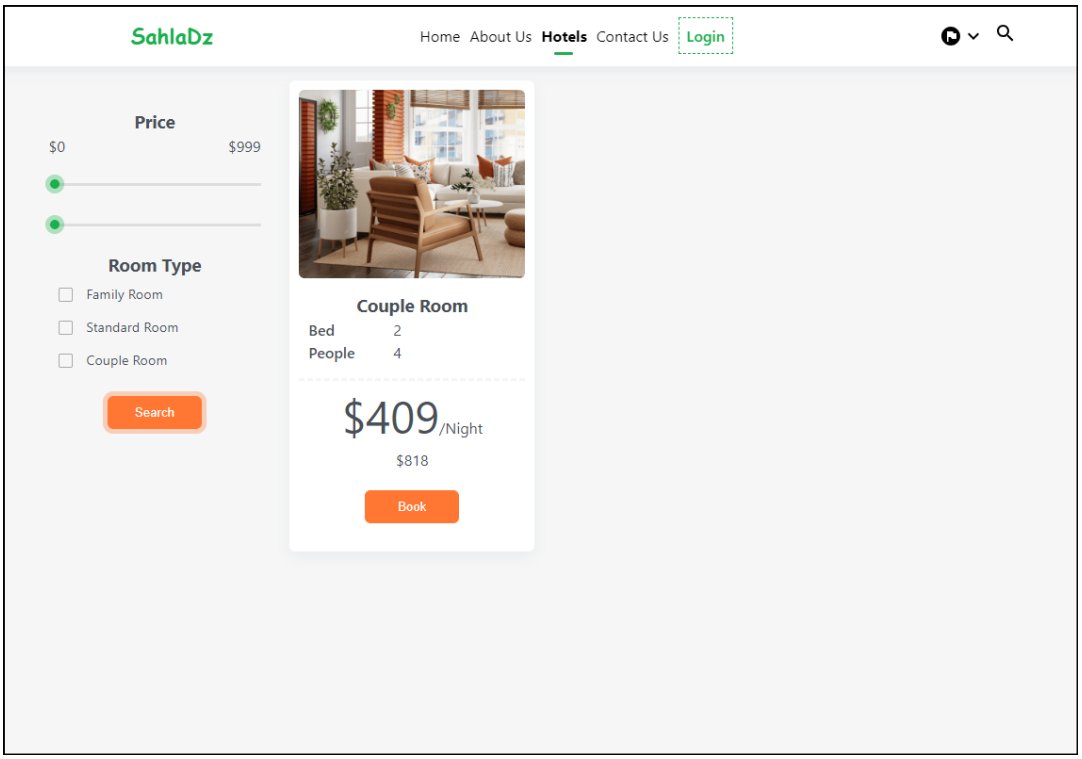

FIGURE 3.21: room page

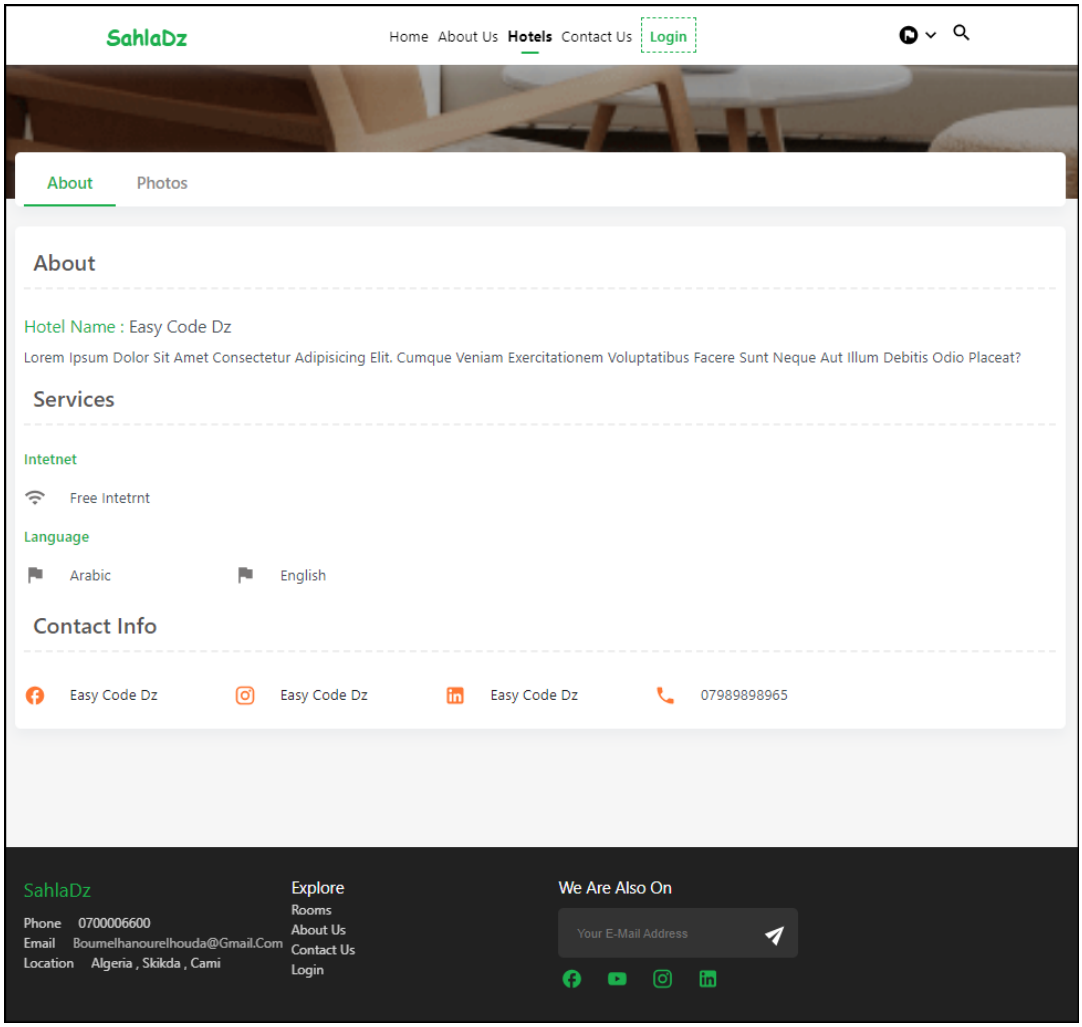

FIGURE 3.22: hote page description

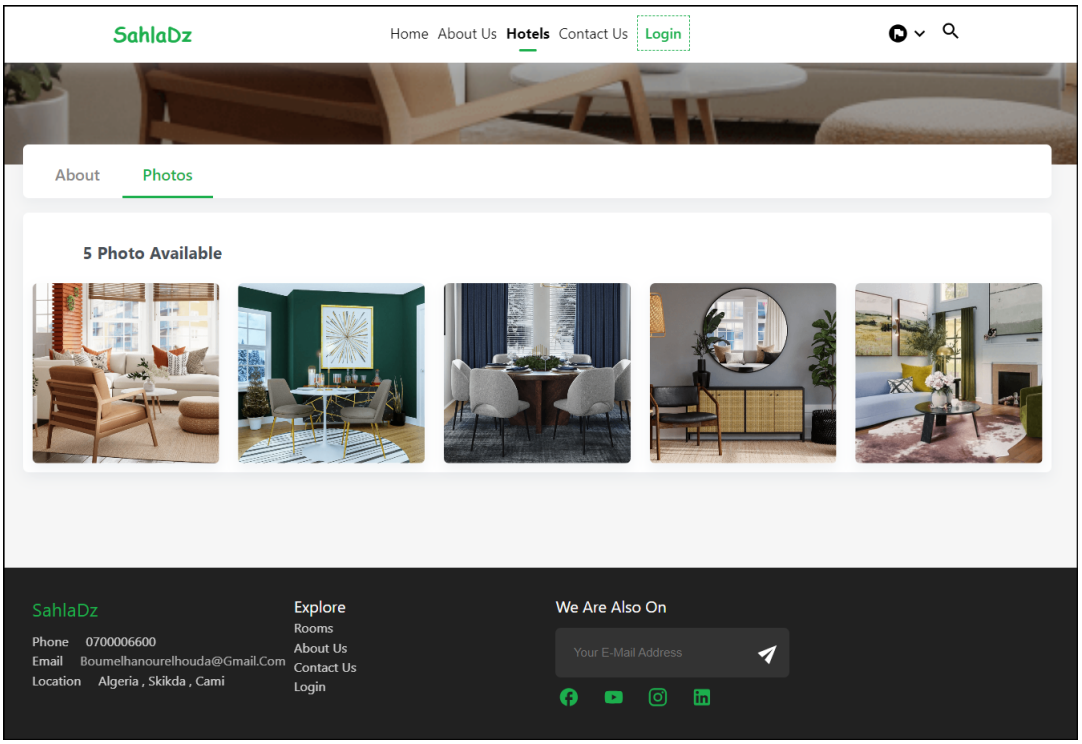

FIGURE 3.23: room page photos

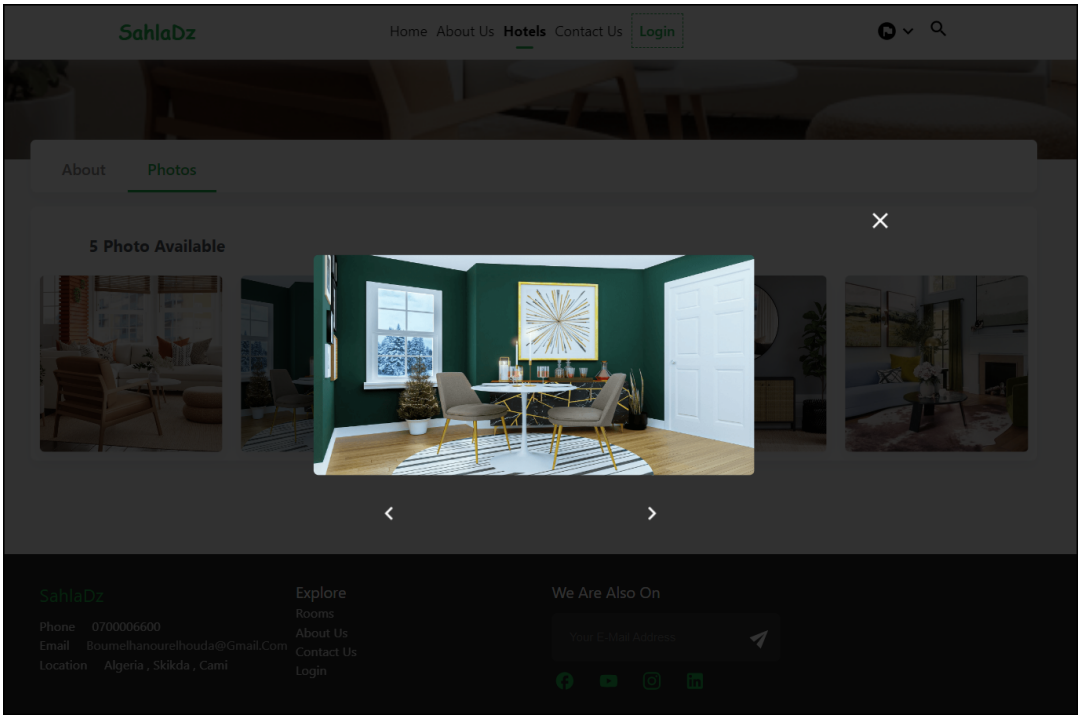

FIGURE 3.24: room carousel photos

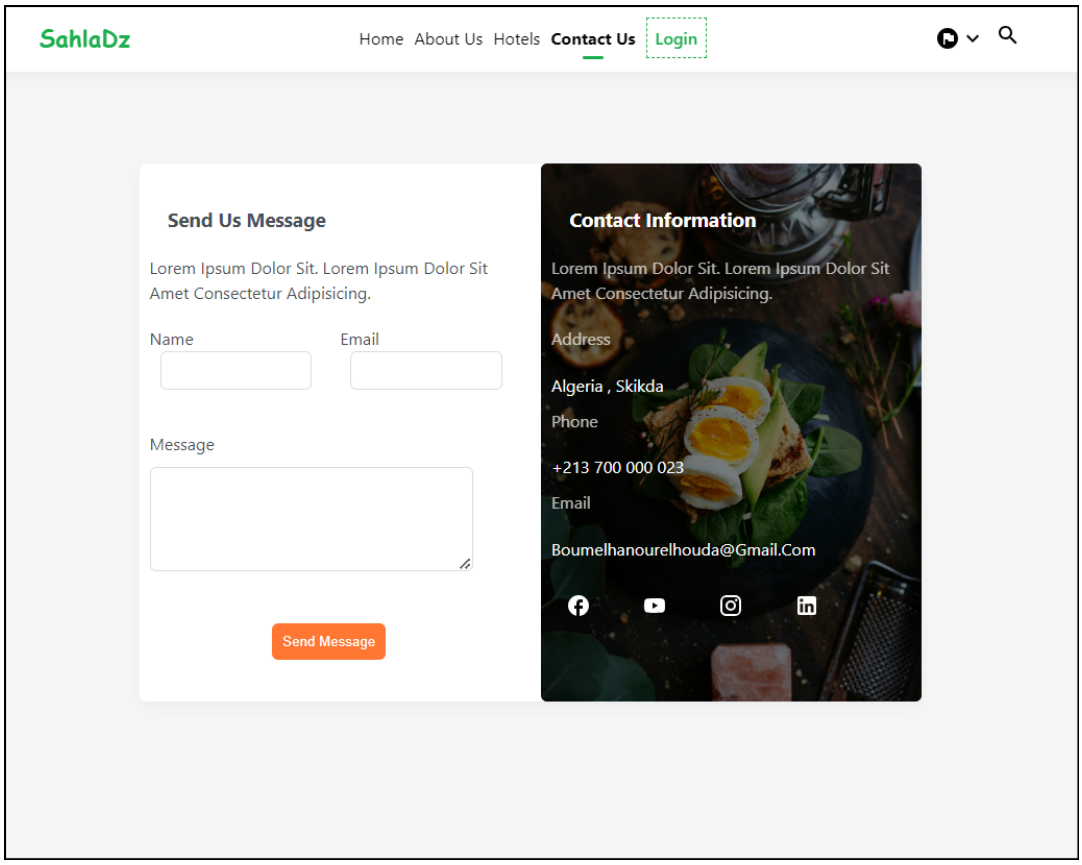

FIGURE 3.25: contact us page

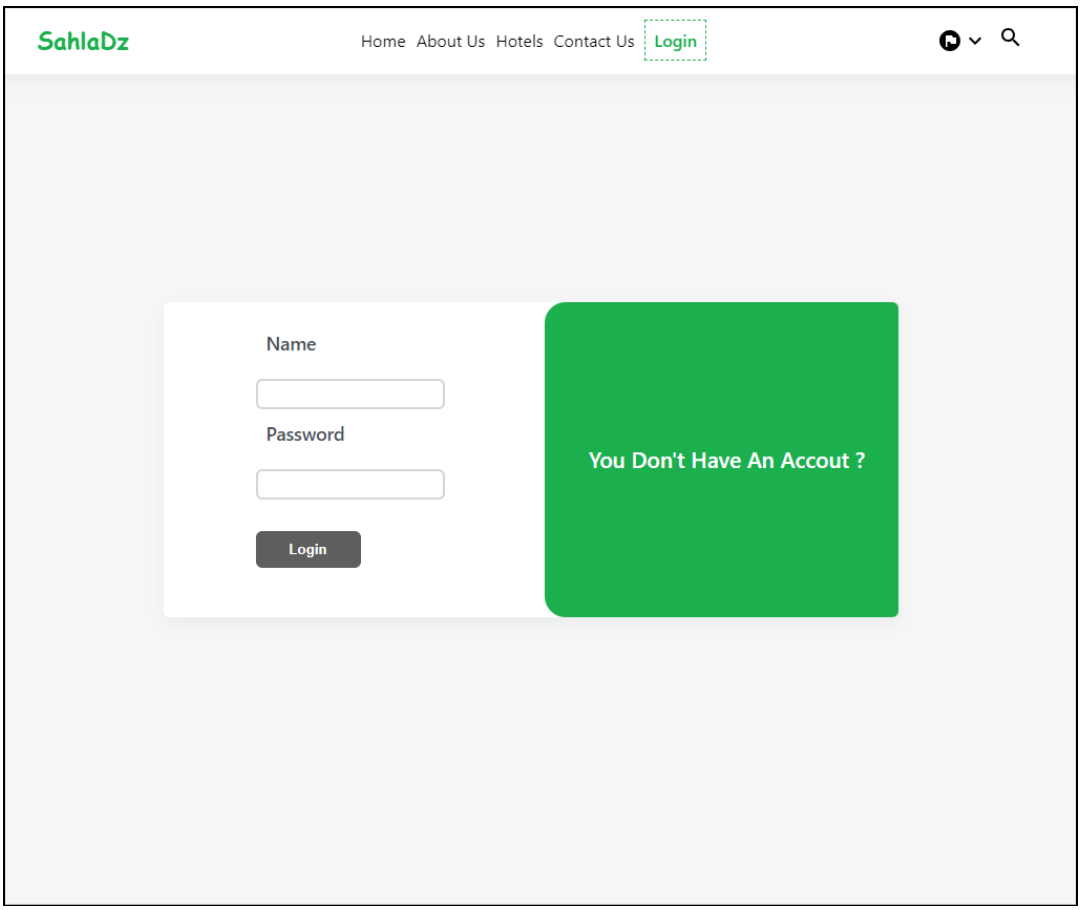

FIGURE 3.26: login page

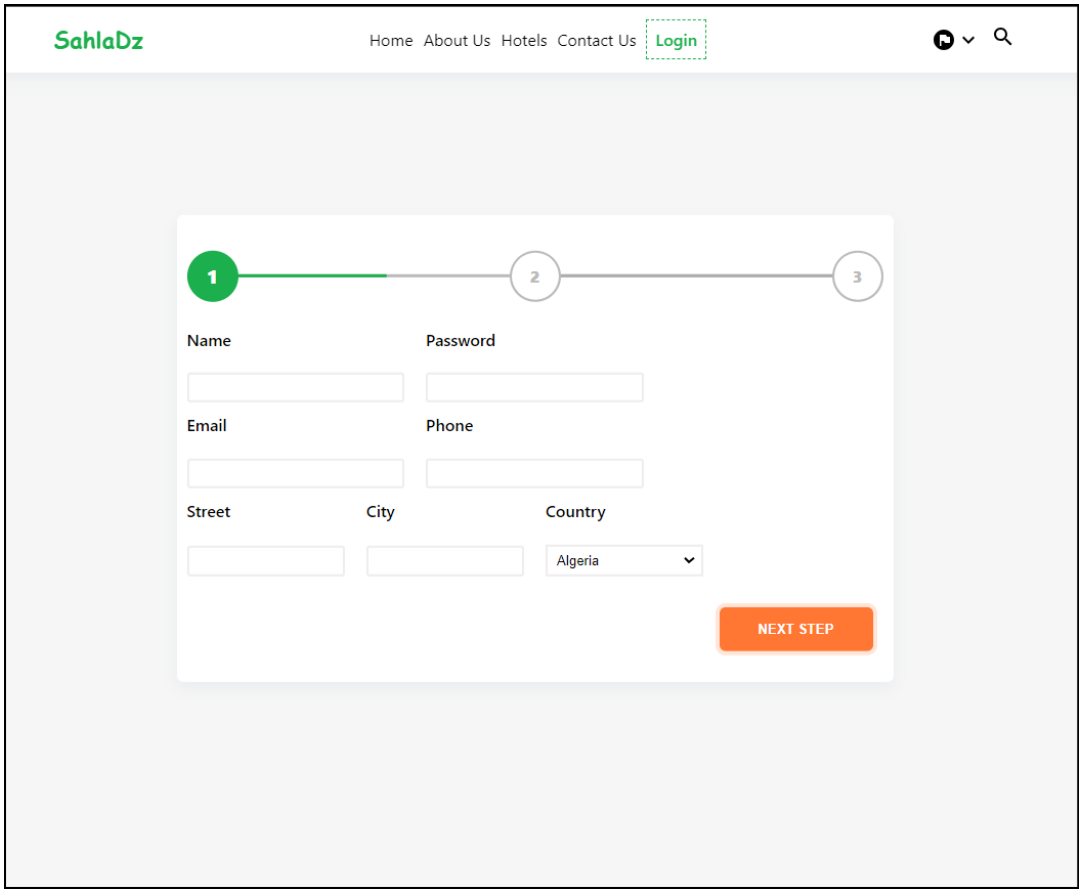

FIGURE 3.27: payment form

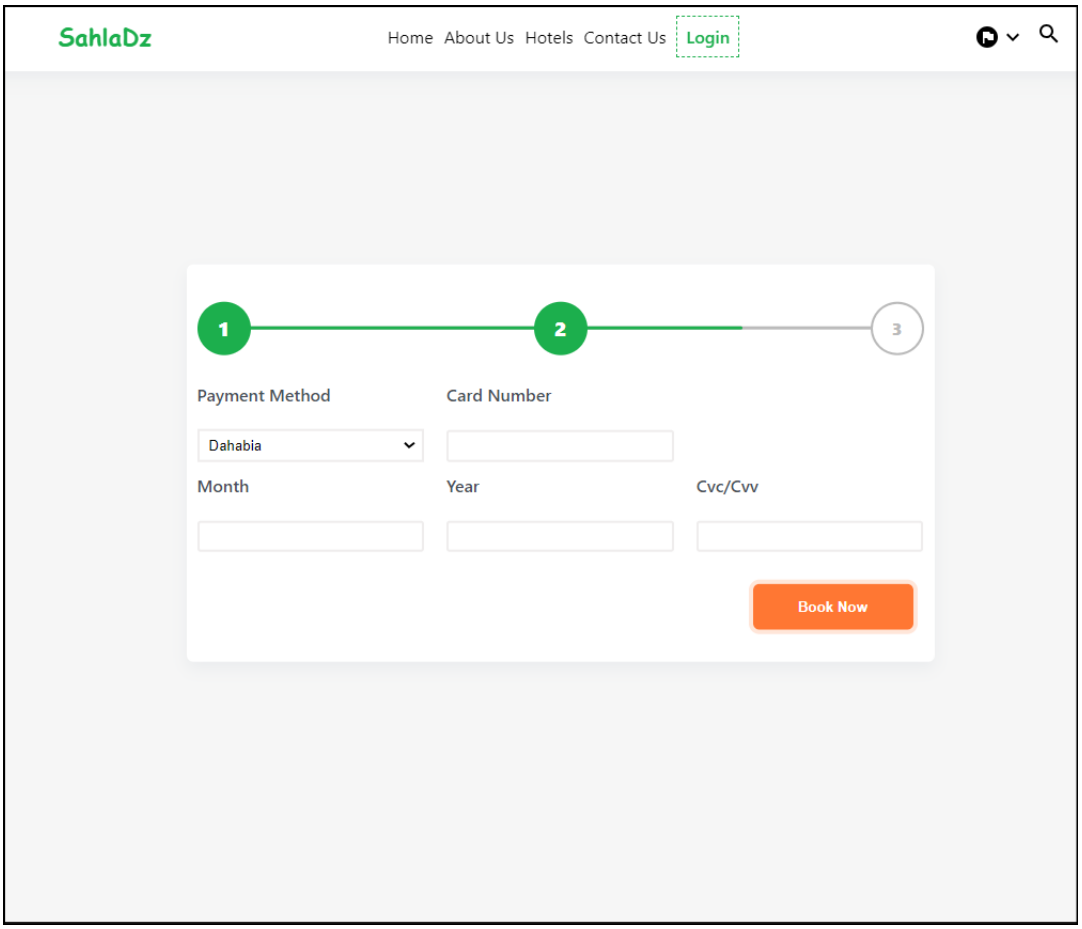

FIGURE 3.28: payment form

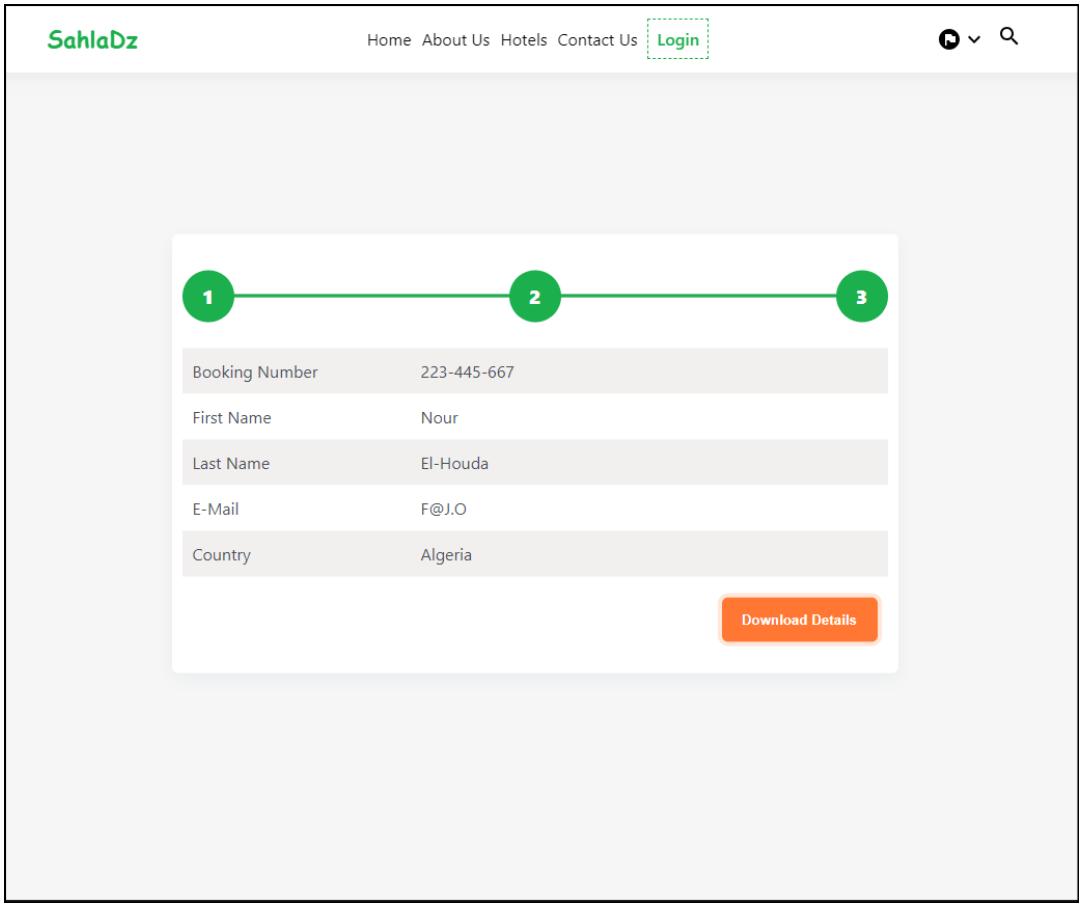

FIGURE 3.29: payment form

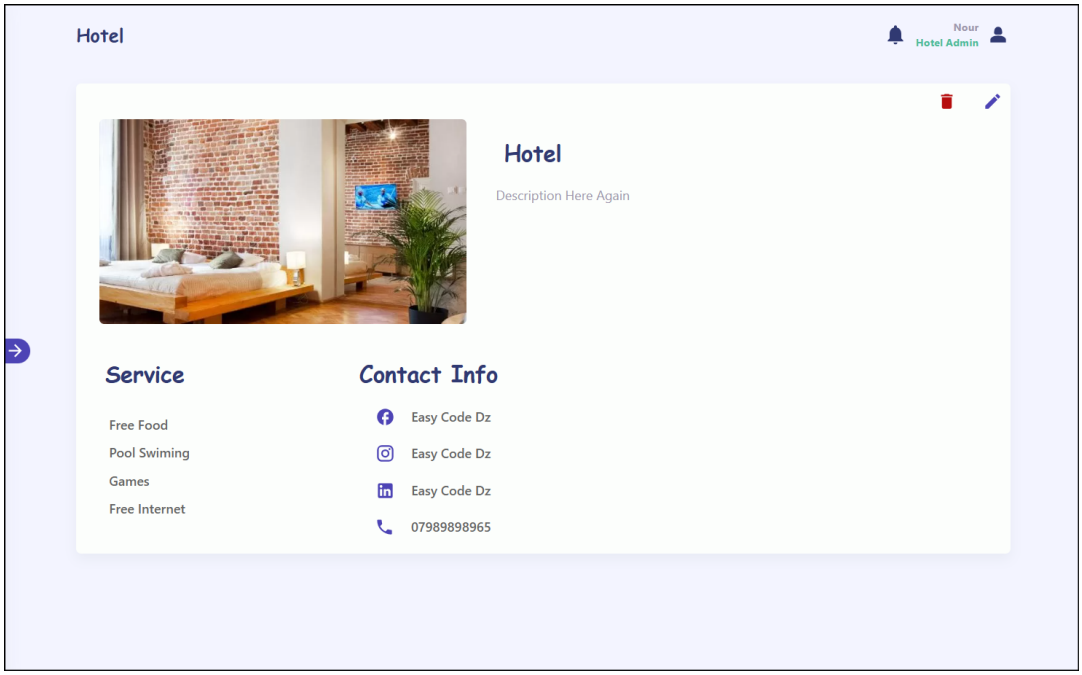

FIGURE 3.30: hotel dashboard

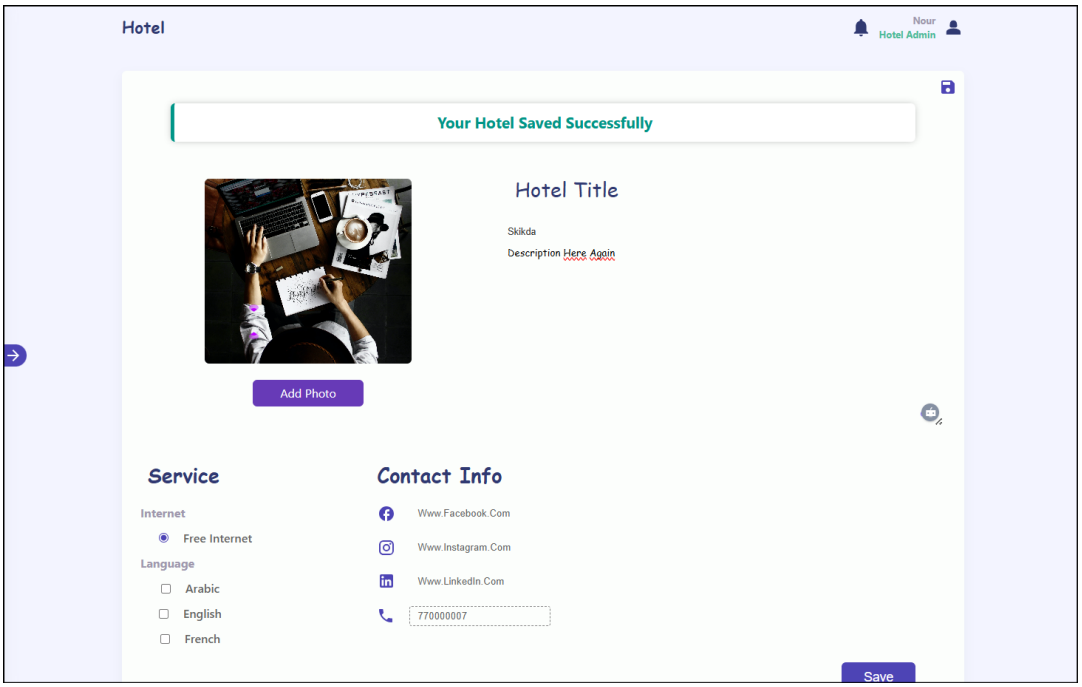

FIGURE 3.31: hotel dashboard

FIGURE 3.32: open desktop app

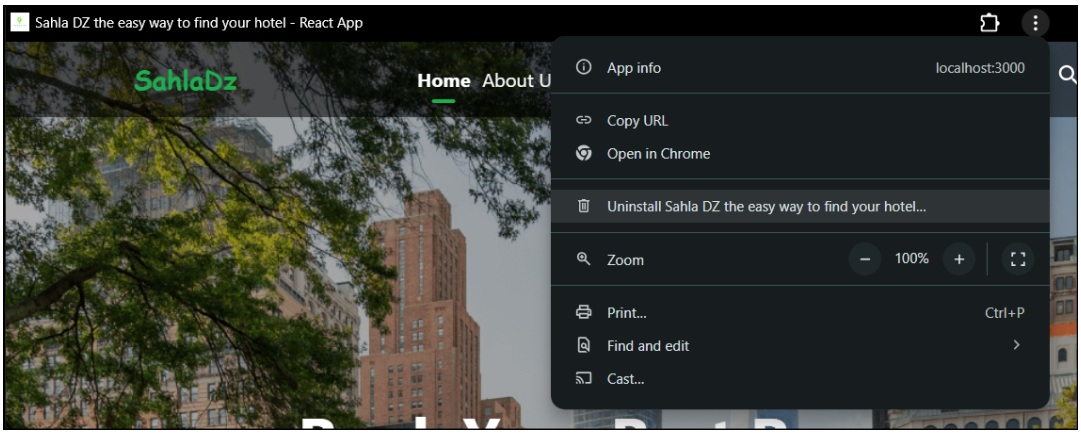

FIGURE 3.33: uninstal sahla dz

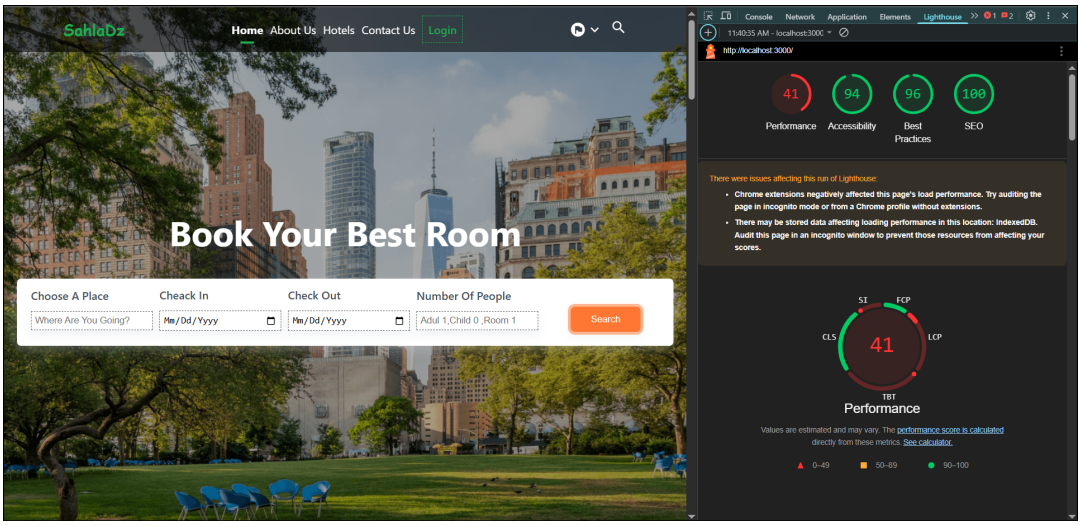

FIGURE 3.34: seo

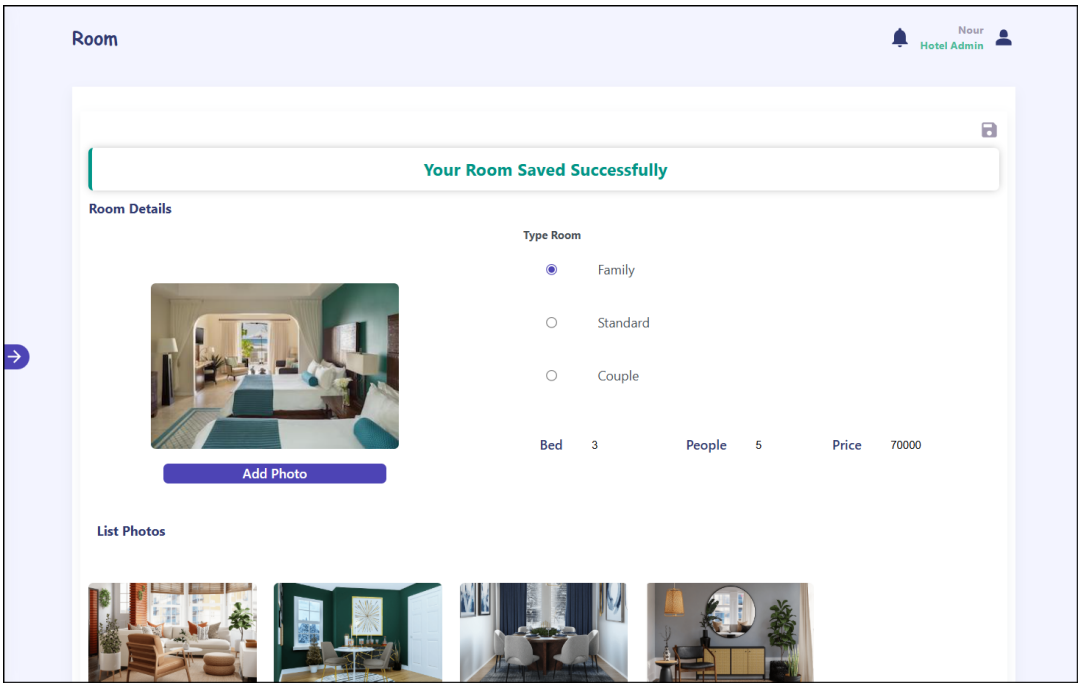

FIGURE 3.35: room dashboard

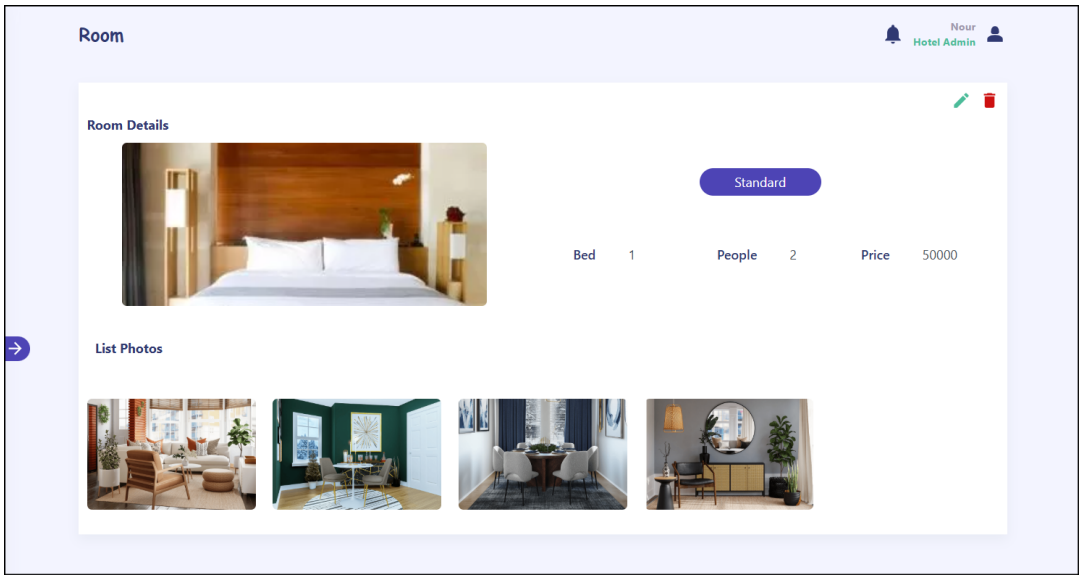

FIGURE 3.36: room dashboard

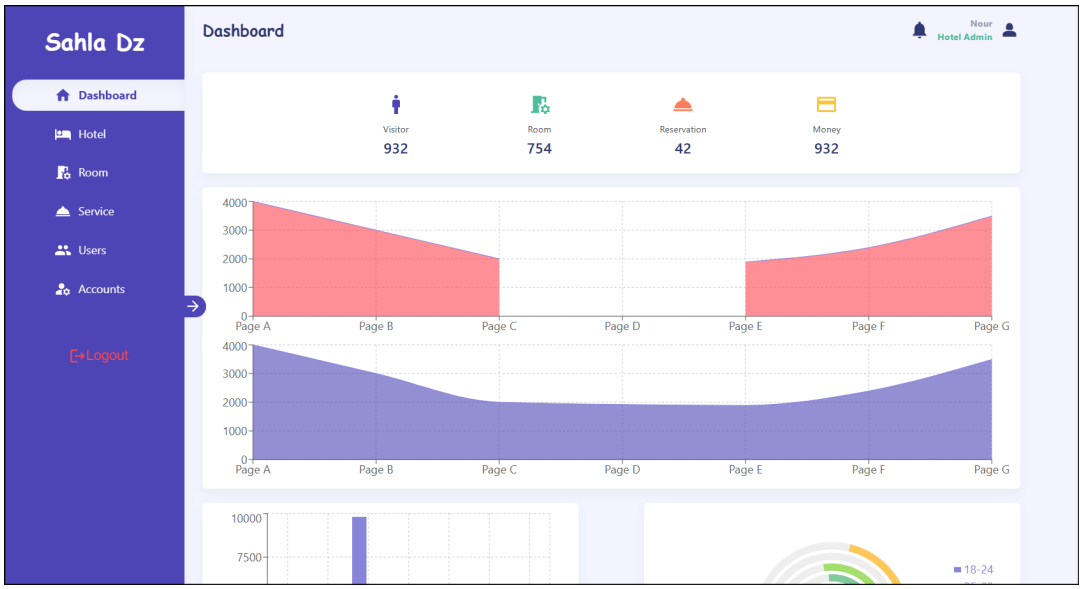

FIGURE 3.37: dashboard page

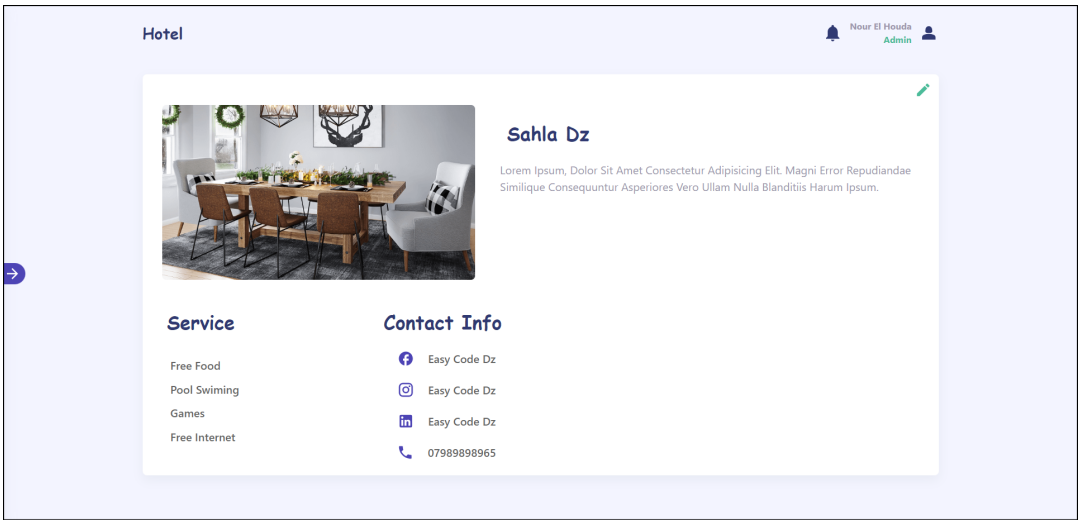

FIGURE 3.38: dashboard hotel page

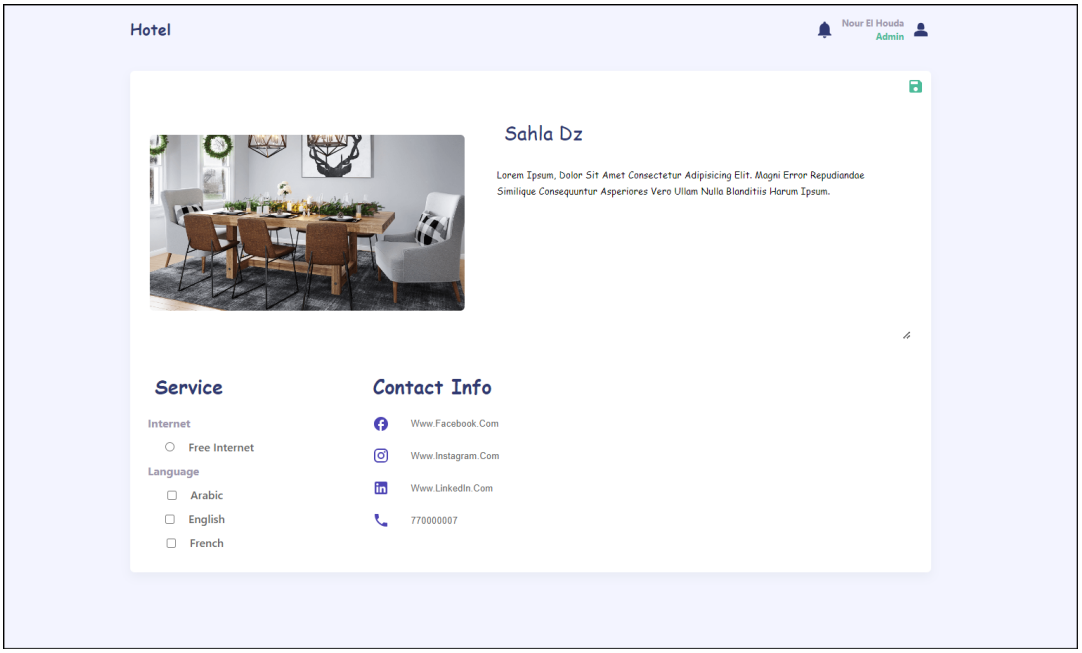

FIGURE 3.39: dashboard hotel page

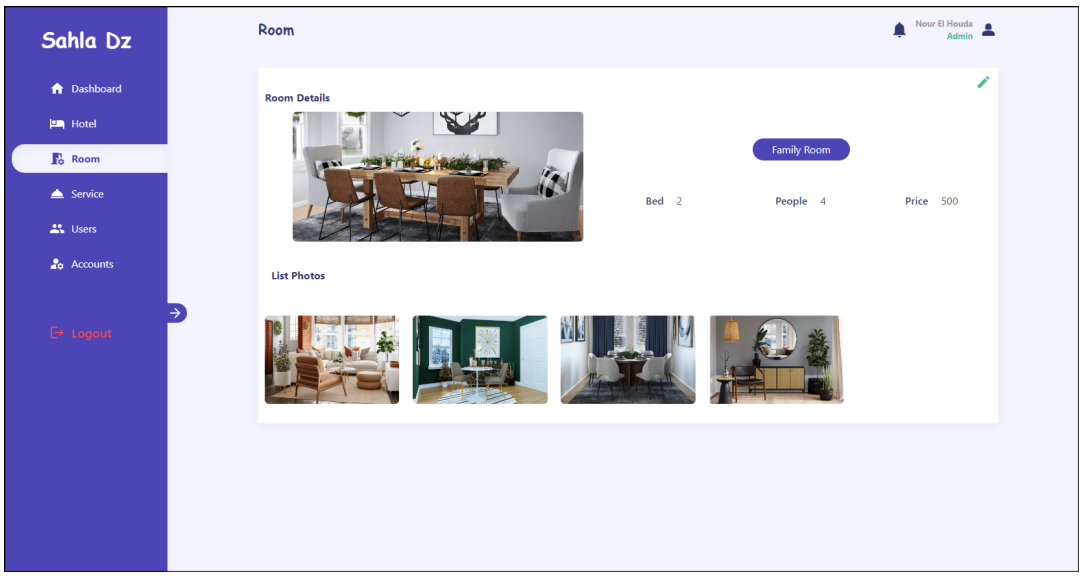

FIGURE 3.40: dashboard room page

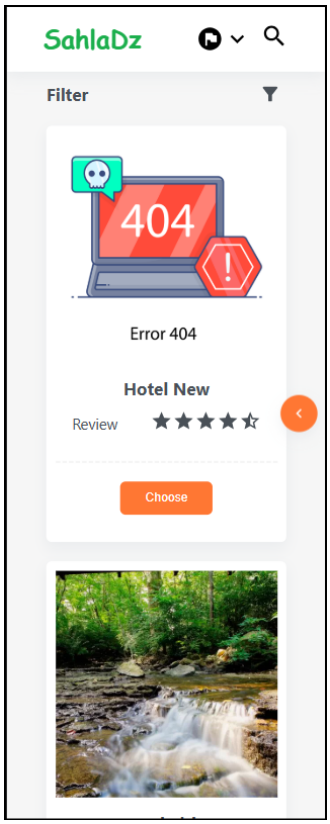

FIGURE 3.41: hotel list

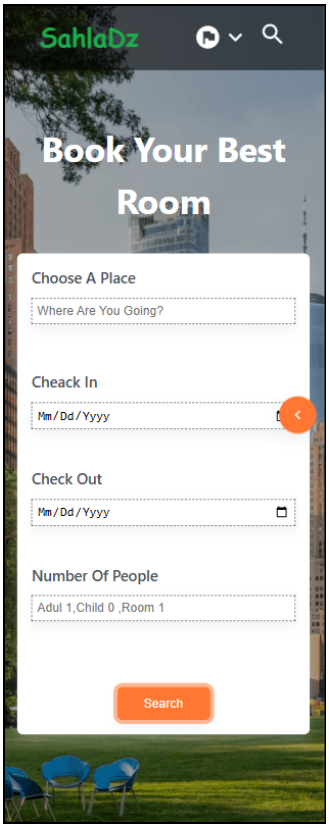

FIGURE 3.42: home page

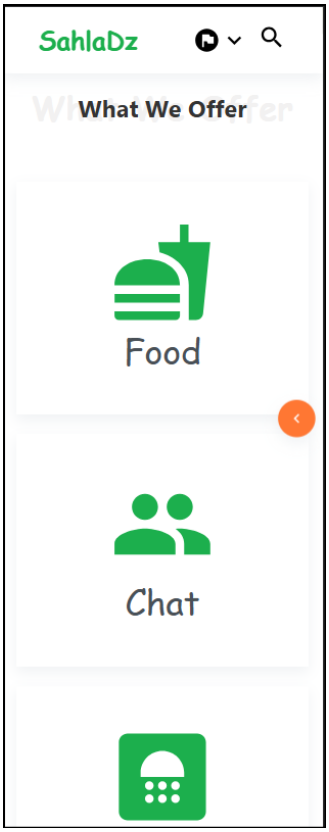

FIGURE 3.43: home page

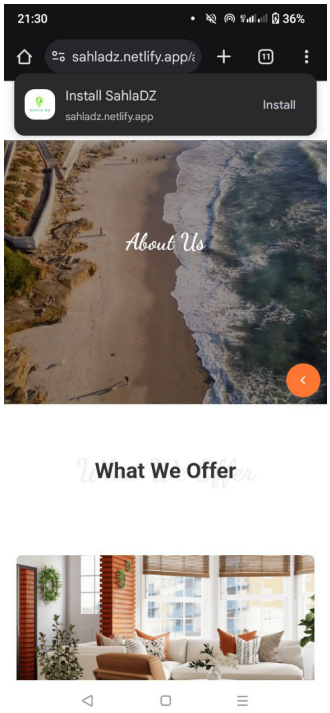

FIGURE 3.44: application install

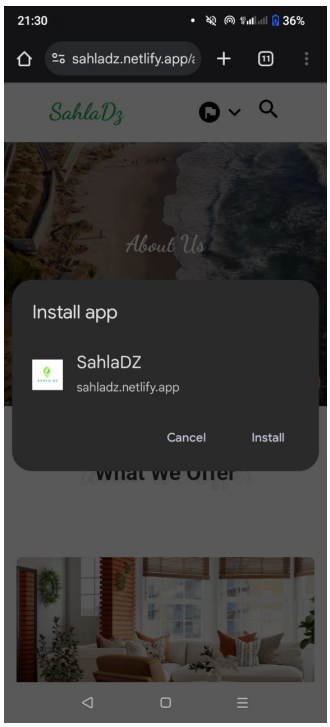

FIGURE 3.45: application install

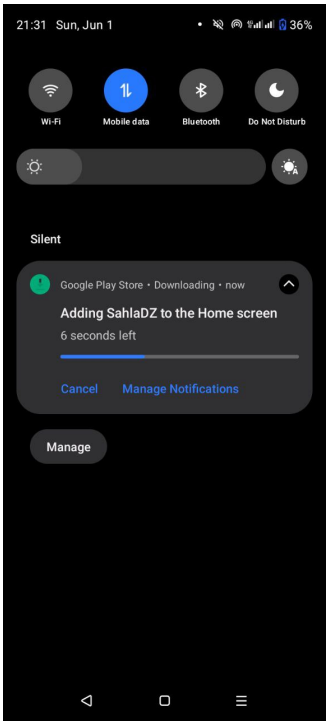

FIGURE 3.46: application install

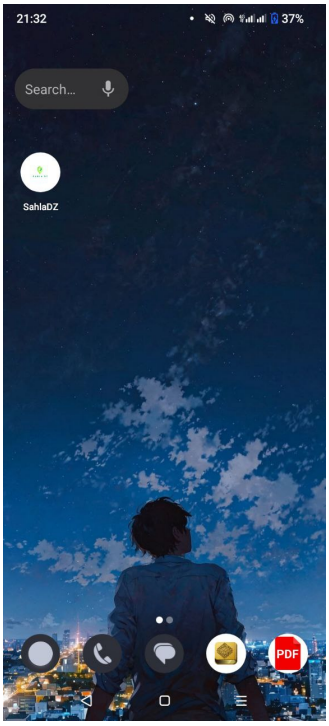

FIGURE 3.47: application install

FIGURE 3.48: application install

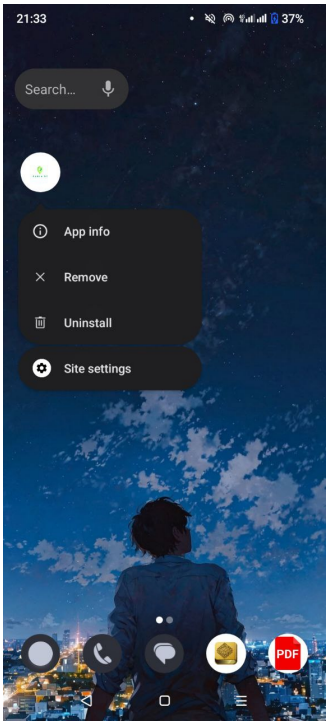

FIGURE 3.49: application install

3.7 Conclusion

In this chapter, we described the project development environment and architecture. We discussed what it was made of, the hardware and the software, mainly highlighting client-server advantages. Our frontend was built on HTML, CSS, JavaScript and ReactJS and the backend was Java using Spring + MySQL. We also integrated external APIs (Amadeus among) and the frontend. This broadened our approach which is necessary to provide a robust and effective system, in coordination with the project goals.

General Conclusion

The hotel industry has to wake up to the changing face of technology, needed for better workflow and next level user experience in this digital era. Sahla DZ, our project is aimed at solving this need by creating a single touch point, easy to use solution for both the travelers and hotel owners while booking a hotel.

In this thesis we researched the current systems, its shortcomings and put forward a simplified solution that combines different booking platforms to give users an easy booking experience. Using React for frontend, Spring Boot for backend and MySQL for data base we created a reliable, scalable, safe web application. Moreover, Leaflet maps and Amadeus APIs add to the features of Sahla DZ and it becomes a total hotel ERP system.

Client, server architecture implementation made the communication between frontend and backend smooth that build a swift, stable system. We also produced a good amount of UML diagrams and used the structured development process in order for writing verifiable application that is easy to maintain.

All in all, Sahla DZ, provides a seamless and smart option for the hotel booking in Algeria. Intuitive, actionable and eases time management, decision-making and the user experience enabling its users to search, compare and book accommodations seamlessly. This is the foundation for potential future iterations (both AI-based recommendations, mobile and more hotel providers partnerships).

Digitisation of hotel booking process means pushing the hospitality industry in Algeria towards evolution, technological advancement and usability.

Bibliography

[1] Hadrani Nasution Siregar.A WK Siahaan. "COMPUTER NETWORK". In: *International Journal of Computer Network and Information Security* ((January 2023)). URL: https://www.researchgate.net/publication/367021503_Computer_Network.

[2] *What is Computer Networking*. Last accessed 27.0.2025. (12 Jun, 2024). URL: https://www.geeksforgeeks.org/what-is-computer-networking/.

[3] *Types of Computer Networks*. Last accessed 27.0.2025. (08 Jul, 2024). URL: https://www.geeksforgeeks.org/types-of-computer-networks/.

[4] *Introduction to Internet*. Last accessed 27.0.2025. (28 Jan, 2024). URL: https://www.geeksforgeeks.org/introduction-to-internet/.

[5] *Types of Internet Protocols*. Last accessed 27.0.2025. (21 Oct, 2024). URL: https://www.geeksforgeeks.org/types-of-internet-protocols/.

[6] *World Wide Web (WWW)*. Last accessed 27.0.2025. (03 Feb, 2025). URL: https://www.geeksforgeeks.org/world-wide-web-www/.

[7] *World Wide Web*. Last accessed 27.0.2025. (Dec 19, 2024). URL: https://www.geeksforgeeks.org/types-of-internet-protocols/.

[8] *Client-Server Model*. Last accessed 27.0.2025. (03 Jan, 2025). URL: https://www.geeksforgeeks.org/client-server-model/.

[9] *What is a Website*. Last accessed 04.03.2025. (18 Apr, 2023). URL: https://www.geeksforgeeks.org/what-is-a-website/.

[10] *Web Pages*. Last accessed 04.03.2025. (02 May, 2023). URL: https://www.geeksforgeeks.org/web-pages/.

[11] *Browsing the web*. Last accessed 04.03.2025. (Feb 25, 2025). URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Environment_setup/Browsing_the_web.

[12] *Dynamic Websites*. Last accessed 05.03.2025. (20 Aug, 2024). URL: http://geeksforgeeks.org/dynamic-websites/.

[13] *What is a Website*. Last accessed 05.03.2025. (25 Aug, 2022). URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Environment_setup/Browsing_the_web.

[14] Laurent AUDIBERT. "UML 2 De l'apprentissage à la pratique". In: ((31 octobre 2006 , uppdate 12 janvier 2009)), p. 12. URL: https://d1wqtxts1xzle7.cloudfront.net/53584327/uml2-apprentissage-pratique-libre.pdf?1497899597=&response-content-disposition=inline%3B+filename%3DUML_2_De_lapprentissage_a_la_pratique.pdf&Expires=1741169898&Signature=KE4qfgMRwT~sqtnEGhZZVyYCDNgn%20mjcbYcGpV4yNV1ZGIn7MfPVSyFMr6CaKqZOhijybvfC-43NlgiRbhEuP%20eJF4JEYtGtV8EpknlMcAwC0vZWEgxM3TW-tN-d5qu2PMFqv18vz6~Qrb%20Bzpnhp7MF1t-bP3J-R3Jzo-SeTQQV3q~SwEYdXPSANOD9klEf23MUBhEBL%20~flCSrGGrqkq4NOCcaCrRcz1KQeeVEE0Itmnd99w8J4PbpUGTgKJQE1gM1%

204zPIX~2cvJKkE4jxOw2~xBSDcDbV7Ulz~eN4Ue~4~Vo1eKZ2wSr6oSTfs9C%20PifNijAotk0cOoKLUnY0w7LahQQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA.

[15] *What is UML*. Last accessed 05.03.2025. (Updated July 2005). URL: https://www.uml.org/what-is-uml.htm.

[16] *What are the advantages and disadvantages of UML?* Last accessed 05.03.2025. (23 Feb, 2024). URL: https://www.geeksforgeeks.org/what-are-the-advantages-and-disadvantages-of-uml.

[17] *Introduction to UML Diagrams*. Last accessed 05.03.2025. (21 Oct, 2021). URL: https://www.geeksforgeeks.org/introduction-to-uml-diagrams.

[18] *Use Case Diagram - Unified Modeling Language (UML)*. Last accessed 05.03.2025. (03 Jan, 2025). URL: https://www.geeksforgeeks.org/use-case-diagram/.

[19] *UML Diagram Types Guide: Learn About All Types of UML Diagrams with Examples*. Last accessed 05.03.2025. (6 January 2025). URL: https://creately.com/blog/diagrams/uml-diagram-types-examples.

[20] *Class Diagram | Unified Modeling Language (UML)*. Last accessed 05.03.2025. (03 Jan, 2025). URL: https://www.geeksforgeeks.org/unified-modeling-language-uml-class-diagrams.

[21] *Sequence Diagrams – Unified Modeling Language (UML)*. Last accessed 05.03.2025. (03 Jan, 2025). URL: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/.

[22] *Data Models in DBMS*. Last accessed 10.03.2025. (13 Jan, 2025). URL: https://www.geeksforgeeks.org/data-models-in-dbms/.

[23] *Introduction of ER Model*. Last accessed 10.03.2025. (16 Jan, 2025). URL: https://www.geeksforgeeks.org/introduction-of-er-model/.

[24] *Weak Entity Set in ER diagrams*. Last accessed 10.03.2025. (28 Dec, 2024). URL: https://www.geeksforgeeks.org/weak-entity-set-in-er-diagrams/.

[25] *Relational Model in DBMS*. Last accessed 10.03.2025. (29 Jan, 2025). URL: https://www.geeksforgeeks.org/relational-model-in-dbms/.

[26] *Client-Server Architecture - System Design*. Last accessed 16.03.2025. (16 Jul, 2024). URL: https://www.geeksforgeeks.org/client-server-architecture-system-design/.

[27] *Architecture Styles in Distributed Systems*. Last accessed 16.03.2025. (06 Aug, 2024). URL: https://www.geeksforgeeks.org/architecture-styles-in-distributed-systems/.

[28] *HTML5 | Introduction*. Last accessed 16.03.2025. (29 Jul, 2024). URL: https://www.geeksforgeeks.org/html5-introduction/.

[29] *CSS Introduction*. Last accessed 16.03.2025. (30 Jan, 2025). URL: https://www.geeksforgeeks.org/css-introduction/.

[30] *JavaScript Form Validation*. Last accessed 16.03.2025. (09 Jan, 202). URL: https://www.geeksforgeeks.org/form-validation-using-javascript/.

[31] *ReactJS Functional Components*. Last accessed 16.03.2025. (17 Feb, 2025). URL: https://www.geeksforgeeks.org/reactjs-functional-components/.

[32] *MySQL Tutorial*. Last accessed 16.03.2025. (26 Aug, 2024). URL: https://www.geeksforgeeks.org/mysql-tutorial/.

[33] *Introduction to MySQL*. Last accessed 16.03.2025. URL: https://www.w3schools.com/mySQl/mysql_intro.asp.

[34] *Leaflet*. Last accessed 08.06.2025. 2011. URL: https://leafletjs.com/.

[35] *amadeus*. Last accessed 16.03.2025. 2025-03-11. URL: https://developers.amadeus.com.

[36] *Java Introduction*. Last accessed 16.03.2025. URL: https://www.w3schools.com/java/java_intro.asp.

[37] *Introduction to Spring Framework*. Last accessed 16.03.2025. (03 Mar, 2025). URL: https://www.geeksforgeeks.org/introduction-to-spring-framework/.

[38] *Spring Tutorial*. Last accessed 23.03.2025. (02 Jan, 2025). URL: https://www.geeksforgeeks.org/spring/.

[39] *10 Reasons to Use Spring Framework in Projects*. Last accessed 23.03.2025. (26 May, 2022). URL: https://www.geeksforgeeks.org/10-reasons-to-use-spring-framework-in-projects/.

[40] *Spring Framework Architecture*. Last accessed 23.03.2025. (26 Feb, 2025). URL: https://www.geeksforgeeks.org/spring-framework-architecture/.

[41] *Introduction to Spring Framework*. Last accessed 23.03.2025. (03 Mar, 2025). URL: https://www.geeksforgeeks.org/introduction-to-spring-framework/.

[42] *Spring Security with Maven*. Last accessed 23.03.2025. (20 May, 2024). URL: https://www.geeksforgeeks.org/spring-security-with-maven.

[43] *Spring Security Tutorial*. Last accessed 23.03.2025. (04 Jan, 2025). URL: https://www.geeksforgeeks.org/spring-security-tutorial.

[44] *Authorize HttpServletRequests*. Last accessed 06.04.2025. URL: https://docs.spring.io/spring-security/reference/servlet/authorization/authorize-http-requests.html.

[45] *JPA Query Methods*. Last accessed 06.04.2025. URL: https://docs.spring.io/spring-data/jpa/reference/jpa/query-methods.html.