**TravelBlogWebsite**

**WEBPROJECT(1) FWD291**

**Full Stack Web Developer**

**Supervisor:Dr.EmanMuhammadAl-Jawadi**

**Presented by:**

**BushraAbdulRahmanAl-Rahili**

**4455439**

**HanadiSaadAl-Rashidi4455726**

**DanaAbdullahAl-Wasabi 4456121**

**RawabimunawarAl-Rashidi4470256**

**RimaIbrahimAl-Harbi4454162**

**AcademicYear1445AH**

**Acknowledgement**

We would like to express our sincere thanks and gratitude to the academic staff at the college for their continuous support. We also extend our heartfelt thanks to our families and friends for their support, encouragement, and kind wishes throughout this journey. A special thanks and deep appreciation go to our supervisor, Dr. Eman Muhammad Mukhtar Al-Jawadi, for her valuable guidance and encouragement at every step of this project

**Abstract**

Tourism serves as a vital cornerstone of any country's economy, bearing witness to its civilization and values. It has come to our attention that both domestic and international tourists often encounter challenges in planning their trips to Saudi Arabia, lacking accessible resources to assist them. Consequently, obtaining information about tourism in Saudi Arabia can be time-consuming and cumbersome. That's why we conducted this survey—to ascertain the existence of an issue. Our findings revealed that a majority of individuals struggle to plan their trips without the assistance of a tour guide.

This leads us to our proposal, a “Blog Travel for the Kingdom of Saudi Arabia,” aimed at addressing this issue. Such a platform would assist individuals in efficiently planning their trips, thereby saving them time and effort. Additionally, it would broaden their understanding of Saudi Arabia, showcasing not only its renowned sites like Mecca and Medina but also its diverse array of entertainment, archaeological, and educational attraction

**ListofFigures**

[FIGURE1.1:PROJECTTIMELINE 10](#_bookmark11)

[Figure2.1:TheprincipalpageofwebsiteThedollzter 11](#_bookmark16)

[Figure2.2:Theprincipalpage ofwebsiteTravelBlog.com 12](#_bookmark17)

[Figure2.3:TheprincipalpageofwebsiteWanderlust's 13](#_bookmark18)

[Figure2.4:Theprincipalpage ofwebsiteTravel hero 14](#_bookmark19)

[Figure3.1:Usescase diagramforAdmin 18](#_bookmark30)

[Figure3.2:usescasediagramforUser 20](#_bookmark31)

[Figure3.3:Flow Chartofourwebsite 20](#_bookmark33)

[Figure3.4:AdvancedWaterfallmodel 21](#_bookmark35)

[Figure4.1:ClassDiagram 22](#_bookmark41)

[Figure4.2:Sequence DiagramforviewBlog 23](#_bookmark43)

[Figure4.3:Sequencediagramforcreateaccount 24](#_bookmark44)

TableofContent

Contents

[**ACKNOWLEDGEMENT 2**](#_bookmark0)

[**ABSTRACT 3**](#_bookmark1)

[**LISTOFTABLES 4**](#_bookmark2)

[**LISTOFFIGURES 5**](#_bookmark3)

[**TABLEOFCONTENT 6**](#_bookmark4)

[**CHAPTER1:INTRODUCTION 8**](#_bookmark5)

* 1. [Introduction 8](#_bookmark6)
  2. [ProblemDefinition 8](#_bookmark7)
  3. [ProjectObjectives 8](#_bookmark8)
  4. [ProjectScope 9](#_bookmark9)
  5. [ProjectTimeline 10](#_bookmark10)
  6. [DocumentOrganization 10](#_bookmark12)

[**CHAPTER2:LITERATUREREVIEW 11**](#_bookmark13)

* 1. [INTRODUCTION 11](#_bookmark14)
  2. [RelatedworkandSimilarSystems 11](#_bookmark15)
  3. [Synthesisand discussion 14](#_bookmark20)

[2.3 Summary 15](#_bookmark22)

[**CHAPTER3:PROJECTREQUIREMENTSANDSYSTEMANALYSIS 16**](#_bookmark23)

* 1. [Introduction 16](#_bookmark24)
  2. [Requirements 16](#_bookmark25)
     1. [*FunctionalRequirements 16*](#_bookmark26)
     2. [*Non-FunctionalRequirements 17*](#_bookmark27)
  3. [SYSTEMANALYSIS 18](#_bookmark28)
     1. [*UseCaseDiagram 18*](#_bookmark29)
     2. [*Flow Chart 19*](#_bookmark32)
  4. [DevelopmentMethodology 20](#_bookmark34)
  5. [SUMMARY 21](#_bookmark36)

[Chapter4:System Design 22](#_Toc184082456)

[**4.1 Introduction** 22](#_Toc184082457)

[**4.2 Architectural design** 22](#_Toc184082458)

[**4.3 Object Oriente dDesign** 22](#_Toc184082459)

[4.1.1 Class Diagram 22](#_Toc184082460)

[4.1.2 SequenceDiagram 24](#_Toc184082461)

[24](#_Toc184082462)

[4.1.3 Activity Diagram 25](#_Toc184082463)

[**4.4 User Interface Design** 26](#_Toc184082464)

[4.4.1 User interface design of CREATE AN ACCOUNT 26](#_Toc184082465)

[4.4.1 User Interface Of Login 26](#_Toc184082466)

[4.4.2 User Interface Of Home 27](#_Toc184082467)

[4.4.3 User Interface Of Blog Posts 29](#_Toc184082468)

[4.4.4 User Interface Of Add new post 30](#_Toc184082469)

[4.4.5 User Interface Of Profile 31](#_Toc184082470)

[4.4.6 User Interface Of About 32](#_Toc184082471)

[4.4.7 User Interface Of Contact 33](#_Toc184082472)

[**4.5 Database Design** 34](#_Toc184082473)

[Chapter 5: Implementation and Results 38](#_Toc184082474)

[**5.1 Introduction** 38](#_Toc184082475)

[**5.2 Tools and Technologies Used** 38](#_Toc184082476)

[**5.3 Implementation** 39](#_Toc184082477)

[5.3.1 **Database Design**: 39](#_Toc184082478)

[5.3.2 **Backend Development:** 40](#_Toc184082479)

[ **Sing Up page code:** 40](#_Toc184082480)

[ **Login page code:** 43](#_Toc184082481)

[ **Mange Blogs:** 45](#_Toc184082482)

[ **Api blogs code:** 53](#_Toc184082483)

[ **Show details blogs:** 57](#_Toc184082484)

[**Show details and save blogs posts to database** 57](#_Toc184082485)

[ **Post Management System** 60](#_Toc184082486)

[**5.3.3 Frontend Integration:** 67](#_Toc184082487)

[**5.4 Testing Case** 80](#_Toc184082488)

[ **Sing up Functionality**: 80](#_Toc184082489)

[ **Login Functionality:** 82](#_Toc184082490)

[**5.5 Results** 83](#_Toc184082491)

[Chapter 6: Conclusion and Future Work 84](#_Toc184082492)

[6.1 Conclusion 84](#_Toc184082493)

[6.2 Limitations 84](#_Toc184082494)

[6.3 Future Work 85](#_Toc184082495)

[References 86](#_Toc184082496)

**Chapter1:Introduction**

* 1. **Introduction**

Traveling serves as a significant form of entertainment for many individuals. Exploring new destinations, experiencing diverse cultures, and learning about civilizations are all integral parts of this experience. As Saudis, we hold a particular passion for travel. Our region boasts remarkable diversity; the North differs from the South, and the East varies from the West. Moreover, each region possesses its unique climate and terrain, adding to the richness of our travel experiences.

* 1. **Problem Definition**

Tourists and residents face several challenges when searching for tourist destinations in SaudiArabia. These difficulties stemfromalack of guidanceregarding thebestproducts, foods, hotels, and tourist attractions. Overall, there is a prevalent issue in Saudi Arabia regardingthedifficultyofintroducingtouriststonewculturesandefficientlyguidingthem to destinations in a concise, helpful, and reliable manner.

In recent years, with the advent of the Internet and mobile phone networks, numerous mobile applications and websites have emerged to assist tourists in the Kingdom of Saudi Arabia. However, most of these electronic solutions fail to fully meet the requirements of travelers. Among the technical solutions that developers should provide to alleviate these challenges is the availability of websites offering a wide range of products, foods, and tourist destinations. These websites should provide comprehensive information about various places that users can search through, including evaluations and prices, ultimately enhancing the overall travel experience in Saudi Arabia.

* 1. **ProjectObjectives**

Ourobjectives inthis projectaretocreateawebsitethatcan:

1. **ProvidingaReliableSource**:Theblogaimstobeatrustedsourcefortravelersseeking information about cities and places visited by others.Visitors can rely on the information providedby othertravelers,as theydescribetheirpersonalexperiences,observations,and explorations of places.
2. **Sharing Experiences and Knowledge:** Different travelers can exchange their experiences andknowledgethroughtheblog.Tips maybeoffered,favoriteplaces shared, and fun activities recommended in various cities. New travelers can benefit from this information to enhance their personal experience and plan their future trips.
3. **Communication and:** The project works on enhancing direct communication among travelers.BalancePeopleinterestedinvisitingspecificplacescanconnectwiththosewho havealreadyvisitedthem.Former travelerscanprovideadvice,guidance,and answerany questions that new visitors may have.
4. **EncouragingExplorationandAdventure**:Theblogcaninspirereaderstoexploreand discover new places. Articles and personal experiences can showcase the spirit of adventure, different cultures, and amazing tourist attractions awaiting visitors around the world**.**
5. **Building an Online Community:** The project can contribute to building an online communityfortravelersandtravelenthusiasts.Thiscommunitycanserveasaplatformfor sharing, collaboration, and exchanging ideas and experiences in the field of travel and tourism.
   1. **ProjectScope**

We aim to create a blog for individuals who travel the world and wish to share their experiences and discoveries from the places they have visited. They will have the opportunity to write articles about the cities, tourist attractions, activities, and restaurants they have enjoyed. Additionally, they can offer ideas and tips for those who plan to visit theseplaces.Readerswillbeabletointeractwiththeauthorsbyaskingquestionsorseeking advice.Thisplatformseemslikeitwillbebeneficialformanyindividualsseekingauthentic experiences and practical information before embarking on their travels.

We also aim to promote global culture and broaden people's horizons by exploring and learning about new places.

* 1. **ProjectTimeline**



**Figure1.1: Project timeline**

* 1. **DocumentOrganization**

The remainder of this document is organized as follows: Chapter 2 will analyze existing websites that serve travel bloggers. In Chapter 3 we will present the system analysis process, which aids in studying the site and its components, and defining its goals. In Chapter 4, we will design our system using class and sequence diagrams, as well as designing some user interfaces for our website.

**Chapter2:LiteratureReview**

* 1. **Introduction**

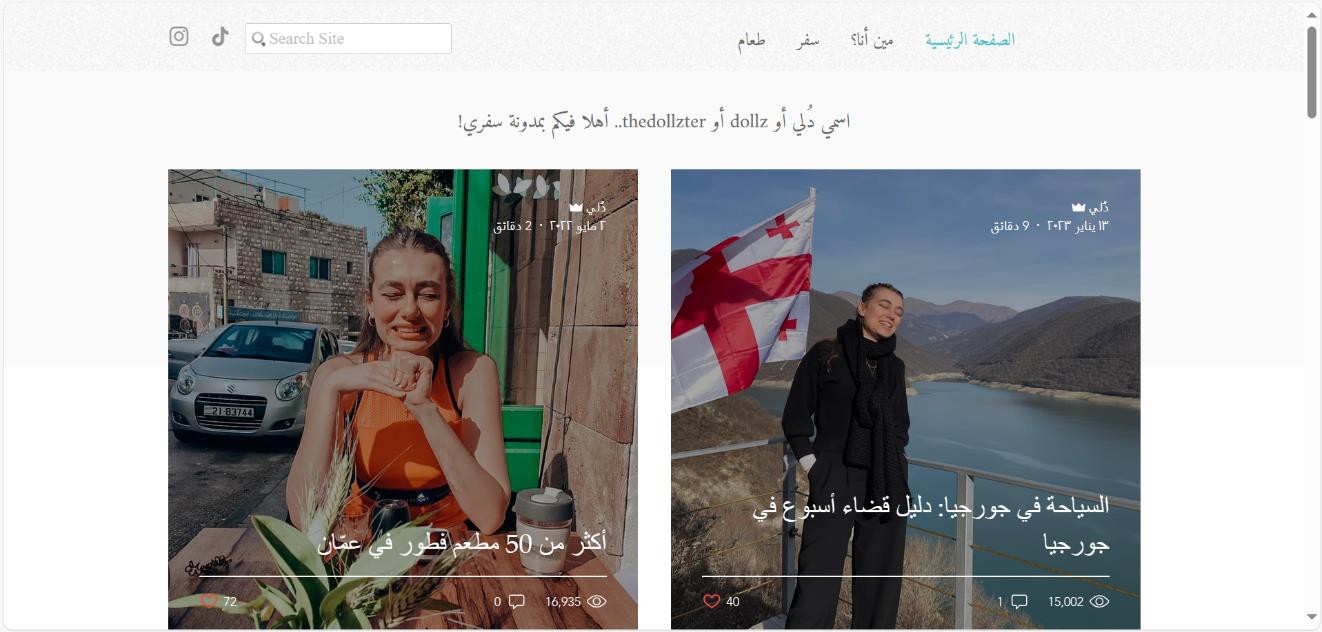
In this chapter, we will evaluate various websites that cater to travel bloggers. We will provide descriptions and analyze the features of each site to capitalize on their strengths and mitigate their weaknesses in the development of our system

* 1. **RelatedworkandSimilarSystems**

In this section, we will showcase some of the best travel blog websites that have been developed in the past to cater to people interested in tourism**.**

#### Thedollzter

ThewebsiteThedollzter**[1]**isownedbyDollzandprimarilyfeaturesherpersonaljourney, traveltips,travelreviews,andtravel-relatedsocialmediacontent.Thiswebsiteallowsusers toaccessfrequentlyaskedquestionsandviewphotosoftheplacesvisitedbyDollz(Figure 2.1).



**Figure2.1: TheprincipalpageofwebsiteThedollzter**

##### Disadvantagesof Thedollzter

Itdoesnotallowuserstoviewotherpeople'sopinionsabouttheblogor their travels.

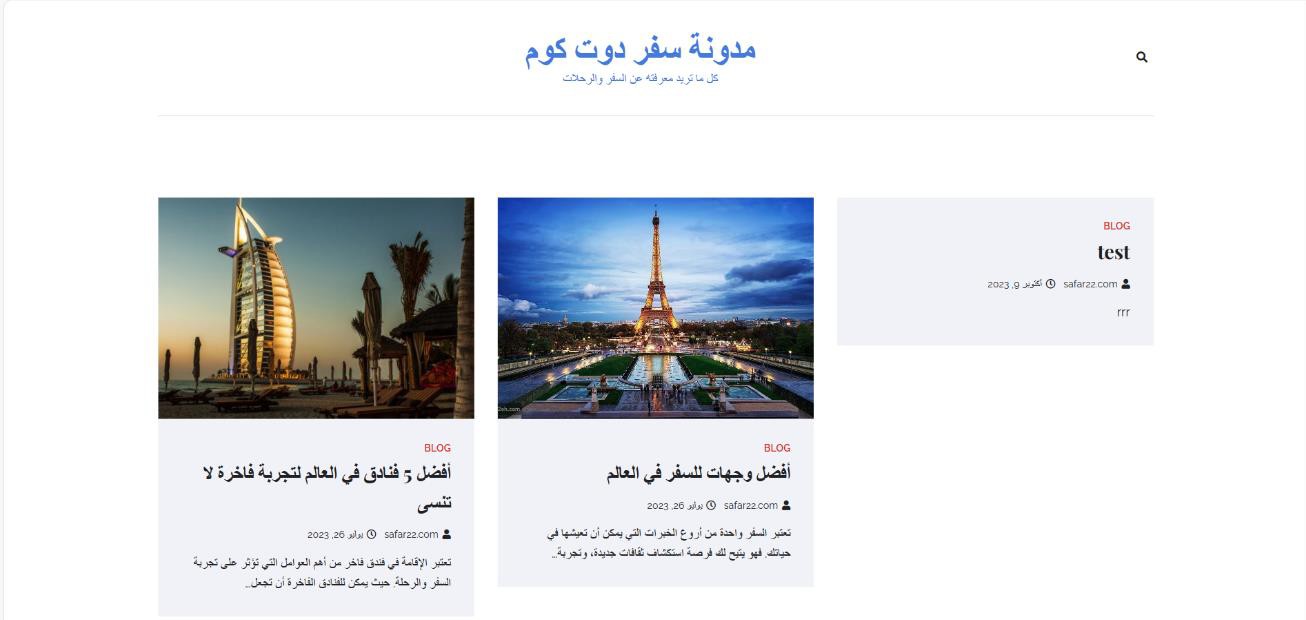
Itdoesnotenableuserstosharetheiropinionsabouttheirtravels.

Itdoes notprovideinformationonpricesinmostareas.

Itdoes notclarifywhethertheseareas aresafeforfamilies ornot.

#### TravelBlog.com

The website travel Blog.com**[2]** provides information about the most renowned touristcities and highlights the essential hotels for a pleasant and comfortable stay (Figure 2.2).



**Figure 2.2: Theprincipalpage of websiteTravelBlog.com**

##### DisadvantagesofTravelBlog.com

Itlacksinformationontravellogisticsandsafetyconsiderations.

Userscannotviewothers'opinionsaboutthis blog.

Itprovidesonlybriefdescriptions oftheareasandaccommodations.

Itdoes notincludepicturesofthetourist cities discussed.

#### Wanderlust's

The website Wanderlust's **[3]** is Charli's travel blog, where she provides tips and informationabouthertravels,alongwithsharingherpersonalexperiences.Onthiswebsite, Charlirespondstoalluserinquiries,interactswithusers,andprovidesdetailedaccountsof theplacesshevisits.Wanderlust'salsoallowsuserstoviewpicturesandsomevideosfrom each of Charli's trips (Figure 2.3).



**Figure2.3: TheprincipalpageofwebsiteWanderlust's**

##### DisadvantagesofWanaderlust's:

Theuserisunabletoaccessinformationregardingcostsandbudgetfor certain places

Theusercannotsharetheirexperience;Charlionly speaksfromherown experience**.**

### Travelhero:

The website **Travel hero** [4] is the first interactive travel guide that allows users to customizetheirtripsbasedontheirmoodandbudgetwiththeassistanceof travelexperts. It enables the user to discuss restaurants, hotels, and potential activities, as well as to upload photos and videos (Figure 2.4).



**Figure 2.4: Theprincipalpage of websiteTravelhero**

##### DisadvantagesofTravelhero:

Theusercannotcommentonblogs.

Thewebsitedoesn'tprovideenoughadvicefortravelers.

* 1. **Synthesisanddiscussion**

Weconductedacomparativeanalysisofothersitesbasedontheobjectivesdescribed in the introduction (Table 2.1).

Based on Table 2.1, we can conclude that some characteristics have considered by allor some websites, such as encouraging exchanging experiences and knowledge, providing reliablesources and quickly search. Other characteristics, such as contains travel in Saudi Arabia, thatbeen taken into account only by Travel Hero website.

**Table2.1:Comparative tableofwebsitesfor blog travel**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **feature** | **Thedollzter** | **Travel Blog.com** | **Wanderlust's** | **Travelhero** |
| **Providing a Reliablesource** | + | \_ | \_ | + |
| **Sharing Experiences**  **and Knowledge** | \_ | \_ | \_ | \_ |
| **Communication and Balance** | \_ | \_ | \_ | \_ |
| **Encouraging Exploration andAdventure** | + | + | + | + |
| **Buildingan Online Community** | \_ | \_ | \_ | \_ |
| **Contains travel inSaudiArabia** | \_ | \_ | \_ | + |
| **Quicklysearch** | \_ | \_ | \_ | + |

**2.3 Summary**

Inthischapter,wehaveintroducedseveralsitesthat hosttravel blogs.Wehaveclassified them and analyzed the features of each in order to leverage their advantages and mitigate their disadvantages in developing our system.

**Chapter3:Project Requirements and System Analysis**

* 1. **Introduction**

In this chapter we will analyze the proposed system through a comprehensive discussion of the feasibility study and functional and non-functional requirements. Additionally, we willanalyzeourwebsite,whichinvolvesstudyingthesiteanditscomponents,anddefining itsgoals,usinguse case andflowchartdiagrams. Finally, wewilldiscussthedevelopment methodology that will be followed to realize the project.

* 1. **Requirements**

This section summarizes the functional and non-functional requirements of the projects’ deliverables. Depending on the nature of the requirements, they will be categorised in to two categories, functional and non-functional.

* + 1. **Functional Requirements**

Functional requirements are a description of the system's behaviour in terms of relevance tothesystem'sfunction.Inoursystem,wehavetwousers:theadminandtheuser,eachof which has its own processes.

##### User

Ourwebsitewillallow userto:

* + Createaccount,
  + Login
  + Updateprofileinformation.
  + Communicatewithcustomerservicewhen needed.
  + Addblog posts
  + Viewblog posts:
    - Viewratingsforeach citywhenselected.
    - Searchforthemostimportant monumentsinthedesiredcity
    - Viewblogpost details
    - Addhisratingforcitiesand reviews
    - Readcomments
    - Addcomments
    - Viewlocationinformationforthe place
    - Likeblogpost
    - SearchforBlogPost

##### Admin

* + Login
  + Manageusers
  + Manageblogposts
  + ModerateComments
  + Respondingtocustomer inquiries.
  + Processingandupdating requirements.
    1. **Non-FunctionalRequirements**

Thenon-functionalrequirementdescribestheperformancecharacteristicofthesystem.We can specify the non-functional requirements of our website through the following points:

##### Accessibilityandavailability:

Allcustomerscanaccessthesite.

##### Capacity:

Theability tocontaina largenumberofdataandalargenumberofuserscan register on the site.

##### Efficiency:

Thesiteisdistinguishedbyitsefficiencybecauseitisanunusualideaandwill solvetheproblemof many peoplewho suffer fromfinding archaeologicaland tourism sites in the Kingdom.

##### Privacyand security:

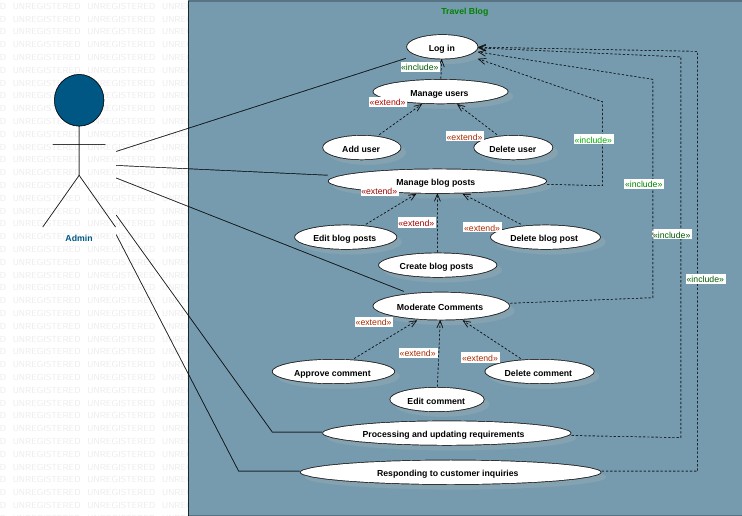
Itoffershighprivacyasalluserdataissecure,and accessisrestrictedto authorized individuals only.

* 1. **System Analysis**

Inthissection,wehighlighthowourwebsiteinteractsandflowsusingavarietyof functions.

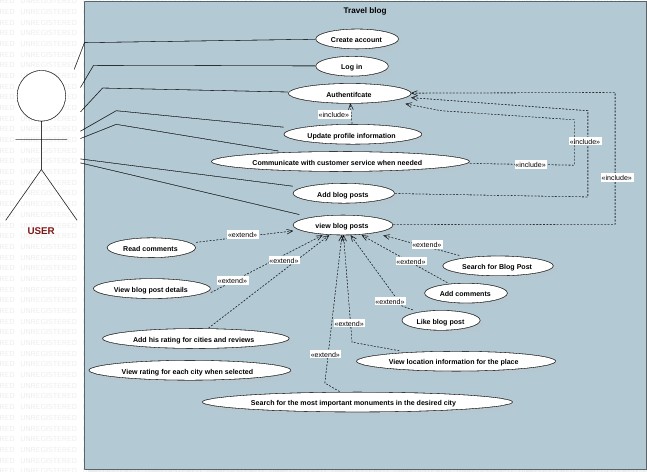
* + 1. **UseCase Diagram**

Thefigurebelow(Figure3.1)illustratestheusecasediagramfortheadminroleonthesite. This role entails administrative access, enabling efficient management of user administration, blog post management, comment moderation, customer query response, and processing and updating of requirements.



**Figure3.1:UsescasediagramforAdmin**

An active user has the ability to create an account, log in, utilize it for multiple purposes, update their account information, create posts, view blog posts, including reading comments,andinteractwithblogcontentbylikingposts,amongotherfunctionalities(refer to Figure 3.2).

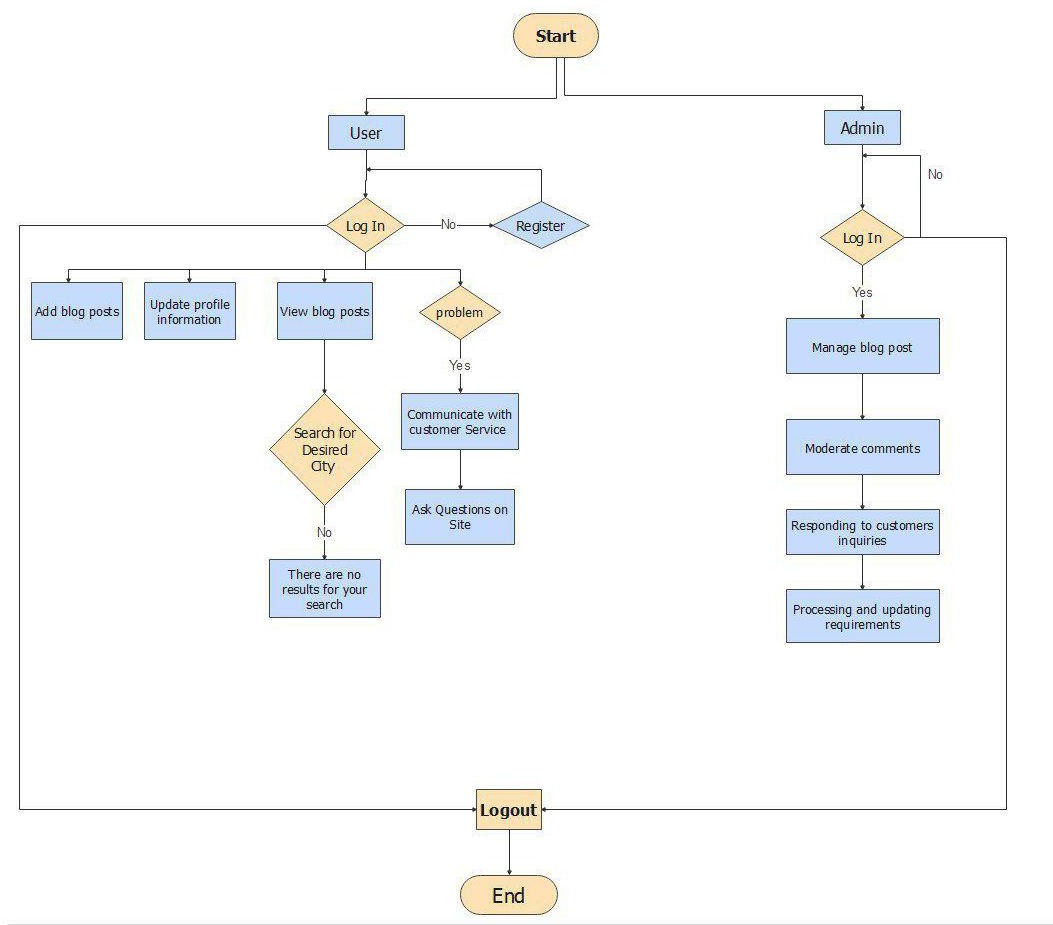


**Figure3.2:uses casediagramforUser**

* + 1. **FlowChart**

Aflowchartisagraphicalrepresentationofaprocess,system,oralgorithm.Itusesvarious symbolsandarrowstoillustratethestepsinvolvedincompletingataskorachievingagoal. Each symbolinaflowchartrepresentsadifferentactionordecisionpoint,andthearrows

indicatetheflowofcontrolordirectionoftheprocess.Thefigurebelow(Figure3.3)show the flowchart of our website.



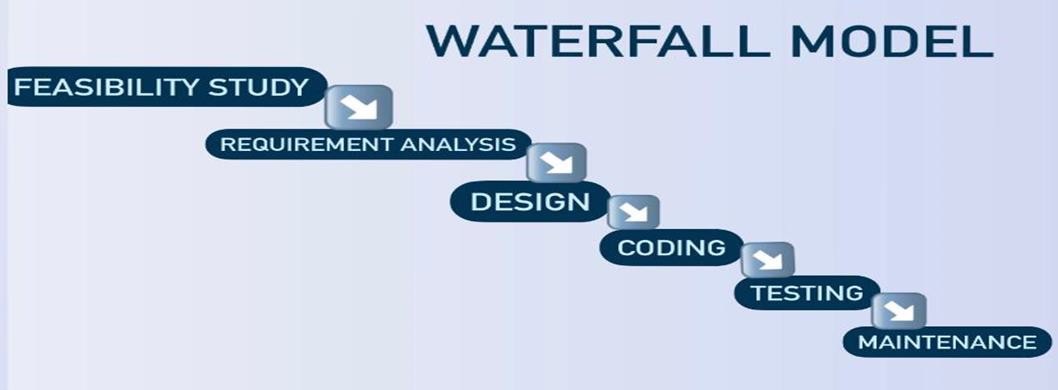
**Figure3.3: FlowChart ofourwebsite**

* 1. **DevelopmentMethodology**

Themethodologyoftheprojectisdifferentandnumerous,includingsixdifferent methodologies, the most common of which is:

* WaterfallModel,
* V-ShapedModel,
* IterativeModel,
* SpiralModel,
* BigBang Model,
* AgileModel.

Inthisproject,werelyonthewaterfall model,which isthesimplestmodeloftheproject's methodology anddependsonsixsimple andeasystepsforall teammembers andhelpsus movebetween thestepswithout any problemsespeciallyfor peoplewhodo nothavehigh experience in developing and managing projects. The Figure 3.4describes the phases of the iterative waterfall.



**Figure 3.4:AdvancedWaterfall model**

* 1. **Summary**

Inthischapterwestudied anddetermined theanalysisofthissystembyspecifying the functional and non-functional requirements.

# Chapter4:System Design

1. **Introduction**

This chapter presents the system design considering the main dimensions of our proposed system. Further, it discusses the various aspects of system design, including object- oriented, Database design, and user interface design.

1. **Architectural design**

We will use a **three-tier architecture** in this project, which consists of three main levels: **the user, the system, and the database**.

**The three levels:**

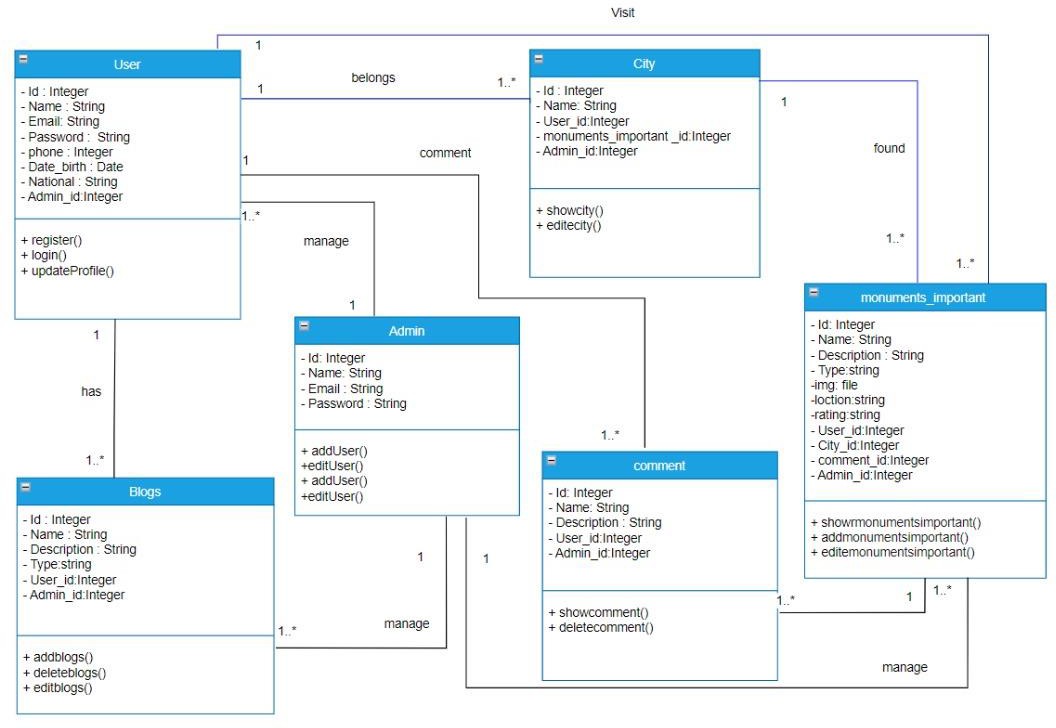
1. **User**: Represents the interface through which the user interacts with the system Using ( HTML, CSS, Javascript )

2. **System**: Comprises the application and processes that manage the interaction between the user and the data Using ( PHP )

3. **Database**: Responsible for storing and managing data Using MySQL

1. **Object Oriente dDesign**

## Class Diagram

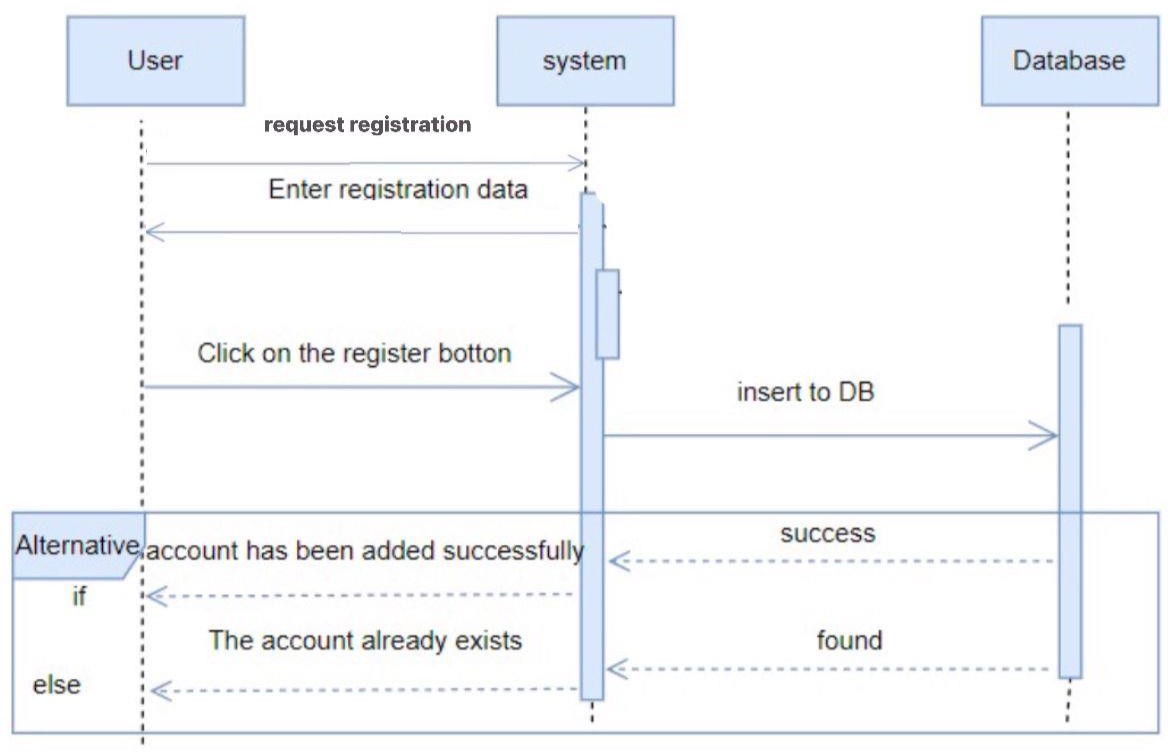


**Figure 4. 1: Class Diagram**

## SequenceDiagram

## 

**Figure4.2:SequenceDiagramforviewBlog**



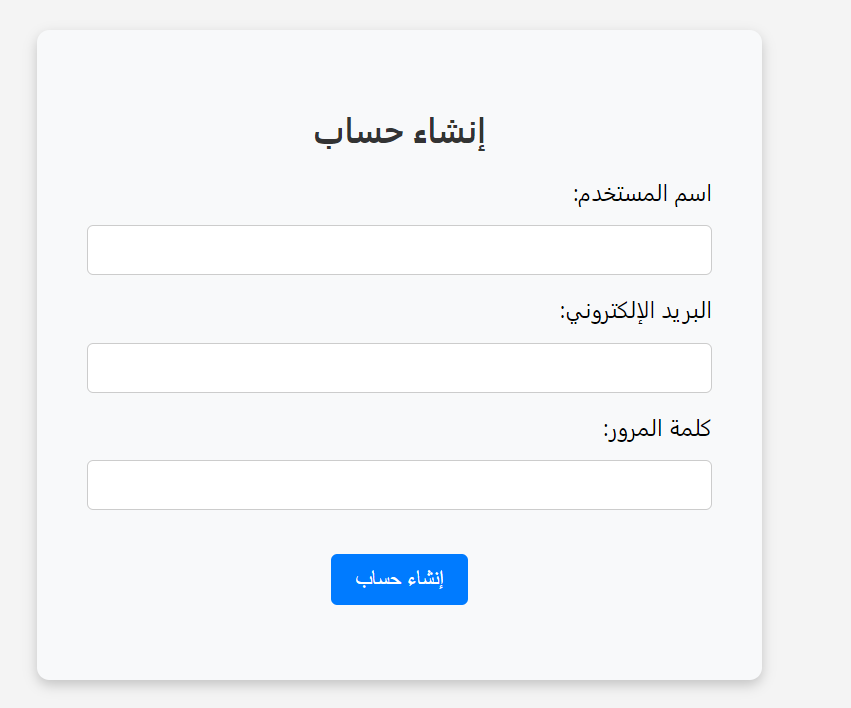
**Figure4.3:Sequencediagramforcreate account.**

## Activity Diagram

1. **User Interface Design**

## User interface design of CREATE AN ACCOUNT

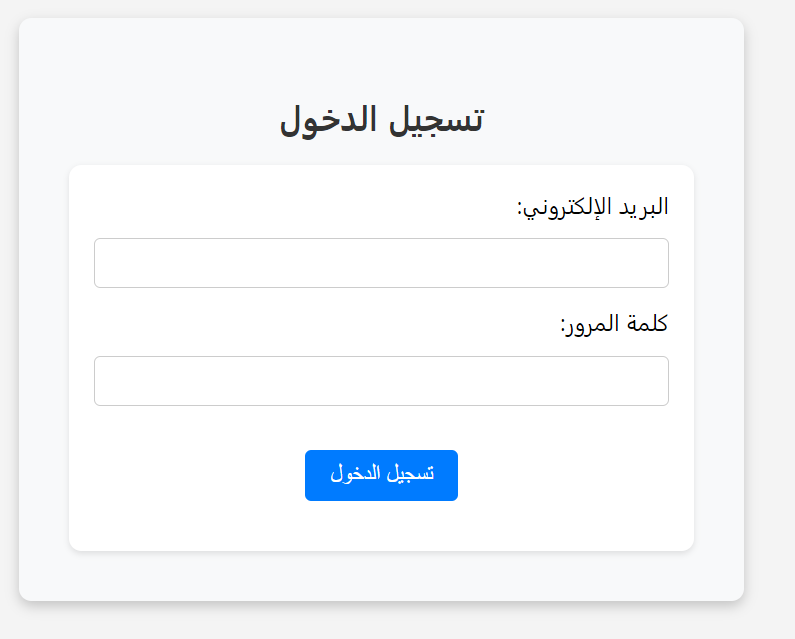
The "Sign Up" section allows new users to register on the website by filling out a simple form. It contains three main input fields: **Username**, **Email**, and **Password**. Each field is required to be filled by the user. After completing the form, the user can click the "Sign Up" button to submit the information to the server for processing.



**Figure4.4: create an account user interface design**

## User Interface Of Login

The login form allows users to enter their **email** and **password** to access their account. It includes two required input fields for the user's email and password, along with a submit button labeled "Login" to send the information for authentication.



**Figure4.5: Log in user interface design**

## User Interface Of Home

The Home Page serves as the main introduction to the website, showcasing its key offerings. It includes:

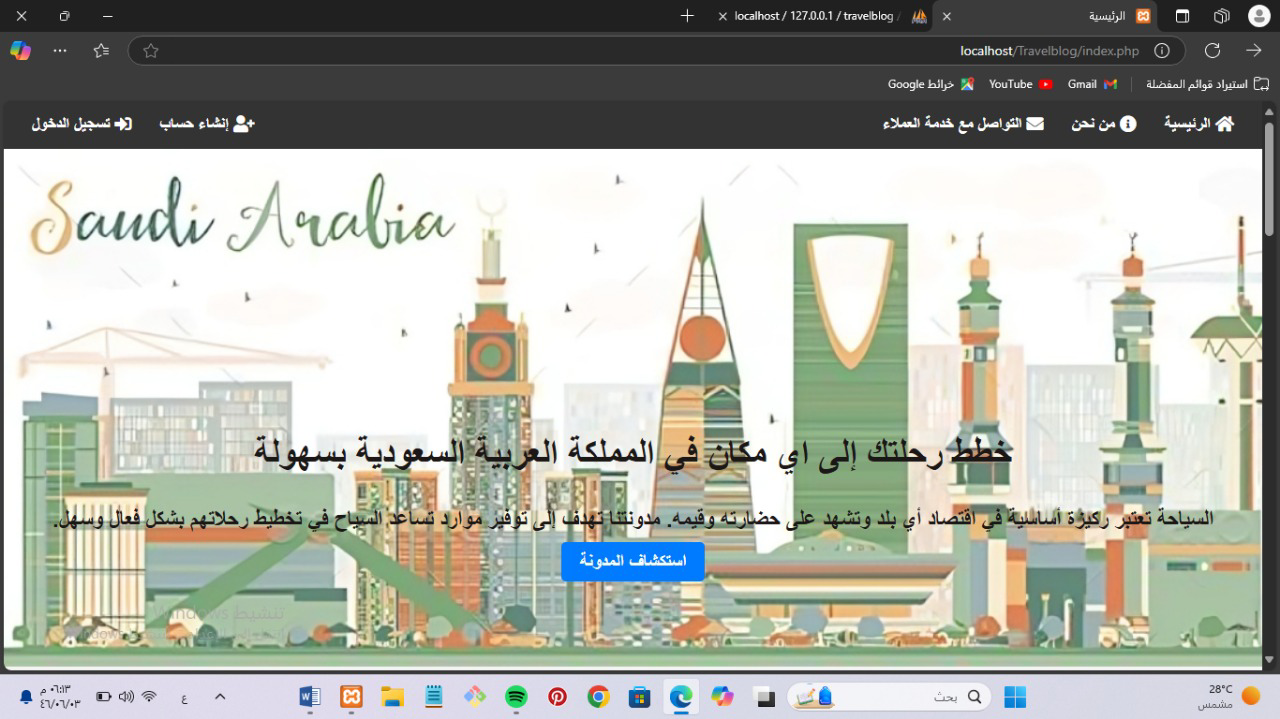
A visually striking banner with an image, headline, and a call-to-action button to explore the blog.

Explains the purpose of the blog—helping users plan trips easily.

Highlights popular tourist spots with images and descriptions.

Displays testimonials from previous visitors, providing credibility.

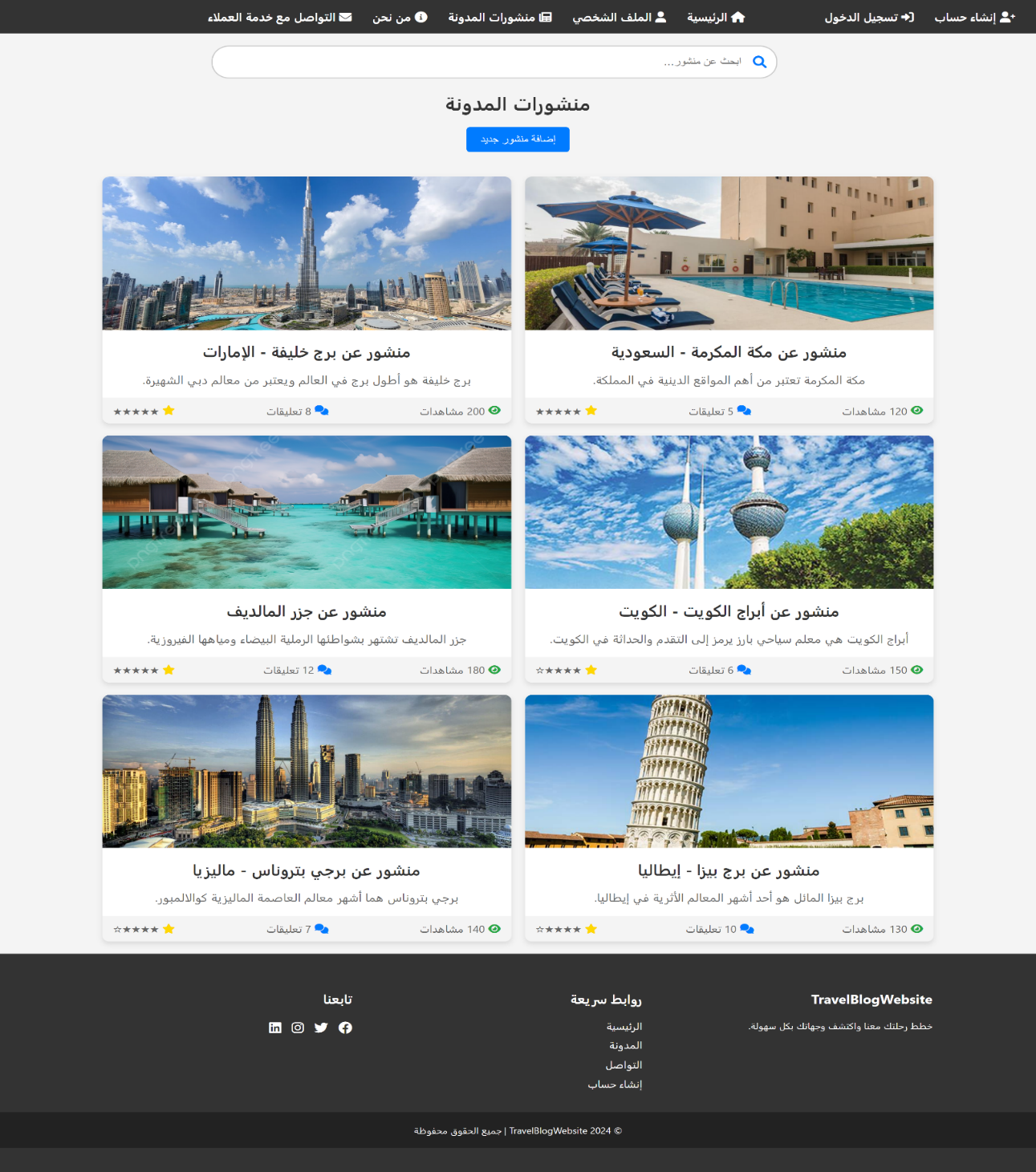
Offers details on the blog’s services like tour guides, hotel bookings, and organized tours.

****

**Figure4.6: Home page user interface design**

## User Interface Of Blog Posts

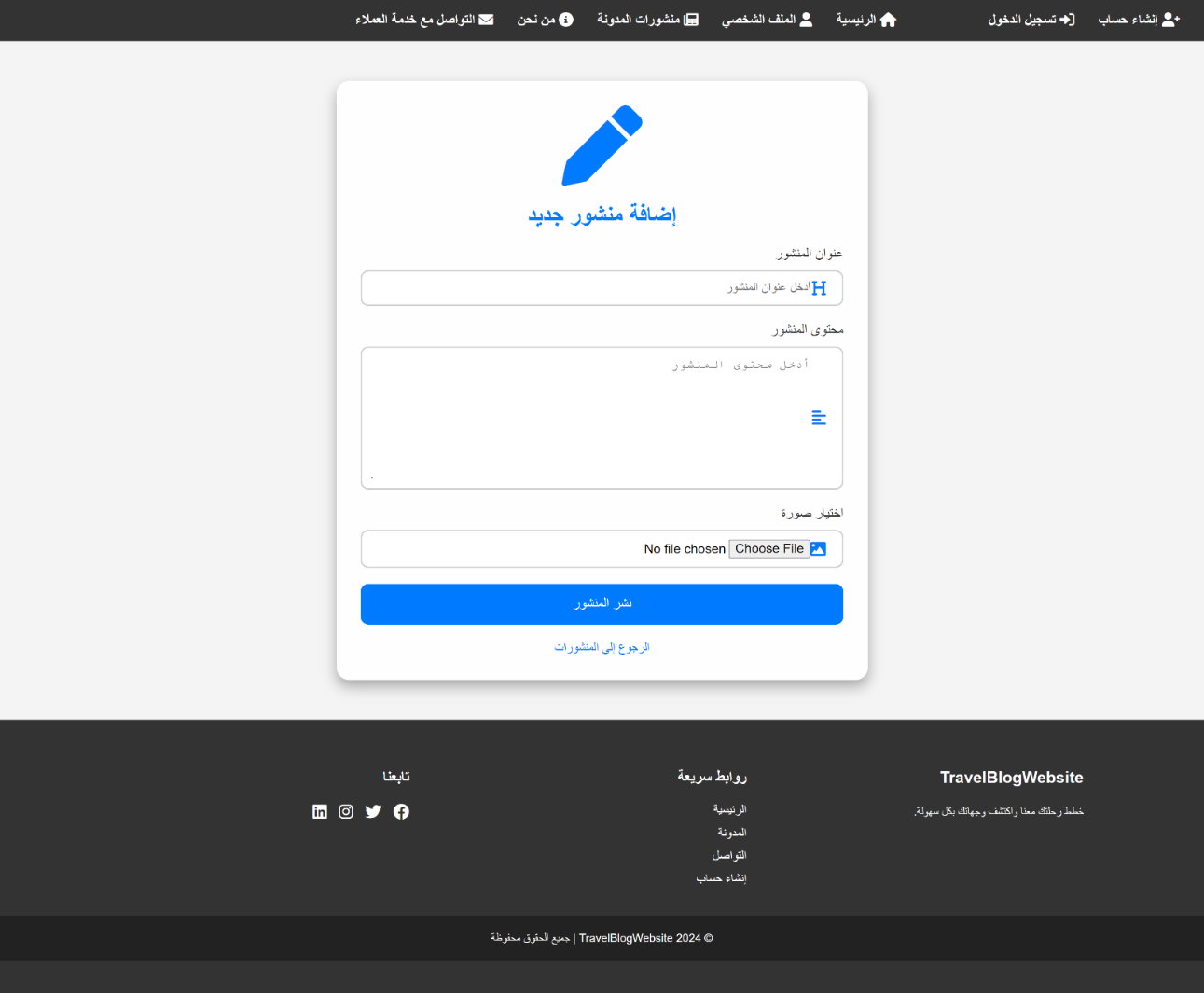
The travel blog website provides users with an engaging platform to explore various travel destinations through dynamic blog posts. It features a search functionality, allowing users to easily find specific content, and encourages interaction by enabling users to create their own posts. The overall design aims to enhance trip planning and foster a community of travel enthusiasts.



**Figure4.7: blog posts user interface design**

## User Interface Of Add new post

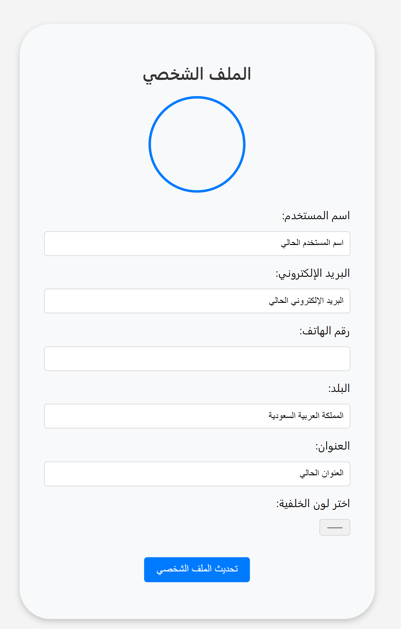
This page represents an interface for adding a new post on a travel blog website. It includes forms for entering the post title and content, along with an option to upload a related image. The page also features a navigation bar that allows access to other pages, such as login and profile, with an appealing design that ensures a comfortable user experience.



**Figure4.8: add new post user interface design**

## User Interface Of Profile

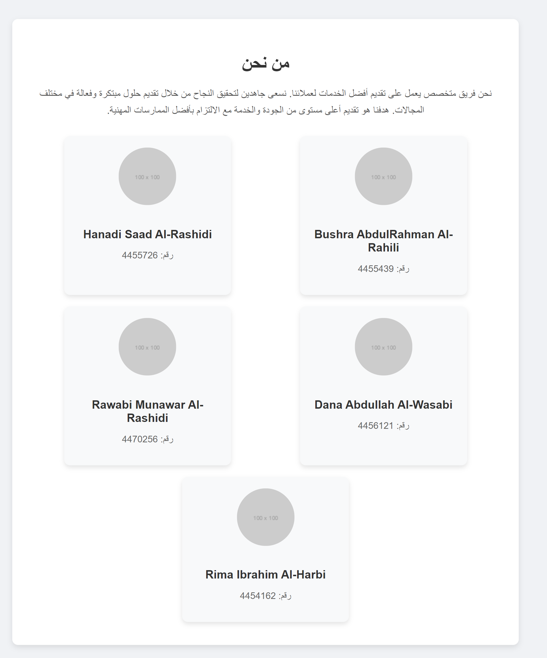
the page is dedicated to the user profile, allowing users to view and update their personal information, including username, email, phone number, country, and address. It features a placeholder image for the user's profile picture and includes a color picker for choosing a background color.



**Figure4.9: profile user interface design**

## User Interface Of About

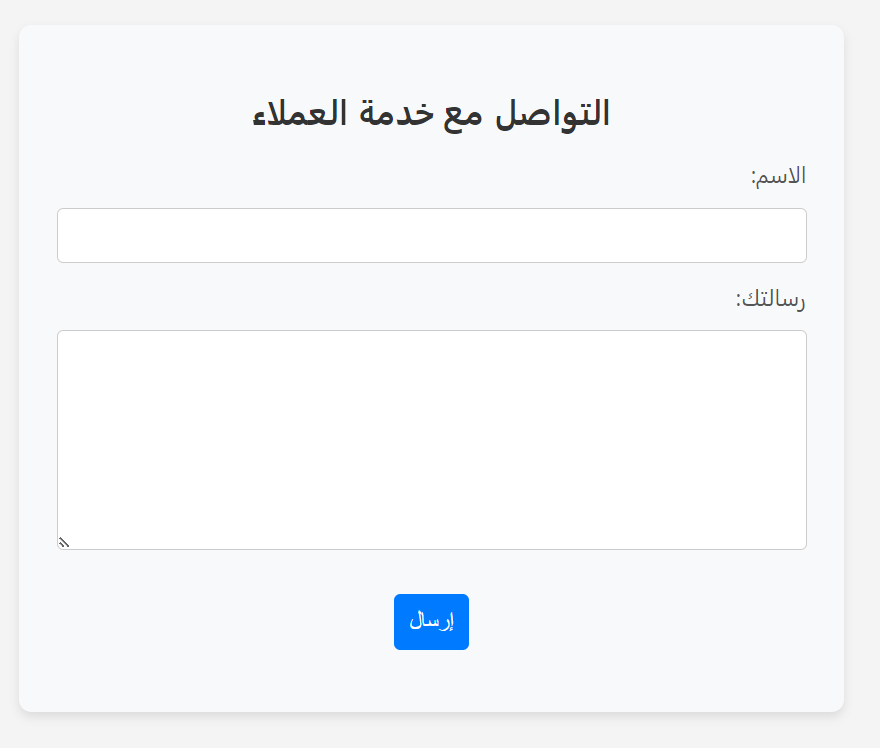
It features a brief description of the team's mission and values, focusing on professionalism and effective service. Below the description, there are team member cards displaying their names, images, and identification numbers, highlighting the individuals who make up the team.



**Fugure4.10: about user interface design**

## User Interface Of Contact

The page provides a straightforward form for users to reach out with inquiries or messages. It includes fields for the user's name and message, both of which are required for submission.



**Fugure4.11: contact user interface design**

1. **Database Design**

**\* User**

*Table 4.1:User table*

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **id** | **integer** | **PK** |
| **created\_at** | **datetime** |  |
| **name** | **string** |  |
| **Email** | **string** |  |
| **password** | **string** |  |
| **phone** | **integer** |  |
| **Date-birth** | **Date** |  |
| **National** | **string** |  |

**\*Admin**

**Table 4.2: Admin table**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **id** | **integer** | **PK** |
| **created\_at** | **datetime** |  |
| **Name** | **string** |  |
| **Email** | **string** |  |
| **password** | **string** |  |
| **phone** | **integer** |  |

**\* Blogs**

*Table 4.3:Blog table*

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **id** | **integer** | **PK** |
| **created\_at** | **datetime** |  |
| **Name** | **string** |  |
| **Description** | **string** |  |
| **Type** | **string** |  |
| **City** | **string** |  |
| **status** | **string** |  |
| **image** | **string** |  |
| **is\_approved** | **bool** |  |
| **User\_id** | **integer** | **FK** |

**\* blog ratings**

**Table 4.4:** *blog\_ratings* **table**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **Id** | **integer** | **PK** |
| **created\_at** | **datetime** |  |
| **blog\_id** | **i**nteger | **FK** |
| **user\_id** | **i**nteger | **FK** |
| **rating** | **double** |  |

**\* Posts**

*Table 4.5:Post table*

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **id** | **integer** | **PK** |
| **created\_at** | **datetime** |  |
| **title** | **string** |  |
| **content** | **string** |  |
| **B**log\_id | **i**nteger | **FK** |
| **user\_id** | **i**nteger | **FK** |
| **status** | **string** |  |

**\* post comments**

**Table 4.6:post Comment table**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **Id** | **integer** | **PK** |
| **comment** | **string** |  |
| **created\_at** | **datetime** |  |
| **post\_id** | **i**nteger | **FK** |
| **user\_id** | **i**nteger | **FK** |

**\* post likes**

*Table 4.7:post* **likes** *table*

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **Id** | **integer** | **PK** |
| **created\_at** | **datetime** |  |
| **post\_id** | **i**nteger | **FK** |
| **user\_id** | **i**nteger | **FK** |

**\* post\_image**

**Table 4.8:monuments\_image table**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **Id** | **integer** | **PK** |
| **p**ost**\_**id | **integer** | **FK** |
| **image\_path** | **string** |  |

**\* Monuments\_Imporant**

*Table 4.9:monuments\_imporant table*

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **Id** | **integer** | **PK** |
| **created\_at** | **datetime** |  |
| **Name** | **string** |  |
| **Description** | **string** |  |
| **Type** | **string** |  |
| **City** | **string** |  |

**\* Monuments\_image**

**Table 4.10:monuments image table**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Rule** |
| **Id** | **integer** | **PK** |
| **Monuments\_Imporant\_**id | **integer** | **FK** |
| **image\_path** | **string** |  |

# Chapter 5: Implementation and Results

1. **Introduction**

This chapter outlines the technology infrastructure, tools, and methodologies used in the development, implementation, and testing phases of the project. It also provides insights into the results achieved and the testing processes followed to validate the system's functionality.

1. **Tools and Technologies Used**

To ensure the successful development of the project, a variety of tools, programming languages, and frameworks were utilized. Below is a summary of the main technologies:

#### ****Programming Languages****

1. **PHP**:
   * Used for backend development, managing server-side operations, and database interactions.
2. **JavaScript**:
   * Applied for front-end scripting to enhance interactivity and provide a dynamic user experience.
3. **HTML/CSS**:
   * Used for building the structure and styling of the web application, ensuring it is visually appealing and responsive.

#### ****Frameworks and Libraries****

1. **PDO (PHP Data Objects)**:
   * Enabled secure database operations with prepared statements to prevent SQL injection attacks.

**Database**

**MySQL**:

* + A robust relational database management system used to store and manage all project data efficiently.

**Development Tools**

1. **XAMPP**:
   * Provided a local server environment for running the application and hosting the MySQL database during development.
2. **Visual Studio Code**:
   * Served as the primary text editor, with support for syntax highlighting and debugging extensions for PHP, JavaScript, and HTML.

**Testing Tools**

1. **Browser Developer Tools**:
   * Utilized for debugging JavaScript and inspecting front-end elements.
2. **Implementation**

The implementation phase involved translating the project's design and requirements into a functional web application. Below are the key steps undertaken during this phase:

1. **Database Design**:

The database schema was designed to include tables for users, posts, comments, and roles. Relationships between tables were defined to maintain data integrity.

Below is an example of the database table schema:

CREATE TABLE `blogs` (

  `id` int(11) NOT NULL,

  `user\_id` int(11) DEFAULT NULL,

  `name` varchar(255) NOT NULL,

  `image` varchar(255) DEFAULT NULL,

  `description` text DEFAULT NULL,

  `type` enum('السياحة الترفيهية','السياحة الثقافية','السياحة الدينية','السياحة الشاطئية','سياحة التأمل','سياحة المغامرات') NOT NULL,

  `city` enum('جده','الرياض','الشرقيه','مكه','المدينه','مناطق الجنوب','مناطق الشمال') NOT NULL,

  `created\_at` timestamp NOT NULL DEFAULT current\_timestamp(),

  `is\_approved` tinyint(1) DEFAULT 0,

  `status` enum('موافق','مرفوض','قيد المراجعة') DEFAULT 'قيد المراجعة'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `blog\_ratings` (

  `id` int(11) NOT NULL,

  `user\_id` int(11) NOT NULL,

  `blog\_id` int(11) NOT NULL,

  `rating` float DEFAULT NULL CHECK (`rating` >= 0 and `rating` <= 5),

  `created\_at` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `places` (

  `id` int(11) NOT NULL,

  `name` varchar(255) NOT NULL,

  `type` enum('السياحة الترفيهية','السياحة الثقافية','السياحة الدينية','السياحة الشاطئية','سياحة التأمل','سياحة المغامرات') NOT NULL,

  `city` enum('جده','الرياض','الشرقيه','مكه','المدينه','مناطق الجنوب','مناطق الشمال') NOT NULL,

  `description` text NOT NULL,

  `created\_at` timestamp NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `posts` (

  `id` int(11) NOT NULL,

  `title` varchar(255) NOT NULL,

  `content` text NOT NULL,

  `user\_id` int(11) NOT NULL,

  `blog\_id` int(11) NOT NULL,

  `created\_at` timestamp NOT NULL DEFAULT current\_timestamp(),

  `status` enum('موافق','مرفوض','قيد المراجعة') DEFAULT 'قيد المراجعة'

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `post\_comments` (

  `id` int(11) NOT NULL,

  `post\_id` int(11) NOT NULL,

  `user\_id` int(11) NOT NULL,

  `comment` text NOT NULL,

  `created\_at` timestamp NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `users` (

  `id` int(11) NOT NULL,

  `full\_name` varchar(255) NOT NULL,

  `email` varchar(255) NOT NULL,

  `password` varchar(255) NOT NULL,

  `phone\_number` varchar(20) DEFAULT NULL,

  `profile\_image` varchar(255) DEFAULT NULL,

  `role` enum('admin','user') DEFAULT 'user',

  `created\_at` timestamp NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

1. **Backend Development:**

PHP was used for server-side logic, including user authentication, form handling, and comment moderation. Security measures, such as input validation and prepared statements, were implemented to protect the system from common vulnerabilities.

* **Sing Up page code:**

<?php

session\_start();

$\_SESSION['title'] = 'إنشاء حساب';

require('header.php');

// تهيئة المتغيرات

$full\_name = $email = $phone= $password = '';

if (isset($\_POST['register'])) {

    $full\_name = trim($\_POST['full\_name']);

    $email = trim($\_POST['email']);

    $phone = trim($\_POST['phone']);

    $password = trim($\_POST['password']);

    if (!empty($\_POST["remember"])) {

        setcookie("email", $email, time() + 3600);

        setcookie("password", $password, time() + 3600);

    }

    $errors\_r = array();

    if (empty($full\_name)) {

        $errors\_r['full\_name'] = "يرجى إدخال الاسم الثلاثي.";

    } else if (!preg\_match("/^[\p{Arabic}a-zA-Z\s]+$/u", $full\_name)) {

        $errors\_r['full\_name'] = "يجب أن يحتوي الاسم على أحرف فقط.";

    }

    // التحقق من رقم الجوال

    if (empty($phone)) {

        $errors\_r['phone'] = "يرجى إدخال رقم الجوال.";

    } else if (!preg\_match("/^\d+$/", $phone)) {

        $errors\_r['phone'] = "يجب أن يحتوي رقم الجوال على أرقام فقط.";

    }

    // التحقق من البريد الإلكتروني

    if (empty($email)) {

        $errors\_r['email'] = "يرجى إدخال البريد الإلكتروني.";

    } else {

        if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

            $errors\_r['email'] = "يرجى إدخال بريد إلكتروني صحيح.";

        } else if (strlen($email) > 200) {

            $errors\_r['email'] = "يجب ألا يتجاوز البريد الإلكتروني 200 حرف.";

        } else {

            // التحقق من فريدية البريد الإلكتروني

            $q = $con->prepare('SELECT \* FROM Users WHERE email = :email');

            $q->execute(array("email" => $email));

            if ($q->rowCount()) {

                $errors\_r['email'] = "البريد الإلكتروني مسجل بالفعل.";

            }

        }

    }

    // شروط كلمة المرور

    if (empty($password)) {

        $errors\_r['password'] = "يرجى إدخال كلمة المرور.";

    } else {

        if (strlen($password) < 8) {

            $errors\_r['password'] = "يجب أن تكون كلمة المرور على الأقل 8 أحرف.";

        } else if (strlen($password) > 200) {

            $errors\_r['password'] = "يجب ألا تتجاوز كلمة المرور 200 حرف.";

        }

    }

    // إذا لم توجد أخطاء

    if (empty($errors\_r)) {

        $sql = "INSERT INTO Users (full\_name, email, password, role,phone\_number) VALUES (:x1, :x2, :x3, :x4, :x5)";

        $q = $con->prepare($sql);

        $q->execute(array(

            "x1" => $full\_name,

            "x2" => $email,

            "x3" => $password,

            "x5" => $phone,

            "x4" => 'user'

        ));

        if ($q->rowCount()) {

            $q = $con->prepare('SELECT \* FROM Users WHERE id = :id');

            $q->execute(array("id" => $con->lastInsertId()));

            $user = $q->fetch(PDO::FETCH\_ASSOC);

            if ($user) {

                $\_SESSION['auth'] = $user;

                header("LOCATION:index.php");

                exit();

            }

        }

    }

}

?>

<body class="body-auth">

    <?php require('navbar.php'); ?>

    <section id="signup" class="signup-section">

        <h2 class="signup-title">إنشاء حساب</h2>

        <form id="signupForm" class="signup-form" method="POST">

            <label for="username">الاسم الثلاثي:</label>

            <input type="text" placeholder="الاسم الثلاثي" value="<?php echo htmlspecialchars($full\_name); ?>" id="full\_name" name="full\_name" required>

            <?php if (isset($errors\_r['full\_name'])) { echo "<p class='error-message'>".$errors\_r['full\_name']."</p>"; } ?>

            <label for="username">الجوال:</label>

            <input type="number" placeholder="رقم الجوال" value="<?php echo htmlspecialchars($phone); ?>" id="phone" name="phone" required>

            <?php if (isset($errors\_r['phone'])) { echo "<p class='error-message'>".$errors\_r['phone']."</p>"; } ?>

            <label for="email">البريد الإلكتروني:</label>

            <input type="email" placeholder="البريد الالكتروني" value="<?php echo htmlspecialchars($email); ?>" name="email" id="email" required>

            <?php if (isset($errors\_r['email'])) { echo "<p class='error-message'>".$errors\_r['email']."</p>"; } ?>

            <label for="password">كلمة المرور:</label>

            <input type="password" placeholder="كلمة المرور" value="<?php echo htmlspecialchars($password); ?>" id="password" name="password" required>

            <?php if (isset($errors\_r['password'])) { echo "<p class='error-message'>".$errors\_r['password']."</p>"; } ?>

            <div class="top">

                <span>هل لديك حساب؟ <a href="login.php">تسجيل الدخول</a></span>

                <div class="one">

                    <input type="checkbox" id="login-check" name="remember">

                    <label for="login-check">تذكرني</label>

                </div>

            </div>

            <button type="submit" name="register" class="signup-button">إنشاء حساب</button>

        </form>

    </section>

</body>

</html>

* **Login page code:**

<?php

session\_start();

$\_SESSION['title'] = 'تسجيل الدخول';

require('header.php');

// التحقق من وجود الـ cookies للبريد الإلكتروني وكلمة المرور

$email = $password = '';

if (isset($\_COOKIE["email"])) {

    $email = htmlspecialchars($\_COOKIE["email"]);

}

if (isset($\_COOKIE["password"])) {

    $password = htmlspecialchars($\_COOKIE["password"]);

}

if (isset($\_POST['login'])) {

    $email = trim($\_POST['email']);

    $password = trim($\_POST['password']);

    if (!empty($\_POST["remember"])) {

        setcookie("email", $email, time() + 3600);

        setcookie("password", $password, time() + 3600);

    } else {

        // إذا لم يتم تحديد "تذكرني"، قم بحذف الـ cookies

        if (isset($\_COOKIE["email"])) {

            setcookie("email", "", time() - 3600);

        }

        if (isset($\_COOKIE["password"])) {

            setcookie("password", "", time() - 3600);

        }

    }

    // التحقق من الحقول

    $errors = [];

    // التحقق من البريد الإلكتروني

    if (empty($email)) {

        $errors['email'] = "يرجى إدخال البريد الإلكتروني.";

    }

    // التحقق من كلمة المرور

    if (empty($password)) {

        $errors['password'] = "يرجى إدخال كلمة المرور.";

    }

    // إذا لم توجد أخطاء

    if (empty($errors)) {

        $sql = "SELECT \* FROM Users WHERE email = :email";

        $q = $con->prepare($sql);

        $q->execute(['email' => $email]);

        $user = $q->fetch(PDO::FETCH\_ASSOC);

        if ($user) {

            if ($password === $user['password']) {

                $\_SESSION['auth'] = $user;

                redirct\_login();

                exit();

            } else {

                $errors['password'] = "كلمة المرور غير صحيحة.";

            }

        } else {

            $errors['email'] = "البريد الإلكتروني غير صحيح.";

        }

    }

}

?>

<body class="body-auth">

    <?php require('navbar.php'); ?>

    <section id="login" class="login-section">

        <h2 class="login-title">تسجيل الدخول</h2>

        <div class="login-card-container">

            <div class="login-card">

                <form id="loginForm" method="POST">

                    <label for="loginEmail">البريد الإلكتروني:</label>

                    <input type="email" value="<?php echo $email; ?>" id="loginEmail" name="email" required>

                    <?php

                        if(isset($errors['email'])){

                            echo "<p class='error-message'>".$errors['email']."</p>";

                        }

                    ?>

                    <label for="loginPassword">كلمة المرور:</label>

                    <input type="password" value="<?php echo $password; ?>" id="password" name="password" required>

                    <?php

                        if(isset($errors['password'])){

                            echo "<p class='error-message'>".$errors['password']."</p>";

                        }

                    ?>

                    <button type="submit" name="login" class="login-button">تسجيل الدخول</button>

                </form>

            </div>

        </div>

    </section>

</body>

</html>

* **Mange Blogs:**

<?php

session\_start();

$\_SESSION['title'] = 'المدونات';

require('header.php');

// تهيئة المتغيرات

$dirPath = 'uploads/blogs';

$name = $image = $description =$city=$blog\_type= '';

$form\_visible = false;

$allowed\_extension = array('jpg', 'jpeg', 'png', 'gif');

$errors = [];

$blog\_id='';

$action = "create";

if (isset($\_POST['blogs'])) {

    $action = trim($\_POST['action']);

    $name = trim($\_POST['name']);

    $description = trim($\_POST['description']);

    $blog\_type = trim($\_POST['type']);

    $city = trim($\_POST['city']);

    // التحقق من الأخطاء

    if (empty($name)) {

        $errors["name"] = 'الاسم مطلوب.';

    }

    if (empty($description)) {

        $errors["description"] = 'الوصف مطلوب.';

    }

    if (empty($blog\_type)) {

        $errors['type'] = 'نوع المدونة مطلوب.';

    }

    if (empty($city)) {

        $errors['city'] = 'المدينة مطلوبة.';

    }

    if (!isset($\_SESSION['auth'])) {

        $errors['user'] = "يجب تسجيل الدخول أولاً";

        exit;

    }

    else {

        $user\_id = $\_SESSION['auth']['id'];

        // جلب تفاصيل الرحلة للتحقق من الشروط

        $stmt = $con->prepare("SELECT id FROM users WHERE id = :user\_id");

        $stmt->execute(array(

            'user\_id' => $user\_id,

        ));

        $user = $stmt->fetch(PDO::FETCH\_ASSOC);

        if (!$user) {

            $errors['user'] = "المستخدم غير موجود";

        }

    }

    if($action == "create"){

        if(isset($\_FILES['image'])){

            $file =$\_FILES['image'];

            // file conditions

            $file\_name = $file['name'];

            $file\_tmp = $file['tmp\_name'];

            $file\_error = $file['error'];

            $file\_size = $file['size'];

            $file\_extension = pathinfo($file\_name, PATHINFO\_EXTENSION);

            // if photo is null

            if(empty($file\_name)){

                $errors['image'] = "من فضلك اختر صورة للمدونة";

            }else{

                // if photo not in allowed extensions

                if(!in\_array($file\_extension,$allowed\_extension)){

                    $errors['image'] = "الملفات المسموح بها هي الصور التي من نوع " . implode(',',$allowed\_extension);

                } else if($file\_size >3000000){

                    $errors['image'] = "اعلى حجم مسموح به لارسال ملف هو 3 ميجا";

                }

            }

        }else{

            $errors['image'] = "من فضلك اختر صورة للمدونة";

        }

        $stmt = $con->prepare("SELECT id FROM blogs WHERE user\_id = :user\_id LIMIT 1");

        $stmt->execute(['user\_id' => $user\_id]);

        $existing\_blog = $stmt->fetch(PDO::FETCH\_ASSOC);

        if ($existing\_blog) {

            $errors['user'] = "لقد قمت بنشر مدونة بالفعل.";

        }

        if (!empty($errors)) {

            $form\_visible = true;

        }else{

            if (!is\_dir($dirPath)) {

                mkdir($dirPath, 0777, true);

            }

            // upload file

            $file\_path = $dirPath."/".uniqid('file\_', true) . '.' . $file\_extension;

            if(move\_uploaded\_file($file\_tmp,$file\_path)){

                // success upload

            }

            else{

                // faild to upload

                errorJsonResponse("فشل في رفع الملف");

            }

            $sql = "INSERT INTO blogs (city, type, name, description, image, user\_id) VALUES (:x1, :x2, :x3, :x4, :x5, :x6)";

            $q = $con->prepare($sql);

            $q->execute(array(

                "x1" => $city,

                "x2" => $blog\_type,

                "x3" => $name,

                "x4" => $description,

                "x5" => $file\_path,

                "x6" => $user\_id,

            ));

            $form\_visible = false;

            $name = $image = $description =$city=$blog\_type= '';

            $action = "create";

            $blog\_id='';

        }

    }else if($action == "update"){

        $action = "update";

        $blog\_id = $\_POST['id'];

        // استعلام لجلب بيانات المدونة الحالية

        $stmt = $con->prepare("SELECT user\_id, image FROM blogs WHERE id = :id");

        $stmt->execute(['id' => $blog\_id]);

        $existing\_blog = $stmt->fetch(PDO::FETCH\_ASSOC);

        if (!$existing\_blog) {

            // المدونة غير موجودة

            $errors["error"]= "المدونة غير موجودة.";

        }

        if ($existing\_blog['user\_id'] != $\_SESSION['auth']['id']) {

            // المستخدم ليس صاحب المدونة

            $errors["error"]= "ليس لديك الصلاحية لتعديل هذه المدونة.";

        }

        if(isset($\_FILES['image'])){

            $file =$\_FILES['image'];

            // file conditions

            $file\_name = $file['name'];

            $file\_tmp = $file['tmp\_name'];

            $file\_error = $file['error'];

            $file\_size = $file['size'];

            $file\_extension = pathinfo($file\_name, PATHINFO\_EXTENSION);

            // if photo is null

            if(empty($file\_name)){

                $errors['image'] = "من فضلك اختر صورة للمدونة";

            }else{

                // if photo not in allowed extensions

                if(!in\_array($file\_extension,$allowed\_extension)){

                    $errors['image'] = "الملفات المسموح بها هي الصور التي من نوع " . implode(',',$allowed\_extension);

                } else if($file\_size >3000000){

                    $errors['image'] = "اعلى حجم مسموح به لارسال ملف هو 3 ميجا";

                }

            }

        }

        $executeParams = array(

            "x1" => $name,

            "x2" => $blog\_type,

            "x3" => $city,

            "x4" => $description,

            "id" => $blog\_id

        );

        if(empty($errors) && isset($file\_name) && !empty($file\_tmp)){

            if (!is\_dir($dirPath)) {

                mkdir($dirPath, 0777, true);

            }

            // upload file

            $file\_path = $dirPath."/".uniqid('file\_', true) . '.' . $file\_extension;

            if(move\_uploaded\_file($file\_tmp,$file\_path)){

                // success upload

                //  حذف الصورة القديمة إذا كانت موجودة

                if(!empty($existing\_blog['image']) && file\_exists($existing\_blog['image'])){

                    unlink($existing\_blog['image']);

                }

            }

            else{

                // faild to upload

                errorJsonResponse("فشل في رفع الملف");

            }

            $sql = "UPDATE blogs SET name = :x1, type = :x2, city = :x3, description = :x4, image=:x5 WHERE id = :id";

            $executeParams['x5'] = $file\_path;

        } else {

            $sql = "UPDATE blogs SET name = :x1, type = :x2, city = :x3, description = :x4 WHERE id = :id";

        }

        if(empty($errors)){

            $q = $con -> prepare($sql);

            $q->execute($executeParams);

        }

        $form\_visible = false;

        $name = $image = $description =$city=$blog\_type= '';

        $action = "create";

        $blog\_id='';

    }

}

$hide\_button = false;

if (isset($\_SESSION['auth'])) {

    $user\_id = $\_SESSION['auth']['id'];

    // التحقق إذا كان لدى المستخدم مدونة

    $stmt = $con->prepare("SELECT id FROM blogs WHERE user\_id = :user\_id LIMIT 1");

    $stmt->execute(['user\_id' => $user\_id]);

    $existing\_blog = $stmt->fetch(PDO::FETCH\_ASSOC);

    // تحديد ما إذا كان الزر يجب أن يظهر

    $hide\_button = false;

    if ($existing\_blog) {

        $hide\_button = true; // إذا كان لديه مدونة، أخف الزر

    }

}

?>

<body>

<?php require('navbar.php'); ?>

    <section id="posts">

        <div id="form-container" style="<?php echo $form\_visible ? 'display: block;' : 'display: none;'; ?>">

            <div class="form-container">

                <form method="post" id="form\_id" enctype="multipart/form-data">

                <?php if (isset($errors['user'])) echo "<p class='error-message'>" . $errors['user'] . "</p>"; ?>

                    <input name="action" value="<?php echo isset($action) ? htmlspecialchars($action) : ''; ?>" type="hidden">

                    <input name="id"  value="<?php echo isset($blog\_id) ? htmlspecialchars($blog\_id) : ''; ?>" type="hidden">

                    <?php if (isset($errors['error'])) echo "<p class='error-message'>" . $errors['error'] . "</p>"; ?>

                    <div class="form-layout">

                        <div class="form-group-container">

                            <div class="form-group">

                                <input

                                    name="name"

                                    id="name"

                                    value="<?php echo htmlspecialchars($name); ?>"

                                    class="input-fieldf "

                                    placeholder="اسم المدونة" />

                                <?php if (isset($errors['name'])) echo "<p class='error-message'>" . $errors['name'] . "</p>"; ?>

                            </div>

                            <!-- حقل نوع المدونة -->

                            <div class="form-group">

                                <select class="select-form2 select-f" name="type" id="type" required>

                                    <option value="">اختر نوع المدونة</option>

                                    <option value="السياحة الترفيهية" <?php if ($blog\_type == "السياحة الترفيهية") echo "selected"; ?>>السياحة الترفيهية</option>

                                    <option value="السياحة الثقافية" <?php if ($blog\_type == "السياحة الثقافية") echo "selected"; ?>>السياحة الثقافية</option>

                                    <option value="السياحة الدينية" <?php if ($blog\_type == "السياحة الدينية") echo "selected"; ?>>السياحة الدينية</option>

                                    <option value="السياحة الشاطئية" <?php if ($blog\_type == "السياحة الشاطئية") echo "selected"; ?>>السياحة الشاطئية</option>

                                    <option value="سياحة التأمل" <?php if ($blog\_type == "سياحة التأمل") echo "selected"; ?>>سياحة التأمل</option>

                                    <option value="سياحة المغامرات" <?php if ($blog\_type == "سياحة المغامرات") echo "selected"; ?>>سياحة المغامرات</option>

                                </select>

                                <?php if (isset($errors['type'])) echo "<p class='error-message'>" . $errors['type'] . "</p>"; ?>

                            </div>

                            <!-- حقل المدينة -->

                            <div class="form-group">

                                <select class="select-form2 select-f" name="city" id="city" required>

                                    <option value="">اختر المدينة</option>

                                    <option value="جده" <?php if ($city == "جده") echo "selected"; ?>>جده</option>

                                    <option value="الرياض" <?php if ($city == "الرياض") echo "selected"; ?>>الرياض</option>

                                    <option value="الشرقيه" <?php if ($city == "الشرقيه") echo "selected"; ?>>الشرقيه</option>

                                    <option value="مكه" <?php if ($city == "مكه") echo "selected"; ?>>مكه</option>

                                    <option value="المدينه" <?php if ($city == "المدينه") echo "selected"; ?>>المدينه</option>

                                    <option value="مناطق الجنوب" <?php if ($city == "مناطق الجنوب") echo "selected"; ?>>مناطق الجنوب</option>

                                    <option value="مناطق الشمال" <?php if ($city == "مناطق الشمال") echo "selected"; ?>>مناطق الشمال</option>

                                </select>

                                <?php if (isset($errors['city'])) echo "<p class='error-message'>" . $errors['city'] . "</p>"; ?>

                            </div>

                        </div>

                        <!-- حقل تحميل الصورة -->

                        <div class="file-input-container">

                            <input type="file" class="input-fiele" name="image" id="image" accept="image/\*">

                            <label for="image" class="file-label">

                                <i id="imageIcon" class="fa fa-image"></i>

                                <img id="imagePreview" src="" alt="Image Preview" class="image-preview">

                            </label>

                            <?php if (isset($errors['image'])) echo "<p class='error-message'>" . $errors['image'] . "</p>"; ?>

                        </div>

                    </div>

                    <div class="form-group full-width-description">

                    <textarea

                        name="description"

                        id="description"

                        class="textarea-fieldf"

                        placeholder="Enter Description"

                        rows="5"><?php echo htmlspecialchars($description); ?></textarea>

                    <?php if (isset($errors['description'])) echo "<p class='error-message'>" . $errors['description'] . "</p>"; ?>

                </div>

                    <div class="button-group">

                        <button class="btn btn-save" name='blogs' type="submit">حفظ</button>

                        <button class="btn btn-close" type="button" id="close\_form">إغلاق</button>

                    </div>

                </form>

            </div>

        </div>

        <!-- الحاوية الخاصة بجدول المنشورات -->

        <div class="container" id="table-container" style="<?php echo !$form\_visible ? 'display: block;' : 'display: none;'; ?>">

            <div class="search-container">

                <div class="search-box">

                    <input type="text" id="search" name="search" placeholder="ابحث عن مدونة...">

                    <i class="fas fa-search search-icon"></i>

                </div>

                <div class="filter-form">

                <select class="select-form2" id="filter-city">

                    <option value="">اختر المدينة</option>

                    <option value="جده">جده</option>

                    <option value="الرياض">الرياض</option>

                    <option value="الشرقيه">الشرقيه</option>

                    <option value="مكه">مكه</option>

                    <option value="المدينه">المدينه</option>

                    <option value="مناطق الجنوب">مناطق الجنوب</option>

                    <option value="مناطق الشمال">مناطق الشمال</option>

                </select>

                <select class="select-form2" id="filter-type">

                    <option value="">اختر النوع</option>

                    <option value="السياحة الترفيهية">السياحة الترفيهية</option>

                    <option value="السياحة الثقافية">السياحة الثقافية</option>

                    <option value="السياحة الدينية">السياحة الدينية</option>

                    <option value="السياحة الشاطئية">السياحة الشاطئية</option>

                    <option value="سياحة التأمل">سياحة التأمل</option>

                    <option value="سياحة المغامرات">سياحة المغامرات</option>

                </select>

                <button class="btn btn-primary" id="filter-data">فلتر</button>

                </div>

                <?php if (!$hide\_button): ?>

                <button class="btn btn-primary" id="add\_new">إضافة مدونة جديدة</button>

                <?php endif; ?>

            </div>

            <div class="card-container blogs-grid">

            </div>

        </div>

    </section>

</body>

</html>

* **Api blogs code:**

This PHP code handles several actions related to managing blogs. It is triggered by AJAX requests and allows the admin and users to perform tasks such as viewing, approving, searching, and deleting blogs, as well as retrieving blog ratings.

<?php

require('../database/db.php');

session\_start();

require('../helper.php');

if (isset($\_POST['action'])) {

    switch ($\_POST['action']) {

        case 'get':

            // استعلام للحصول على بيانات المعلم مع أول صورة له

            $sql = "

                SELECT

                    name,

                    description,

                    type,

                    city,

                    image AS images

                FROM blogs

                WHERE id = :id

            ";

            $q = $con->prepare($sql);

            $q->execute(array(

                "id" => $\_POST["id"]

            ));

            // الحصول على البيانات المدمجة (المعلم والصورة)

            $data = $q->fetch(PDO::FETCH\_ASSOC);

            // دمج المعلم مع الصورة في هيكل واحد

            $response = array(

                'success' => true,

                'data' => $data, // هنا البيانات تتضمن المعلم والصورة

            );

            echo json\_encode($response);

            break;

        case 'approvrd':

            $blog\_id = $\_POST['blog\_id'];

            // تحقق من كون المستخدم أدمن

            if (isset($\_SESSION['auth']) && $\_SESSION['auth']['role'] === 'admin') {

                $stmt = $con->prepare("UPDATE blogs SET is\_approved = TRUE WHERE id = :id");

                $stmt->bindValue(':id', $blog\_id, PDO::PARAM\_INT);

                if ($stmt->execute()) {

                    echo json\_encode(['success' => true]);

                } else {

                    echo json\_encode(['success' => false]);

                }

            } else {

                echo json\_encode(['success' => false, 'message' => 'ليس لديك صلاحية لاعتماد المدونات مطلوب حساب ادمن']);

            }

            break;

        case 'search':

            $isAdmin = isset($\_SESSION['auth']) && $\_SESSION['auth']['role'] === 'admin'; // تحقق إذا كان المستخدم أدمن

            $blog\_type = trim($\_POST['blog\_type']);

            $city = trim($\_POST['city']);

            $search = trim($\_POST['search']);

            $sql = "SELECT

                blogs.id,

                blogs.name,

                blogs.description,

                blogs.image,

                blogs.city,

                COUNT(DISTINCT posts.id) AS comments\_count,

                COALESCE(AVG(blog\_ratings.rating), 0) AS avg\_rating,

                blogs.user\_id,

                blogs.is\_approved,

                blogs.type

            FROM blogs

            LEFT JOIN posts ON blogs.id = posts.blog\_id

            LEFT JOIN blog\_ratings ON blogs.id = blog\_ratings.blog\_id";

            $conditions = [];

            $params = [];

            // إذا كان هناك قيمة للبحث

            if (!empty($search)) {

                $conditions[] = "(blogs.name LIKE :search OR blogs.description LIKE :search)";

                $params['search'] = "%$search%";

            }

            // إذا تم اختيار نوع مدونة

            if (!empty($blog\_type)) {

                $conditions[] = "blogs.type = :blog\_type";

                $params['blog\_type'] = $blog\_type;

            }

            // إذا تم اختيار مدينة

            if (!empty($city)) {

                $conditions[] = "blogs.city = :city";

                $params['city'] = $city;

            }

            // إضافة شرط اعتماد المدونات إذا لم يكن المستخدم أدمن

            if (!$isAdmin) {

                $conditions[] = "(blogs.is\_approved = TRUE OR blogs.user\_id = :current\_user\_id)";

                $params['current\_user\_id'] = $\_SESSION['auth']['id']; // إضافة معرف المستخدم الحالي للمعلمات

            }

            // بناء شروط الاستعلام

            if (!empty($conditions)) {

                $sql .= " WHERE " . implode(' AND ', $conditions);

            }

            // إكمال الاستعلام

            $sql .= " GROUP BY blogs.id, blogs.name, blogs.description, blogs.image, blogs.city";

            // تنفيذ الاستعلام

            $stmt = $con->prepare($sql);

            $stmt->execute($params);

            $results = $stmt->fetchAll(PDO::FETCH\_ASSOC);

            // تجهيز المخرجات

            $output = '';

            foreach ($results as $blog) {

                $stars = generateStars($blog['avg\_rating']); // إنشاء النجوم بناءً على التقييم

                $output .= "

                    <div class='card'>

                        <img src='{$blog['image']}' alt='{$blog['city']}'>

                        <h3>{$blog['name']}</h3>

                        <p>{$blog['description']}</p>

                        <div class='card-footer'>

                            <div class='comments'><i class='fas fa-comments'></i> {$blog['comments\_count']} عدد المنشورات</div>

                ";

                // إذا كانت المدونة غير معتمدة ولكن المستخدم الحالي هو المالك

                if ($\_SESSION['auth']['id'] === $blog['user\_id']) {

                    if(!$blog['is\_approved'])

                    $output .= "<p class='text-danger'> غير معتمدة</p>";

                    else

                    $output .= "<p class='text-info'>معتمدة</p>";

                }

                if ($blog['is\_approved']) {

                    $output .= "<p ><a href='blog\_details.php?id={$blog['id']}' title='عرض تفاصيل المدونة'><i class='fa fa-eye text-primary'></i></a></p>";

                }

                // عرض أزرار التعديل والحذف إذا كان المستخدم هو صاحب المدونة

                if ($\_SESSION['auth']['id'] === $blog['user\_id']) {

                    $output .= "

                        <button class='icone' data-id='{$blog['id']}' onclick='fetchEditData({$blog['id']})'>

                            <i class='fa fa-edit text-success'></i>

                        </button>

                        <button class='icone' data-id='{$blog['id']}' onclick='deleteRecord({$blog['id']})'>

                            <i class='fa fa-trash text-danger'></i>

                        </button>

                    ";

                }

                // عرض زر الاعتماد إذا كان المستخدم أدمن والمدونة غير معتمدة

                if ($isAdmin && !$blog['is\_approved']) {

                    $output .= "

                        <button class='approve-btn' data-id='{$blog['id']}' onclick='approveBlog({$blog['id']})'>

                            <i class='fa fa-check text-primary'></i> اعتماد

                        </button>

                    ";

                }

                $output .= "

                            <div class='rating'>{$stars}</div>

                        </div>

                    </div>

                ";

            }

            echo $output ?: '<p>لا توجد اي مدونة.</p>';

            break;

        case 'delete':

            $sql = "delete from blogs  where id=:id";

            $q = $con -> prepare($sql);

            $q -> execute(array(

                "id"=> $\_POST["id"],

            ));

            if($q -> rowCount()){

                echo json\_encode(array(

                    "success"=> true,

                    "message"=> 'تم الحذف بنجاح',

                ));

            }else{

                echo json\_encode(array(

                    "success"=> false,

                    "message"=> 'فشل في الحذف',

                ));

            }

            break;

        default:

                echo "لا توجد بيانات ";

    }

}

?>

* **Show details blogs:**

**Show details and save blogs posts to database**

<?php

session\_start();

require('header.php');

$\_SESSION['title'] = 'المدونات';

$form\_visible =false;

// الحصول على معرّف المدونة من الرابط (query string)

$blog\_id ='';

$blogId = isset($\_GET['id']) ? intval($\_GET['id']) : 0;

if(isset($\_GET['id'])){

    if ($blogId === 0) {

        echo "<p>المدونة غير موجودة.</p>";

        exit;

    }

    $sql = "

        SELECT

            blogs.id,

            blogs.name,

            blogs.description,

            blogs.image,

            blogs.city,

            COUNT(posts.id) AS comments\_count,

            COALESCE(AVG(blog\_ratings.rating), 0) AS avg\_rating,

            blogs.user\_id

        FROM blogs

        LEFT JOIN posts ON blogs.id = posts.blog\_id

        LEFT JOIN blog\_ratings ON blogs.id = blog\_ratings.blog\_id

        WHERE blogs.id = :blogId

        GROUP BY blogs.id, blogs.name, blogs.description, blogs.image, blogs.city, blogs.user\_id

    ";

    $stmt = $con->prepare($sql);

    $stmt->execute(['blogId' => $blogId]);

    $blog = $stmt->fetch(PDO::FETCH\_ASSOC);

    // إذا لم توجد المدونة

    if (!$blog) {

        echo "<p>لم يتم العثور على المدونة.</p>";

        exit;

    }

    $stars = generateStars($blog['avg\_rating']);

    $blog\_id = $blog['id'];

    $user\_plogs = $blog['user\_id'];

}

$dirPath = 'uploads/posts';

$title = $images = $content ='';

$form\_visible = false;

$allowed\_extension = array('jpg', 'jpeg', 'png', 'gif');

$errors = [];

$action = "create";

if (isset($\_POST['posts'])) {

    $title = trim($\_POST['title']);

    $content = trim($\_POST['content']);

    $blog\_id = trim($\_POST['blog\_id']);

    if (!isset($\_FILES['images']) || count($\_FILES['images']['name']) === 0 || empty($\_FILES['images']['name'][0])) {

        $errors['images'] = 'يجب اختيار صوره على الاقل';

    }

    if (empty($title)) {

        $errors['title'] = 'ادخل العنوان';

    }

    if(isset($\_FILES['images'])){

        foreach ($\_FILES['images']['tmp\_name'] as $key => $tmp\_name) {

            $file\_name = $\_FILES['images']['name'][$key];

            $file\_tmp = $\_FILES['images']['tmp\_name'][$key];

            $file\_extension = pathinfo($file\_name, PATHINFO\_EXTENSION);

            $file\_size = $\_FILES['images']['size'][$key];

            // التحقق من امتداد الملف

            if (!in\_array($file\_extension, $allowed\_extension)) {

                $errors['images'][] = "الملف $file\_name ليس من الأنواع المسموحة.";

            }

            // التحقق من حجم الملف

            if ($file\_size > 3000000) {

                $errors['images'][] = "الملف $file\_name يتجاوز الحجم المسموح به.";

            }

        }

    }

    if (!empty($errors)) {

        $form\_visible = true;

    } else {

        // إنشاء مجلد التخزين إذا لم يكن موجودًا

        if (!is\_dir($dirPath)) {

            mkdir($dirPath, 0777, true);

        }

        // حفظ المنشور في قاعدة البيانات

        $sql = "INSERT INTO posts (title, content, user\_id, blog\_id) VALUES (:title, :content, :user\_id, :blog\_id)";

        $stmt = $con->prepare($sql);

        $stmt->execute([

            'title' => $title,

            'content' => $content,

            'user\_id' => $\_SESSION['auth']['id'],

            'blog\_id' => $blog\_id,

        ]);

        // الحصول على معرّف المنشور الجديد

        $post\_id = $con->lastInsertId();

        // رفع الصور وحفظ المسارات في قاعدة البيانات

        foreach ($\_FILES['images']['tmp\_name'] as $key => $tmp\_name) {

            $file\_name = $\_FILES['images']['name'][$key];

            $file\_extension = pathinfo($file\_name, PATHINFO\_EXTENSION);

            $unique\_name = uniqid('img\_', true) . '.' . $file\_extension;

            $file\_path = $dirPath . "/" . $unique\_name;

            if (move\_uploaded\_file($\_FILES['images']['tmp\_name'][$key], $file\_path)) {

                // إدخال مسار الصورة في قاعدة البيانات

                $sql = "INSERT INTO post\_images (post\_id, image\_path) VALUES (:post\_id, :image\_path)";

                $stmt = $con->prepare($sql);

                $stmt->execute([

                    'post\_id' => $post\_id,

                    'image\_path' => $file\_path,

                ]);

            }

        }

        $form\_visible = false;

    }

}

?>

* **Post Management System**

This code handles various POST requests to create, edit, delete, publish, unpublish, comment on, and like posts based on the provided data.

<?php

require('../database/db.php');

session\_start();

require('../helper.php');

if (isset($\_POST['action'])) {

    switch ($\_POST['action']) {

        case 'approve111':

            $post\_id = $\_POST['post\_id'];

            // // تحقق من كون المستخدم أدمن

            if (isset($\_SESSION['auth']) && $\_SESSION['auth']['role'] === 'admin') {

                $stmt = $con->prepare("UPDATE posts SET status = 'موافق' WHERE id = :post\_id");

                $stmt->bindValue(':post\_id', $post\_id, PDO::PARAM\_INT);

                if ($stmt->execute()) {

                    echo json\_encode(['success' => true]);

                } else {

                    echo json\_encode(['success' => false, 'message' => 'هناك خطأ']);

                }

            } else {

                echo json\_encode(['success' => false, 'message' => 'ليس لديك صلاحية للموافقه على المنشور مطلوب حساب ادمن']);

            }

            break;

        case 'reject':

            $post\_id = $\_POST['post\_id'];

            // تحقق من كون المستخدم أدمن

            if (isset($\_SESSION['auth']) && $\_SESSION['auth']['role'] === 'admin') {

                // استعلام للتحقق من حالة المنشور

                $stmt = $con->prepare("SELECT status FROM posts WHERE id = :post\_id");

                $stmt->bindValue(':post\_id', $post\_id, PDO::PARAM\_INT);

                $stmt->execute();

                $post = $stmt->fetch(PDO::FETCH\_ASSOC);

                // إذا كانت حالة المنشور "قيد المراجعة"، يمكننا الرفض

                if ($post['status'] === 'قيد المراجعة') {

                    // تحديث الحالة إلى "مرفوض"

                    $stmt = $con->prepare("UPDATE posts SET status = 'مرفوض' WHERE id = :post\_id");

                    $stmt->bindValue(':post\_id', $post\_id, PDO::PARAM\_INT);

                    if ($stmt->execute()) {

                        echo json\_encode(['success' => true, 'message' => 'تم رفض المنشور بنجاح']);

                    } else {

                        echo json\_encode(['success' => false, 'message' => 'فشل في رفض المنشور']);

                    }

                } else {

                    // إذا كانت الحالة ليست "قيد المراجعة"، لا نقبل الرفض

                    echo json\_encode(['success' => false, 'message' => 'لا يمكن رفض هذا المنشور، حالته ليست قيد المراجعة']);

                }

            } else {

                echo json\_encode(['success' => false, 'message' => 'ليس لديك صلاحية لرفض المنشور، مطلوب حساب أدمن']);

            }

            break;

        case 'get':

                $blog\_id = $\_POST['blog\_id'];

                $user\_id = $\_SESSION['auth']['id'];

                $is\_admin = $\_SESSION['auth']['role'] == 'admin'; // فرضية أن الدور موجود في الـ session

                $sql = " SELECT

                            posts.id AS post\_id,

                            posts.title,

                            posts.content,

                            posts.created\_at,

                            users.full\_name AS user\_name,

                            users.profile\_image AS user\_image,

                            posts.user\_id,

                            posts.status,  -- حالة المنشور (مقبول، مرفوض، قيد المراجعة)

                            (SELECT COUNT(\*) FROM post\_likes WHERE post\_likes.post\_id = posts.id) AS likes\_count,

                            (SELECT COUNT(\*) FROM post\_comments WHERE post\_comments.post\_id = posts.id) AS comments\_count

                        FROM posts

                        JOIN users ON posts.user\_id = users.id

                        WHERE posts.blog\_id = :blog\_id ";

                $params = [];

                $params['blog\_id'] = $blog\_id;

                // إذا كان المستخدم عادي، عرض المنشورات المقبولة فقط

                if (!$is\_admin) {

                    $sql .= " AND (posts.status = 'موافق' OR posts.user\_id = :current\_user\_id) ";

                    $params['current\_user\_id'] = $user\_id;

                }

                if ($is\_admin) {

                    $sql .= " AND posts.status != 'مرفوض' ";

                }

                $sql .= " ORDER BY posts.created\_at DESC ";

                // تحضير الاستعلام وتنفيذه

                $stmt = $con->prepare($sql);

                $stmt->execute($params);

                $posts = $stmt->fetchAll(PDO::FETCH\_ASSOC);

                if ($posts) {

                    foreach ($posts as $post) {

                        // جلب جميع الصور لهذا المنشور

                        $images\_sql = "SELECT image\_path FROM post\_images WHERE post\_id = :post\_id";

                        $images\_stmt = $con->prepare($images\_sql);

                        $images\_stmt->execute(['post\_id' => $post['post\_id']]);

                        $images = $images\_stmt->fetchAll(PDO::FETCH\_COLUMN);

                        // تحقق إذا كان المستخدم قد أعجب بالمنشور

                        $is\_liked\_sql = "SELECT COUNT(\*) FROM post\_likes WHERE post\_id = :post\_id AND user\_id = :user\_id";

                        $is\_liked\_stmt = $con->prepare($is\_liked\_sql);

                        $is\_liked\_stmt->execute([

                            'post\_id' => $post['post\_id'],

                            'user\_id' => $\_SESSION['auth']['id']

                        ]);

                        $is\_liked = $is\_liked\_stmt->fetchColumn() > 0;

                        $post['is\_liked'] = $is\_liked;

                        // جلب التعليقات

                        $comments\_sql = "SELECT

                                post\_comments.comment AS comment\_content,

                                post\_comments.created\_at AS comment\_time,

                                users.full\_name AS commenter\_name,

                                users.profile\_image AS commenter\_image

                             FROM post\_comments

                             JOIN users ON post\_comments.user\_id = users.id

                             WHERE post\_comments.post\_id = :post\_id

                             ORDER BY post\_comments.created\_at DESC";

                        $comments\_stmt = $con->prepare($comments\_sql);

                        $comments\_stmt->execute(['post\_id' => $post['post\_id']]);

                        $comments = $comments\_stmt->fetchAll(PDO::FETCH\_ASSOC);

                        ?>

                            <div class="fb-post">

                                <div class="top-s">

                                    <div class="top-info">

                                        <div class="profile-picture">

                                            <img src="<?= $post['user\_image'] ?: 'assets/image/default-user.jpeg'; ?>" alt="User Image">

                                        </div>

                                        <div class="top-title">

                                            <div class="profile-name">

                                            <a href="#"><?= htmlspecialchars($post['user\_name']); ?></a>

                                            </div>

                                            <div class="post-time">

                                            <span><?= formatTime($post['created\_at']); ?></span>

                                            </div>

                                        </div>

                                    </div>

                                    <div class="post-content">

                                        <strong><?= htmlspecialchars($post['title']); ?></strong><br />

                                        <p><?= nl2br(htmlspecialchars($post['content'])); ?></p>

                                        <div class="post-images">

                                            <?php if ($images): ?>

                                                <div class="carousel">

                                                    <?php foreach ($images as $image): ?>

                                                        <div class="carousel-item">

                                                            <img src="<?= $image; ?>" alt="Post Image">

                                                        </div>

                                                    <?php endforeach; ?>

                                                </div>

                                            <?php else: ?>

                                                <img src="assets/image/default-post.jpg" alt="Default Post Image">

                                            <?php endif; ?>

                                        </div>

                                    </div>

                                </div>

                                <div class="like-section">

                                    <div class="top-part">

                                        <div class="left-part">

                                            <div class="react">

                                            </div>

                                            <div class="id-name">

                                                <p class="likes-count-id-<?= $post['post\_id']; ?>"><?= $post['likes\_count'] ?>  <span> إعجاب</span></p>

                                            </div>

                                        </div>

                                        <div class="right-part">

                                            <p><span class="comments-count-id-<?= $post['post\_id']; ?>"><?= $post['comments\_count']; ?></span>  تعليقات </p>

                                        </div>

                                    </div>

                                    <div class="bottom-part">

                                        <button class="like-btn" onclick="LikePost(<?= $post['post\_id']; ?>)">

                                            <i class="fas fa-thumbs-up like-button-<?= $post['post\_id']?> <?= $post['is\_liked'] ? 'text-primary' : 'text-secondary'; ?>"></i>

                                            <span>إعجاب</span>

                                        </button>

                                        <div class="comment-btn" >

                                            <i class="fas fa-comment"></i>

                                            <span >   تعليق </span>

                                        </div>

                                        <?php if ($is\_admin): ?>

                                            <div class="approve-reject-btns">

                                                <?php if ($post['status'] == 'قيد المراجعة'): ?>

                                                    <button class="approve-btn" onclick="approvePost(<?= $post['post\_id']; ?>,'approve111')">قبول</button>

                                                    <button class="reject-btn" onclick="rejectPost(<?= $post['post\_id']; ?>,'reject')">رفض</button>

                                                <?php endif; ?>

                                            </div>

                                        <?php elseif ($post['user\_id'] == $user\_id): ?>

                                            <?php

                                                // تحديد الكلاس المناسب حسب حالة المنشور

                                                $status\_class = '';

                                                if ($post['status'] == 'قيد المراجعة') {

                                                    $status\_class = 'text-info'; // حالة قيد المراجعة

                                                } elseif ($post['status'] == 'مقبول') {

                                                    $status\_class = 'text-success'; // حالة مقبول

                                                } elseif ($post['status'] == 'مرفوض') {

                                                    $status\_class = 'text-danger'; // حالة مرفوض

                                                }

                                            ?>

                                            <span class="<?= $status\_class; ?>"><?= $post['status']; ?></span>

                                        <?php endif; ?>

                                        </div>

                                </div>

                                <!-- تعليق-- section -->

                                <div class="all-comments">

                                    <h4>  كل التعليقات </h4>

                                </div>

                                <?php if ($comments): ?>

                                    <?php foreach ($comments as $comment): ?>

                                        <div class="comment-box">

                                            <div class="comment-container">

                                                <div class="comment">

                                                    <img src="<?= $comment['commenter\_image'] ?: 'assets/image/default-user.jpeg'; ?>" alt="" class="comment-img">

                                                    <div class="comment-text">

                                                        <h5 style="margin: 0px;"><?= htmlspecialchars($comment['commenter\_name']); ?></h5>

                                                        <p style="margin: 0px;"><?= nl2br(htmlspecialchars($comment['comment\_content'])); ?></p>

                                                    </div>

                                                </div>

                                                <div class="comment-lks">

                                                    <p><span><?= formatTime($comment['comment\_time']); ?></span></p>

                                                </div>

                                            </div>

                                        </div>

                                    <?php endforeach; ?>

                                <?php else: ?>

                                    <p>لا توجد تعليقات حتى الآن.</p>

                                <?php endif; ?>

                                <div class="comment-s">

                                    <div class="comment-area">

                                        <div class="comment-profile-pic">

                                            <img src="assets/image/default-user.jpeg" alt="user">

                                        </div>

                                        <div class="comment-input-area">

                                            <input type="text" id="comment-text-<?= $post['post\_id']; ?>" placeholder="أكتب تعليقاً..." fpost="0">

                                            <button class="submit-comment-btn" onclick="submitComment(<?= $post['post\_id']; ?>)" data-post-id="<?= $post['post\_id']; ?>"> <i class="fas fa-arrow-circle-left"></i> إرسال</button>

                                        </div>

                                    </div>

                                </div>

                            </div>

                        <?php

                    }

                } else {

                    echo "<p>لا توجد منشورات لعرضها.</p>";

                }

                break;

        case 'like':

            // الحصول على معرّف المنشور

            $post\_id = $\_POST['post\_id'];

            $user\_id = $\_SESSION['auth']['id']; // يجب التأكد من أن المستخدم مسجّل الدخول

            // التحقق من إذا كان المستخدم قد أعجب بالمنشور من قبل

            $sql = "SELECT \* FROM post\_likes WHERE post\_id = :post\_id AND user\_id = :user\_id";

            $stmt = $con->prepare($sql);

            $stmt->execute(['post\_id' => $post\_id, 'user\_id' => $user\_id]);

            $like = $stmt->fetch();

            if ($like) {

                // إذا كان قد أعجب سابقًا، يتم إزالة الإعجاب

                $sql = "DELETE FROM post\_likes WHERE post\_id = :post\_id AND user\_id = :user\_id";

                $stmt = $con->prepare($sql);

                $stmt->execute(['post\_id' => $post\_id, 'user\_id' => $user\_id]);

                $liked = false;

            } else {

                // إذا لم يكن قد أعجب من قبل، يتم إضافة الإعجاب

                $sql = "INSERT INTO post\_likes (post\_id, user\_id) VALUES (:post\_id, :user\_id)";

                $stmt = $con->prepare($sql);

                $stmt->execute(['post\_id' => $post\_id, 'user\_id' => $user\_id]);

                $liked = true;

            }

            // جلب عدد الإعجابات الجديد

            $likes\_count = 0;

            $sql = "SELECT COUNT(\*) AS likes\_count FROM post\_likes WHERE post\_id = :post\_id";

            $stmt = $con->prepare($sql);

            $stmt->execute(['post\_id' => $post\_id]);

            $likes\_count = $stmt->fetch()['likes\_count'];

            // إرجاع الاستجابة

            echo json\_encode(['success' => true, 'likes\_count' => $likes\_count]);

            break;

        case 'add\_comment':

            // الحصول على البيانات

            $post\_id = $\_POST['post\_id'];

            $comment = $\_POST['comment\_text'];

            $user\_id = $\_SESSION['auth']['id']; // تأكد من أن المستخدم مسجل دخوله

            // إضافة التعليق إلى قاعدة البيانات

            $sql = "INSERT INTO post\_comments (post\_id, user\_id, comment) VALUES (:post\_id, :user\_id, :comment)";

            $stmt1 = $con->prepare($sql);

            $stmt1->execute(['post\_id' => $post\_id, 'user\_id' => $user\_id, 'comment' => $comment]);

            // جلب عدد التعليقات الجديد

            $sql = "SELECT COUNT(\*) AS comments\_count FROM post\_comments WHERE post\_id = :post\_id";

            $stmt = $con->prepare($sql);

            $stmt->execute(['post\_id' => $post\_id]);

            $comments\_count = $stmt->fetch()['comments\_count'];

            // إرجاع الاستجابة

            echo json\_encode(['success' => true, 'comments\_count' => $comments\_count]);

            break;

        default :

            // إرجاع الاستجابة الناجحة

            echo json\_encode(['success' => false,'message' => 'هناك خطا']);

            break;

}

}

function formatTime($datetime) {

    $now = new DateTime(); // الوقت الحالي

    $postTime = new DateTime($datetime); // وقت المنشور

    $diff = $now->diff($postTime);

    if ($diff->y > 0) {

        // أكثر من سنة

        return $postTime->format('d M Y'); // يعرض اليوم والشهر والسنة (مثال: 12 ديسمبر 2023)

    } elseif ($diff->m > 0 || $diff->d > 7) {

        // أكثر من أسبوع

        return $postTime->format('d M'); // يعرض اليوم والشهر فقط (مثال: 12 ديسمبر)

    } elseif ($diff->d > 0) {

        // أيام

        return $diff->d == 1 ? 'منذ يوم' : "منذ {$diff->d} أيام";

    } elseif ($diff->h > 0) {

        // ساعات

        return $diff->h == 1 ? 'منذ ساعة' : "منذ {$diff->h} ساعات";

    } elseif ($diff->i > 0) {

        // دقائق

        return $diff->i == 1 ? 'منذ دقيقة' : "منذ {$diff->i} دقائق";

    } else {

        // الآن

        return 'الآن';

    }

}

?>

1. **Frontend Integration:**

The front-end interface was developed using HTML, CSS. JavaScript was added for dynamic functionalities, such as form validation and AJAX requests.

* **Show details blogs page:**

<body>

    <?php require('navbar.php'); ?>

    <div class="containerplogs">

        <div class="blog-details-left">

            <div class="blog-info">

                <h2><?php echo $blog['name']; ?></h2>

                <input name="user\_id" type="hidden" value="<?php echo $\_SESSION["auth"]['id']; ?>">

                <input name="blog\_id" type="hidden" value="<?php echo $blog['id']; ?>">

                <p><strong>المدينة:</strong> <?php echo $blog['city']; ?></p>

                <p><strong>الوصف:</strong> <?php echo $blog['description']; ?></p>

                <p><strong>عدد المنشورات:</strong> <?php echo $blog['comments\_count']; ?></p>

                <div class="row">

                    <div class=" col-sm-4 text-center rating\_averge">

                        <h1 class="text-warning mt-4 mb-4">

                            <b><span id="average\_rating">0.0</span> / 5</b>

                        </h1>

                        <div class="mb-3">

                            <i class="fas fa-star star-light mr-1 main\_star"></i>

                            <i class="fas fa-star star-light mr-1 main\_star"></i>

                            <i class="fas fa-star star-light mr-1 main\_star"></i>

                            <i class="fas fa-star star-light mr-1 main\_star"></i>

                            <i class="fas fa-star star-light mr-1 main\_star"></i>

                        </div>

                        <h3><span id="total\_review">0</span> تقيم</h3>

                    </div>

                    <div class="rating\_progress">

                        <div class="row">

                            <div class="progress-label-left">

                                <b>5</b>

                                <i class="fas fa-star text-warning"></i>

                            </div>

                            <div class="progress">

                                <div class="progress-bar" id="five\_star\_progress" style="width: 0%;"></div>

                            </div>

                            <div class="progress-label-right">(<span id="total\_five\_star\_review">0</span>)</div>

                        </div>

                        <div class="row">

                            <div class="progress-label-left">

                                <b>4</b>

                                <i class="fas fa-star text-warning"></i>

                            </div>

                            <div class="progress">

                                <div class="progress-bar" id="four\_star\_progress" style="width: 0%;"></div>

                            </div>

                            <div class="progress-label-right">(<span id="total\_four\_star\_review">0</span>)</div>

                        </div>

                        <div class="row">

                            <div class="progress-label-left">

                                <b>3</b>

                                <i class="fas fa-star text-warning"></i>

                            </div>

                            <div class="progress">

                                <div class="progress-bar" id="three\_star\_progress" style="width: 0%;"></div>

                            </div>

                            <div class="progress-label-right">(<span id="total\_three\_star\_review">0</span>)</div>

                        </div>

                        <div class="row">

                            <div class="progress-label-left">

                                <b>2</b>

                                <i class="fas fa-star text-warning"></i>

                            </div>

                            <div class="progress">

                                <div class="progress-bar" id="two\_star\_progress" style="width: 0%;"></div>

                            </div>

                            <div class="progress-label-right">(<span id="total\_two\_star\_review">0</span>)</div>

                        </div>

                        <div class="row">

                            <div class="progress-label-left">

                                <b>1</b>

                                <i class="fas fa-star text-warning"></i>

                            </div>

                            <div class="progress">

                                <div class="progress-bar" id="one\_star\_progress" style="width: 0%;"></div>

                            </div>

                            <div class="progress-label-right">(<span id="total\_one\_star\_review">0</span>)</div>

                        </div>

                    </div>

                    </div>

                    <button id="add\_review" class="btn-open-modal">أضف تقييمًا</button>

                    <?php if ($user\_plogs == $\_SESSION['auth']['id']): ?>

                    <button class="btn btn-primary" id="add\_new">إضافة منشور جديداً</button>

                    <?php endif; ?>

                </div>

        </div>

        <div class="blog-details-right">

            <img src="<?php echo $blog['image']; ?>" alt="<?php echo $blog['city']; ?>" class="blog-image">

        </div>

    </div>

    <?php echo $form\_visible?>

    <div id="form-container" style="<?php echo $form\_visible ? 'display: block;' : 'display: none;'; ?>">

        <div class="form-container">

        <form method="post" id="form\_id" enctype="multipart/form-data">

            <input name="blog\_id" type="hidden" value="<?php echo $blog['id']; ?>">

                <div class="form-layout">

                    <div class="form-group-container">

                        <div class="form-group">

                            <input

                                name="title"

                                id="title"

                                value="<?php echo htmlspecialchars($title); ?>"

                                class="input-fieldf "

                                placeholder="عنوان المنشور" />

                            <?php if (isset($errors['title'])) echo "<p class='error-message'>" . $errors['title'] . "</p>"; ?>

                        </div>

                        <textarea

                            name="content"

                            id="content"

                            class="textarea-fieldf"

                            placeholder="بم تفكر؟ اكتب شيـــاً"

                            rows="5"><?php echo htmlspecialchars($content); ?></textarea>

                        <?php if (isset($errors['content'])) echo "<p class='error-message'>" . $errors['content'] . "</p>"; ?>

                    </div>

                    <!-- حقل تحميل الصورة -->

                    <div class="file-input-container">

                        <input type="file" class="input-fiele" name="images[]" id="images" multiple accept="image/\*">

                        <label for="images" class="file-label">

                            <i id="imageIcon" class="fa fa-image"></i>

                            <img id="imagePreview" src="" alt="Image Preview" class="image-preview">

                        </label>

                        <?php if (isset($errors['images'])) echo "<p class='error-message'>" . $errors['images'] . "</p>"; ?>

                    </div>

                </div>

                <div class="form-group full-width-description">

            </div>

                <div class="button-group">

                    <button class="btn btn-save" name='posts' type="submit">حفظ</button>

                    <button class="btn btn-close" type="button" id="close\_form">إغلاق</button>

                </div>

            </form>

        </div>

    </div>

    <div  id="table-container" style="<?php echo !$form\_visible ? 'display: block;' : 'display: none;'; ?>">

        <div class=" post-grid">

        </div>

    </div>

    <!-- مودال التقييم -->

    <div id="review\_modal" class="modal" style="display: none;">

    <div class="modal-dialog">

        <div class="modal-content">

            <div class="modal-header">

                <input name="action" type="hidden" value="addRate">

                <h5 class="modal-title">إضافة تقييم</h5>

                <button type="button" class="close-modal" onclick="modal.style.display = 'none';">&times;</button>

            </div>

            <div class="modal-body">

                <h4 class="text-center mt-2 mb-4">

                    <i class="fas fa-star star-light submit\_star" data-rating="1"></i>

                    <i class="fas fa-star star-light submit\_star" data-rating="2"></i>

                    <i class="fas fa-star star-light submit\_star" data-rating="3"></i>

                    <i class="fas fa-star star-light submit\_star" data-rating="4"></i>

                    <i class="fas fa-star star-light submit\_star" data-rating="5"></i>

                </h4>

            </div>

            <div class="form-group text-center mt-4">

                    <button type="button" class="btn btn-primary" id="save\_review">تقيم</button>

            </div>

        </div>

    </div>

    </div>

* **Code Ajax to Management blogs:**

<script src="assets/js/app.js"></script>

    <script>

    url\_data="helper/blogs\_api.php"

// ربط الحدث بمدخل الملفات

    document.getElementById('image').addEventListener('change', displaySelectedImages);

    document.getElementById('search').addEventListener('change', function(event) {

        event.preventDefault(); // منع إعادة تحميل الصفحة

        get\_Data();

    });

    document.addEventListener('DOMContentLoaded', function() {

        get\_Data(); // استدعاء الدالة عند تحميل الصفحة

    });

    document.getElementById('filter-data').addEventListener('click', function(event) {

        event.preventDefault(); // منع إعادة تحميل الصفحة

        get\_Data();

    });

    function get\_Data(){

        // الحصول على قيمة البحث

        const searchQuery = document.querySelector('input[name="search"]').value;

        const filterCity = document.getElementById('filter-city').value;

        const filterType = document.getElementById('filter-type').value;

        let formData = new FormData();

        formData.append('action', 'search');

        formData.append('search', searchQuery);

        formData.append('city', filterCity);

        formData.append('blog\_type', filterType);

        // استخدام fetch بدلاً من axios لإرسال الطلب

        fetch(url\_data, {

            method: 'POST',

            body: formData

        })

        .then(response => response.text())

        .then(data => {

            // تحديث محتوى الصفحة بالنتائج الجديدة

            document.querySelector('.blogs-grid').innerHTML = data;

        })

        .catch(error => console.error('Error:', error));

    }

    function approveBlog(blogId) {

        let formData = new FormData();

        formData.append('action', 'approvrd');

        formData.append('blog\_id', blogId);

        if (confirm("هل تريد اعتماد هذه المدونة؟")) {

            fetch(url\_data, {

                method: 'POST',

                body: formData

            })

            .then(response => response.json())

            .then(data => {

                if (data.success) {

                    alert('تم اعتماد المدونة بنجاح!');

                    location.reload();

                } else {

                    alert(data.message);

                }

            })

            .catch(error => console.error('Error:', error));

        }

    }

    </script>

* **Blog Post Rating, Review, and Interaction System:**

This code is used to create an interface for rating and reviewing posts (like blogs) with stars, as well as handling various actions such as adding comments, likes, and approving or rejecting posts**.**

<script src="assets/js/app.js"></script>

<script>

    let ratingData = 0;

    var url\_data = "helper/rating\_api.php"

    var url\_post = "helper/posts\_api.php"

// عناصر DOM

const modal = document.getElementById("review\_modal");

const openModalBtn = document.getElementById("add\_review");

const saveReviewBtn = document.getElementById("save\_review");

const stars = document.querySelectorAll(".submit\_star");

document.getElementById('images').addEventListener('change', displaySelectedImages);

// فتح المودال

openModalBtn.addEventListener("click", () => {

        const userId = document.querySelector('input[name=user\_id]').value // افترض وجود عنصر لمعرف المدونة

        const blogId = document.querySelector('input[name=blog\_id]').value // افترض وجود عنصر لمعرف المدونة

        document.querySelector('input[name=action]').value ="addRate"// افترض وجود عنصر لمعرف المدونة

        let formData = new FormData();

        formData.append('action', 'getRating');

        formData.append('user\_id', userId);

        formData.append('blog\_id', blogId);

        // استخدام fetch بدلاً من axios لإرسال الطلب

        fetch(url\_data, {

            method: 'POST',

            body: formData

        })

        .then(response => response.json())

        .then(data => {

            if (data.success) {

            // عرض المودال

            modal.style.display = "flex";

            // إذا كان هناك تقييم، قم بتحديث النجوم

            if (data.rating) {

                resetBackground(); // إعادة تعيين النجوم (إذا كنت تستخدم وظيفة مشابهة)

                // document.querySelector('input[name=action]').value ="udateRate"

                ratingData = data.rating

                for (let i = 1; i <= data.rating; i++) {

                    document.querySelector(`.submit\_star[data-rating="${i}"]`).classList.add("text-warning");

                }

            }

        } else {

            alert(data.message);

        }

        })

        .catch(error => console.error('Error fetching data:', error));

});

// إعادة تعيين النجوم إلى الحالة الافتراضية

function resetBackground() {

    stars.forEach((star) => {

        star.classList.add("star-light");

        star.classList.remove("text-warning");

    });

}

// عند المرور بالفأرة على النجوم

stars.forEach((star, index) => {

    star.addEventListener("mouseenter", () => {

        resetBackground();

        for (let count = 0; count <= index; count++) {

            stars[count].classList.add("text-warning");

            stars[count].classList.remove("star-light");

        }

    });

    // عند الخروج بالفأرة من النجوم

    star.addEventListener("mouseleave", () => {

        resetBackground();

        for (let count = 0; count < ratingData; count++) {

            stars[count].classList.add("text-warning");

            stars[count].classList.remove("star-light");

        }

    });

    // عند النقر على النجمة

    star.addEventListener("click", () => {

        ratingData = index + 1;

        resetBackground();

        for (let count = 0; count < ratingData; count++) {

            stars[count].classList.add("text-warning");

            stars[count].classList.remove("star-light");

        }

    });

});

saveReviewBtn.addEventListener("click", () => {

    if (ratingData > 0) {

        // البيانات المطلوبة

        const userId = document.querySelector('input[name=user\_id]').value // افترض وجود عنصر لمعرف المدونة

        const action = document.querySelector('input[name=action]').value // افترض وجود عنصر لمعرف المدونة

        const blogId = document.querySelector('input[name=blog\_id]').value // افترض وجود عنصر لمعرف المدونة

        let formData = new FormData();

        formData.append('action', action);

        formData.append('user\_id', userId);

        formData.append('blog\_id', blogId);

        formData.append('rating', ratingData);

        // استخدام fetch بدلاً من axios لإرسال الطلب

        fetch(url\_data, {

            method: 'POST',

            body: formData

        })

        .then(response => response.json())

        .then(data => {

            if (data.success) {

                modal.style.display = "none";

            alert(data.message);

            load\_rating\_data();

        } else {

            alert(data.message);

        }

        })

        .catch(error => console.error('Error fetching data:', error));

    }

});

load\_rating\_data();

function load\_rating\_data() {

    const blogId = document.querySelector('input[name=blog\_id]').value; // افترض وجود عنصر لمعرف المدونة

    let formData = new FormData();

    formData.append('action', "getRatingSummary");

    formData.append('blog\_id', blogId);

    // استخدام fetch بدلاً من axios لإرسال الطلب

    fetch(url\_data, {

        method: 'POST',

        body: formData

    })

    .then(response => response.json())

    .then(data => {

        if (data.success) {

            // التحقق من وجود البيانات وتعديل النصوص

            const ratingSummary = data.rating\_summary;

            document.getElementById('average\_rating').textContent = ratingSummary.average\_rating.toFixed(1); // عرض التقييم المتوسط

            document.getElementById('total\_review').textContent = ratingSummary.total\_review; // إجمالي التقييمات

            let count\_star = 0;

            // التعامل مع النجوم باستخدام JavaScript

            document.querySelectorAll('.main\_star').forEach((star, index) => {

                count\_star++;

                if (Math.ceil(ratingSummary.average\_rating) >= count\_star) {

                    star.classList.add('text-warning', 'star-light'); // إضافة تنسيق للنجوم المكتملة

                }

            });

            // تحديث التقييمات والنسب المئوية

            document.getElementById('total\_five\_star\_review').textContent = ratingSummary.five\_star\_review;

            document.getElementById('total\_four\_star\_review').textContent = ratingSummary.four\_star\_review;

            document.getElementById('total\_three\_star\_review').textContent = ratingSummary.three\_star\_review;

            document.getElementById('total\_two\_star\_review').textContent = ratingSummary.two\_star\_review;

            document.getElementById('total\_one\_star\_review').textContent = ratingSummary.one\_star\_review;

            // تحديث الأشرطة باستخدام النسب المئوية

            document.getElementById('five\_star\_progress').style.width = (ratingSummary.five\_star\_review / ratingSummary.total\_review) \* 100 + '%';

            document.getElementById('four\_star\_progress').style.width = (ratingSummary.four\_star\_review / ratingSummary.total\_review) \* 100 + '%';

            document.getElementById('three\_star\_progress').style.width = (ratingSummary.three\_star\_review / ratingSummary.total\_review) \* 100 + '%';

            document.getElementById('two\_star\_progress').style.width = (ratingSummary.two\_star\_review / ratingSummary.total\_review) \* 100 + '%';

            document.getElementById('one\_star\_progress').style.width = (ratingSummary.one\_star\_review / ratingSummary.total\_review) \* 100 + '%';

        } else {

            alert(data.message); // في حال حدوث خطأ

        }

    })

    .catch(error => console.error('Error fetching data:', error)); // التعامل مع الأخطاء

}

document.addEventListener('DOMContentLoaded', function() {

        get\_Data\_post(); // استدعاء الدالة عند تحميل الصفحة

    });

function get\_Data\_post(){

        // الحصول على قيمة البحث

        const blogId = document.querySelector('input[name=blog\_id]').value; // افترض وجود عنصر لمعرف المدونة

        let formData = new FormData();

        formData.append('action', 'get');

        formData.append('blog\_id', blogId);

        // استخدام fetch بدلاً من axios لإرسال الطلب

        fetch(url\_post, {

            method: 'POST',

            body: formData

        })

        .then(response => response.text())

        .then(data => {

            // تحديث محتوى الصفحة بالنتائج الجديدة

            document.querySelector('.post-grid').innerHTML = data;

        })

        .catch(error => console.error('Error:', error));

}

function rejectPost(postId,status){

        // الحصول على قيمة البحث

        let formData = new FormData();

        formData.append('action', status);

        formData.append('post\_id', postId);

        if (confirm("هل تريد رفض المنشور  ؟")) {

        // استخدام fetch بدلاً من axios لإرسال الطلب

        fetch(url\_post, {

            method: 'POST',

            body: formData

        })

        .then(response => response.text())

        .then(data => {

            if (data.success) {

                    alert('تم رفض المنشور بنجاح!');

                    location.reload(); // إعادة تحميل الصفحة لتحديث الحالة

            } else {

               // alert(data.message);

            }

        })

        .catch(error => console.error('Error:', error));

    }

}

function approvePost(postId,status){

        // الحصول على قيمة البحث

        let formData = new FormData();

        formData.append('action', status);

        formData.append('post\_id', postId);

        if (confirm("هل تريد الموافقة على المنشور  ؟")) {

        // استخدام fetch بدلاً من axios لإرسال الطلب

        fetch(url\_post, {

            method: 'POST',

            body: formData

        })

        .then(response => response.text())

        .then(data => {

            if (data.success) {

                    // alert('تم اعتماد المدونة بنجاح!');

                    location.reload(); // إعادة تحميل الصفحة لتحديث الحالة

            } else {

            }

        })

        .catch(error => console.error('Error:', error));

    }

}

function LikePost(postId) {

// إرسال طلب AJAX إلى الخادم لحفظ الإعجاب

    let formData = new FormData();

    formData.append('action', 'like');

    formData.append('post\_id', postId);

    fetch(url\_post, {

        method: 'POST',

        body: formData

    })

    .then(response => response.json())

    .then(data => {

        if (data.success) {

            // تحديث عدد الإعجابات في الواجهة

            let likeButton = document.querySelector(`.like-button-${postId}`);

            let likesCount = document.querySelector(`.likes-count-id-${postId}`);

            likeButton.classList.toggle('text-primary');

            likeButton.classList.toggle('text-secondary');

            likesCount.textContent = data.likes\_count; // تحديث العدد

        } else {

            console.error('Error liking post');

        }

    })

    .catch(error => console.error('Error:', error));

}

// دالة لإظهار حقل التعليق عند النقر على زر التعليق

function showCommentBox(postId) {

    // إخفاء جميع حقول التعليقات الأخرى

    let allCommentBoxes = document.querySelectorAll('.comment-box');

    allCommentBoxes.forEach(box => {

        box.style.display = 'none';

    });

    // إظهار حقل التعليق للمنشور الذي تم النقر عليه

    let commentBox = document.getElementById('comment-box-' + postId);

    commentBox.style.display = 'block';

}

// دالة لإرسال التعليق

function submitComment(postId) {

    let commentText = document.getElementById('comment-text-' + postId).value;

    if (commentText.trim() === "") {

        alert("يرجى كتابة تعليق!");

        return;

    }

    let formData = new FormData();

    formData.append('action', 'add\_comment');

    formData.append('post\_id', postId);

    formData.append('comment\_text', commentText);

    // إرسال التعليق عبر AJAX

    fetch(url\_post, {

        method: 'POST',

        body: formData

    })

    .then(response => response.json())

    .then(data => {

        if (data.success) {

            let commentsCount = document.querySelector(`.comments-count-id-${postId}`);

            commentsCount.textContent = data.comments\_count; // تحديث عدد التعليقات

            document.getElementById('comment-text-' + postId).value = "";

        } else {

            alert("حدث خطأ، يرجى المحاولة مرة أخرى.");

        }

    })

    .catch(error => console.error('Error:', error));

}

</script>

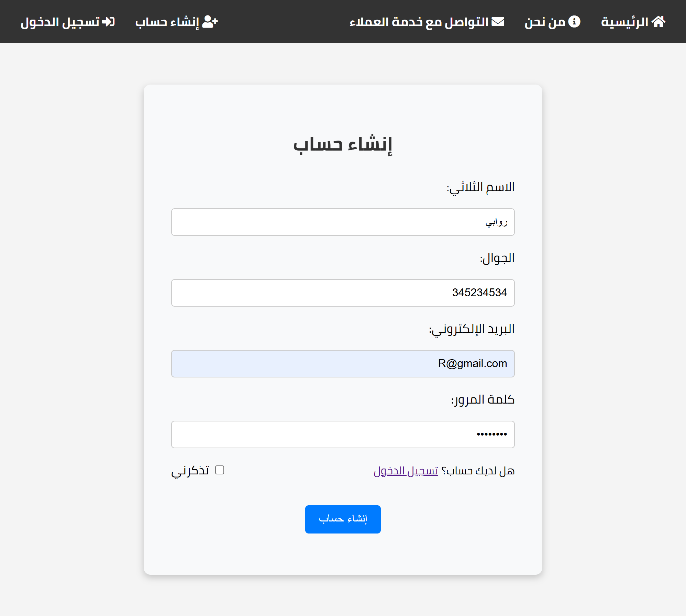
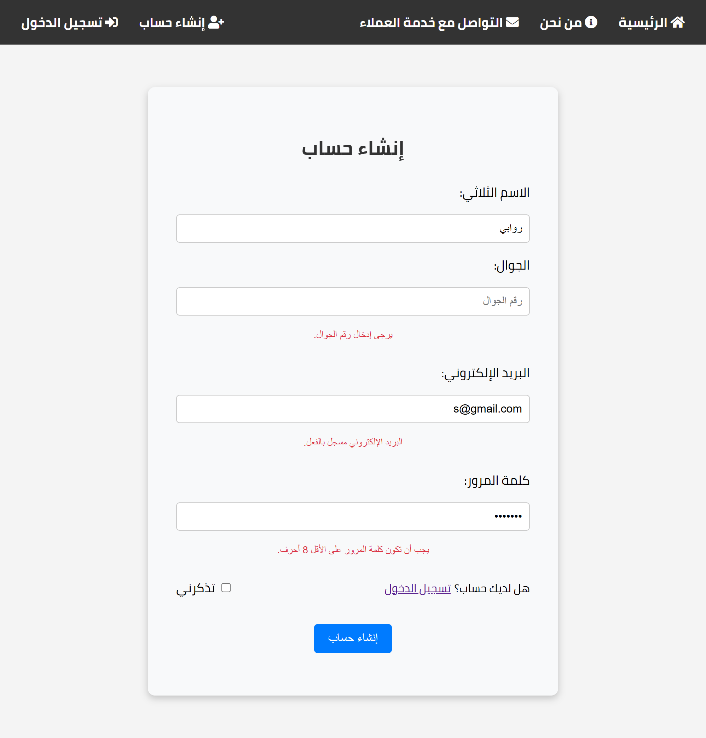
1. **Testing Case**

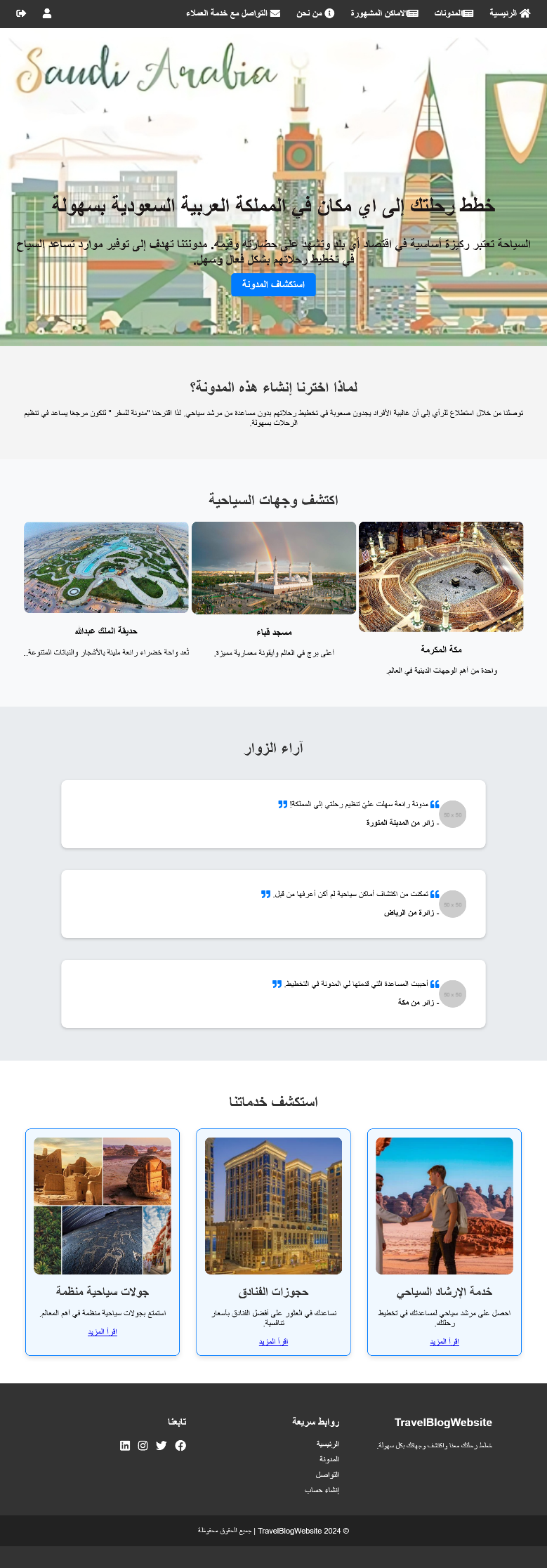
Testing was carried out to ensure the system's reliability and correctness. Below are snapshots of a typical testing case:

* **Sing up Functionality**:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Case Description | Expected Result | Actual Result |
| 1 | Entering an invalid email format. | System displays an error message indicating an invalid email format. | As Expected |
| 2 | Entering an empty name field. | System displays an error message indicating the "Name" field is required. | As Expected |
| 3 | Entering an empty phone number field. | System displays an error message indicating the "Phone" field is required. | As Expected |
| 4 | Entering an email already registered in the system. | System displays an error message: "البريد الإلكتروني مسجل بالفعل" | As Expected |
| 5 | Entering a weak password (e.g., less than 8 characters). | System displays an error message indicating that the password is too weak. | As Expected |
| 6 | Leaving all fields empty. | System displays an error message indicating that all required fields are missing. | As Expected |
| 7 | Entering all valid details (name, phone, email, password). | System successfully registers the user and redirects them to the homepage. | As Expected |

.

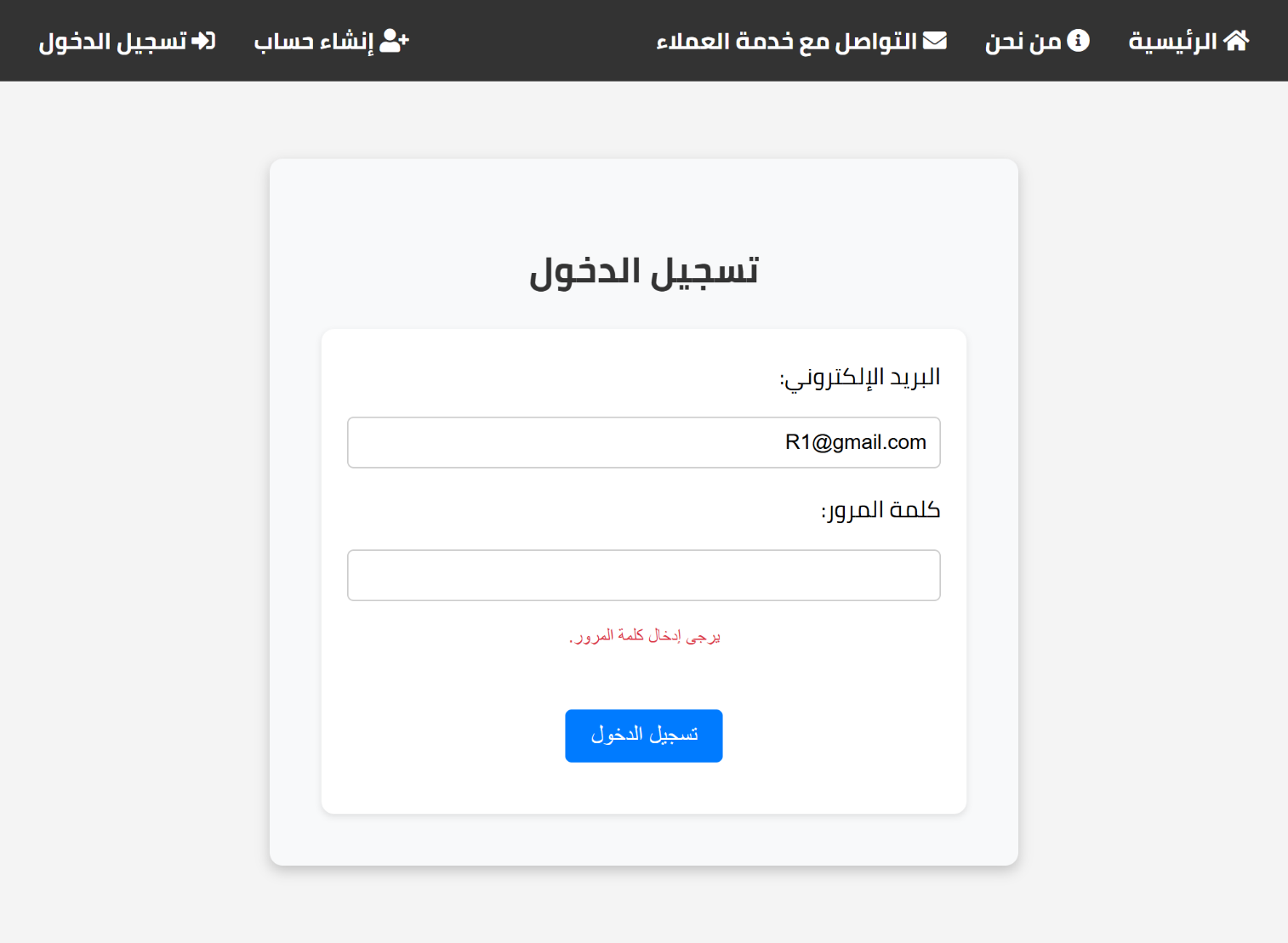




* **Login Functionality:**
  + Function : Login(Admin ,User)
  + Description: Entering correct email and password.
  + Preconditions: User account with valid email and password exists.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Case Description | Expected Result | Actual Result |
| 1 | Entering an invalid email format. | System displays an error message indicating an invalid email format. | As Expected |
| 2 | Entering a valid email with an incorrect password. | System does not allow user to log in and displays an error message: "كلمة المرور خطأ". | As Expected |
| 3 | Attempting login with empty email and password fields. | System displays an error message indicating that both email and password are required. | As Expected |
| 4 | Entering a valid email and password. | System allows the user to log in successfully and redirects to the homepage. | As Expected |
| 5 | Entering a valid email but no registered account. | System displays an error message: "المستخدم غير موجود." | As Expected |





1. **Results**

The final results demonstrate that the web application functions as intended. Key results include:

* **User Management**:
  + Users can register, log in, and manage their sessions without issues.
* **Comment System**:
  + Users can post comments on approved posts, and unauthorized attempts are properly handled.
* **Admin Features**:
  + Administrators can manage posts efficiently, with the system restricting unauthorized access to admin functionalities.
* **Responsiveness**:
  + The application performs well across different devices, with no significant usability issues reported during testing.

# Chapter 6: Conclusion and Future Work

## Conclusion

This project successfully developed a comprehensive system designed to manage user accounts, facilitate the creation and management of posts, and incorporate interactive features like adding popular locations and historical landmarks. The system includes various functionalities, such as user registration and login, posting content, and ensuring data validation throughout the user experience.

The system’s design and implementation allow users to interact with different modules effectively. The registration and login functionalities ensure secure user authentication, while the content management system (CMS) enables users to create, edit, and delete posts. A key feature of the platform is the ability to add locations and historical landmarks, allowing users to enrich their posts with valuable geographical and historical data. This feature improves the overall engagement of the platform by providing users with the option to share and discover notable places.

Additionally, the platform’s user interface is built to be intuitive, ensuring ease of use when managing both posts and user profiles. It also includes necessary validations and error handling to ensure a smooth user experience, even when incorrect or incomplete data is entered.

The system was tested thoroughly through a series of test cases, including validating the registration process, ensuring the proper functioning of the login mechanism, and checking the error messages displayed for incorrect inputs. All core functionalities were successfully implemented and tested to ensure reliability and user satisfaction.

## Limitations

Despite the success of the project, there are several limitations that should be considered:

1. **Scalability**: Although the system is designed to handle a moderate user load, further optimization will be required to handle a significantly larger number of concurrent users. As the project grows, additional measures for scaling the database and improving performance may be necessary.
2. **Platform Compatibility**: Currently, the system is designed for a specific platform (e.g., web or mobile). Expanding support to other platforms or ensuring cross-platform functionality will require additional development effort.
3. **User Experience (UX) Design**: While the basic functionality is complete, the user interface (UI) and user experience (UX) could be improved further for a more intuitive and polished look. Features such as advanced styling, animations, and mobile responsiveness need to be enhanced to provide a more engaging user experience.
4. **Security Measures**: While basic security mechanisms (e.g., password hashing, form validation) have been implemented, additional features such as multi-factor authentication (MFA) and data encryption could be added to improve the security of sensitive user data.
5. **Testing**: Although the project has undergone basic testing, there are areas that still require more extensive testing, including stress testing, usability testing, and compatibility checks on different devices and browsers. More comprehensive test cases, especially edge cases, should be explored to ensure the system operates smoothly under a variety of conditions.

## Future Work

There are several opportunities for future development and enhancements in the system:

1. **Mobile App Development**: Developing a mobile app version of the system would increase accessibility, allowing users to post content, view updates, and interact with others while on the move. The mobile version could also incorporate offline capabilities, enabling users to work without an active internet connection.
2. **Enhanced Content Features**: Expanding the content management system to allow users to add images, videos, and links to their posts will enrich the content experience. Users should be able to upload media related to historical landmarks and popular locations, which would increase user engagement and enrich the platform’s value.
3. **Integration of Social Sharing Features**: Allowing users to share posts on external social media platforms (such as Facebook, Twitter, and Instagram) directly from the system could help attract new users and improve platform visibility.
4. **Advanced Location Integration**: The system could benefit from integrating with mapping and geolocation APIs (such as Google Maps) to automatically fetch details about historical landmarks or popular locations, offering users an even more seamless experience when adding places to their posts.
5. **AI and Machine Learning**: Implementing AI-powered features, such as content recommendation engines based on user preferences, could increase engagement and provide users with more personalized content. Machine learning algorithms could be used to suggest popular locations or historical landmarks that align with user interests.
6. **Performance Optimization**: To accommodate an increasing number of users and content, performance optimization techniques such as caching, database indexing, and load balancing should be implemented. This will ensure the system remains fast and responsive even as user activity increases.
7. **User Analytics**: Integrating user analytics into the platform could help administrators track engagement, popular locations and posts, and detect trends. Insights gained from this data could be used to improve the platform and introduce new features based on user needs.

# References

1. [https://www.thedollztertravels.com/.](https://www.thedollztertravels.com/)
2. <https://safar22.com/>
3. <https://wanderlusters.com/>
4. <https://travelhero.app/blog/>
5. “W3Schools.com.” <https://www.w3schools.com/html/default.asp>
6. “W3Schools.com.” <https://www.w3schools.com/css/default.asp>
7. “W3Schools.com.” <https://www.w3schools.com/js/default.asp>