



Palestine Polytechnic University

College of Information Technology and Computer Engineering

Hebron City Guide Website

Team members:

Yafa Jawabreh Shahd Abu Rmeileh Nareman Qabaja

Supervisor name:

Dr. Ezdehar Jawabreh

2023-2024

الإهداء

قال تعالى: (وَأَخِرُ دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ)

الحمد لله الذي ما تمَّ جُهد ولا خُتم سعي إلا بفضلِهِ، الحمد لله على البُلوغ ثم الحمد لله على التَّمام.

بكل حُب نُهدي ثمرة نجاحنا وتخرُّجنا هذه إلى الذين يعلو حُبهم فوق كل حب، إلى من أناروا طريق العلم وساندوا ووفروا كُل سُبُل السعادة والنجاح، إلى عائلتنا التي دعمت بلا حدود وأعطت بلا مُقابل، إلى من غرست في الروح مكارم الأخلاق، إلى القلوب المعطاءة، الأهل الأعزاء لكم منا كل الحب والامتنان

إلى جميع من أمدونا بالقوة والتوجيه والدعم في الأوقات الصعبة لنصل إلى ما نحن عليه اليوم

إلى الذين كانوا عوناً لنا في مشروعنا هذا ونوراً يُضيء الظلمة التي كانت تقف أحياناً في طريقنا، لكم جزيل الشكر والتقدير

وأيضاً بكل وفاء وتقدير واعترافاً منا بالجميل، نتقدم بجزيل الشكر ل دكتورتنا:

ازدهار جوابرة، على كل ما قدمته لنا من توجيهات ونصائح لإتمام هذا المشروع.

Abstract

The Hebron City Guide Project is dedicated to creating a comprehensive Website that showcases the diverse range of establishments in Hebron, such as universities, pharmacies, hospitals, and more. With detailed information provided for each category, this website serves as an indispensable resource for both locals and visitors, enabling them to navigate the city with ease. Our project recognizes the importance of accessibility and convenience, aiming to bridge the gap between individuals and the various services available in Hebron.

In addition to providing comprehensive details about each location, including addresses and contact information and more. It also incorporates advertisement features, allowing local businesses to promote their services and offerings to a wider audience. Furthermore, the website includes an events section, where users can stay updated on the latest happenings in Hebron, including festivals, exhibitions, and community gatherings. By integrating these features, our project not only facilitates exploration but also enhances the overall experience of discovering the city.

By highlighting the wide range of establishments and services, the Hebron City Guide Website Project plays a crucial role in promoting Hebron's economic growth. The platform creates opportunities for local businesses to increase their visibility and reach potential customers.

الملخص

مشروع موقع دليل مدينة الخليل مخصص لإنشاء منصة شاملة تعرض مجموعة متنوعة من المؤسسات في الخليل، مثل الجامعات والصيدليات والمستشفيات وغيرها. مع المعلومات التفصيلية المقدمة لكل فئة، يعد هذا الموقع بمثابة مورد لا غنى عنه لكل من السكان المحليين والزوار، مما يمكنهم من التنقل في المدينة بسهولة. يدرك مشروعنا أهمية سهولة الوصول والراحة، ويهدف إلى سد الفجوة بين الأفراد والخدمات المختلفة المتاحة في الخليل.

بالإضافة إلى توفير تفاصيل شاملة عن كل موقع، بما في ذلك العناوين ومعلومات الاتصال والمزيد. كما يتضمن أيضًا ميزات إعلانية، مما يسمح للشركات المحلية بالترويج لخدماتها وعروضها لجمهور أوسع. علاوة على ذلك، يتضمن الموقع قسمًا للفعاليات، حيث يمكن للمستخدمين البقاء على اطلاع دائم بأخبار الأحداث في الخليل، بما في ذلك المهرجانات و المعارض والتجمعات المجتمعية. ومن خلال دمج هذه الميزات، لا يسهل مشروعنا الاستكشاف فحسب، بل يعزز أيضًا التجربة الشاملة لاكتشاف المدينة.

من خلال تسليط الضوء على مجموعة واسعة من المؤسسات والخدمات، يلعب مشروع موقع دليل مدينة الخليل دورًا حاسمًا في تعزيز النمو الاقتصادي في الخليل. تخلق المنصة فرصًا للشركات المحلية لزيادة ظهورها والوصول إلى العملاء المحتملين.

Table of content

Abstract.....	2
المخلص.....	3
Table of content.....	4
List of Tables.....	6
List of Figures.....	8
Chapter 1 Introduction.....	10
1.1 Overview.....	11
1.2 Project's Description.....	11
1.3 Motivation.....	11
1.4 Scope.....	11
1.5 Objectives.....	11
1.6 Previous project.....	12
1.7 Alternatives for building the project.....	13
1.8 Context diagram.....	14
Chapter 2 Functional and nonfunctional requirements.....	15
2.1 Overview.....	16
2.2 Actors.....	16
2.3 Functional requirement.....	16
2.3.1 User's side :.....	16
2.3.2 Admin's side:.....	16
2.3.3 System's Side :.....	17
2.3.3.1 Chatbot.....	17
2.3.3.2 Rasa framework.....	17
2.3.3.3 System side functional Requirements:.....	19
2.4 Nonfunctional requirements.....	19
2.5 Use-case diagram.....	20
2.6 User's functional requirements tables.....	21
2.7 Admin's functional requirements tables.....	28
2.8 System's functional requirements tables.....	33
Chapter 3 Architecture Design.....	36
3.1 Overview.....	37
3.2 Architectural alternatives.....	37
3.3 Architecture.....	39
3.3.1 MVC Architecture's components.....	39
3.3.2 MVC Architecture' s components Description.....	39

3.4 Class Diagram.....	42
3.5 Database logical mapping.....	43
3.6 Database normalization.....	44
3.7 Database description.....	44
3.8 Database Tables.....	46
Chapter 4 Implementation.....	50
4.1 Overview.....	51
4.2 Software environment and tools:.....	51
4.3 User interface.....	54
4.3.1 Home page.....	54
4.3.2 Registration page.....	55
4.3.3 Login page.....	55
4.3.4 Event section.....	56
4.3.5 Advertisement section.....	56
4.3.6 Category page.....	57
4.3.7 Places page.....	58
4.3.8 View place page.....	59
4.3.9 Add advertisement page.....	60
4.3.10 Add place page.....	61
4.3.11 Profile page.....	62
4.3.12 Notification section.....	62
4.3.13 Add category page.....	63
4.3.14 Add event page.....	64
4.3.15 Chatbot Section.....	65
4.4 Rasa framework files.....	66
Chapter 5 Testing.....	69
5.1 Overview.....	70
5.2 Validation.....	70
5.3 System check.....	70
Chapter 6 Conclusion and Future work.....	83
6.1 Conclusion.....	84
6.2 Future work.....	84
6.3 References.....	84

List of Tables

Table number	Table name	Page number
2.6.1	Registration	21
2.6.2	Login	22
2.6.3	View profile	22
2.6.4	View all places	23
2.6.5	View all detailed information	23
2.6.6	View advertisements	24
2.6.7	View events	24
2.6.8	View comments posted by other users	25
2.6.9	View users rating	25
2.6.10	Search	26
2.6.11	Add comment	26
2.6.12	Rating	26
2.6.13	Add new place	27
2.6.14	Add new advertisement	28
2.6.15	Interact with chatbot	28
2.7.1	View notification	28
2.7.2	Check suggested place	29
2.7.3	Check suggested advertisement	29
2.7.4	Add place	30
2.7.5	Delete/Update place	30
2.7.6	Add new advertisement	31

2.7.7	Delete/Update advertisement	32
2.7.8	Add new event	32
2.7.9	Delete/Update event	32
2.7.10	Add category	32
2.8.1	Information retrieval	33
2.8.2	Understand User Intent and Provide Relevant Responses	34
2.8.3	Provide information about upcoming events	35
3.8.1	auth_user table description	46
3.8.2	Place table description	47
3.8.3	Category table description	47
3.8.4	Ratings table description	47
3.8.5	comments table description	48
3.8.6	Adv table description	48
3.8.7	Events table description	49
3.8.8	notification table description	49
3.8.9	branch table description	49
3.8.10	phone_number table description	49
5.3.1	Registration process	71
5.3.2	Login process	75
5.3.3	Add new advertisement process	77
5.3.4	Add new place process	78
5.3.5	Chatbot test	80

List of Figures

Figure number	Figure name	Page number
1.8	Context diagram	14
2.3.3.2	Rasa architecture	18
2.5	Use case diagram	20
3.3	MVC architecture	39
3.4	Class diagram	42
3.5	Database logical mapping	43
4.2.1	Bootstrap logo	51
4.2.2	VS code logo	51
4.2.3	Web2py logo	52
4.2.4	Postman logo	52
4.2.5	Rasa logo	53
4.2.6	MySQL logo	53
4.3.1	Home page	54
4.3.2	Registration page	55
4.3.3	Login page	55
4.3.4	Event page	56
4.3.5	Advertisement section	56
4.3.6	Category page	57
4.3.7	Places page	58
4.3.8	View place page	59

4.3.9	Add advertisement page	60
4.3.10	Add place page	61
4.3.11	Profile page	62
4.3.12	Notification section	62
4.3.13	Add category page	63
4.3.14	Add event page	64
4.3.15	Chatbot section	65
4.4	Nlu.yml file content	66
4.4	Stories.yml file content	67
4.4	Domain.yml file content	68
4.4	Actions of rasa	68

Chapter 1 Introduction

1.1 Overview

In this chapter we will discuss a description of the project, why we chose this idea and motivated us to make it our project, what is the scope of this project and what are the objectives that will be achieved and more.

1.2 Project's Description

This project is to create a user-friendly website that serves as a guide to the city of Hebron. This website will aggregate information about the city's landmarks, events, and much more. By consolidating this data into a single platform, visitors and residents of Hebron will have easy access to a wealth of information, enabling them to explore the city. The website will feature interactive maps, detailed descriptions, high-quality images to ensure a seamless experience for users. It also incorporates advertisement features, allowing local businesses to promote their services. Furthermore, the website includes an events section, where users can stay updated on the latest happenings in Hebron, including festivals, exhibitions, and community gatherings.

1.3 Motivation

The motivation behind the Hebron City Guide Website Project lies in addressing the lack of a centralized online platform that provides accurate and up-to-date information about events, landmarks, and local businesses in Hebron. Currently, there is a significant challenge for both residents and visitors in accessing reliable and comprehensive information about the city. By creating a user-friendly website that aggregates and organizes this information, we aim to empower individuals to explore and engage with Hebron more effectively. Also to use the services provided by the website, such as advertisements, events, and interaction with Chatbot. Thus enhancing the overall experience of discovering the city of Hebron.

1.4 Scope

Creating a web application for (the city of Hebron) targeting visitors, tourists, and residents of the city.

1.5 Objectives

- **Providing comprehensive information:** The project aims to gather accurate and up-to-date information about Hebron's landmarks, events, and local businesses.
- **Promoting local businesses:** The project will incorporate advertisement features, allowing local businesses to showcase their services and products.

- **Keeping users informed about events:** The website will include an events section that will keep users updated on the latest happenings in Hebron. This will cover festivals, exhibitions, cultural events, and community gatherings,
- **Engaging user community:** The website will incorporate interactive features such as user comment and ratings where users can share their experience.
- **Intelligent assistance through a chatbot:** The project will integrate a chatbot feature into the website, providing users with intelligent assistance and personalized recommendations. The chatbot will be designed to understand user queries, provide relevant information about attractions, events, and local businesses in Hebron.

1.6 Previous project

There is a project called "دليل إلكتروني تجاري وحرفي خاص بمدينة الخليل" that shares a similar concept with our Hebron City Guide project. However, it has been implemented in a different manner compared to our approach.

The previous project aimed to display commercial, economic, healthcare, and other locations, providing users with information about specific places. It also had the capability to add and showcase advertisements.[\[1\]](#) However, this project utilized outdated tools and techniques that do not align with the technological advancements of today.

Therefore, it has become necessary to keep with technological advancements and create a comprehensive platform. Our project goes beyond the previous one by incorporating modern tools and technologies. It not only displays commercial locations but also features an advanced advertising system, allowing local businesses to promote their services and offerings to a wider audience. Additionally, our website includes an events section where users can stay updated on the latest happenings in Hebron, such as festivals, exhibitions, and community gatherings.

We have also incorporated a chatbot that is capable of answering users questions and providing relevant information. The chatbot serves as a virtual assistant, offering guidance and support to users as they navigate through the platform and explore the various offerings in Hebron.

1.7 Alternatives for building the project

When it comes to building a project, there are several alternatives to consider. Two popular options are developing a website and creating a dedicated mobile app. Each approach has its own set of features and disadvantages.

1. Website:

Features:

- Easy to use by users.
- Access the site from any type of device (computers and smart devices).
- Easy to update at any time.
- It can be opened from a browser running on any operating system.

Disadvantages:

- The constant need to connect to the Internet.

2. Mobile application:

Features:

- Ease of use.
- Always available to the user through the mobile phone.
- Easy to update.

Disadvantages:

- The constant need to connect to the Internet.
- People avoid downloading apps due to space limitations on their phones.
- Diversity of phone operating systems.
- People will use it only when they need it, not all the time.

After looking at the suggested alternatives, we chose the website to build the system due to its accessibility and versatility, but we can add the concept of responsive web design to ensure a better experience when accessed from mobile devices. Responsive web design allows the website layout and content to adapt and adjust based on the screen size and orientation, providing an optimal user experience on mobile devices. By employing responsive design techniques, we can make the website appear and function similarly to a mobile app when accessed on a mobile device.

1.8 Context diagram

Figure 1.8 shows the most important elements surrounding the system or individuals and their interaction mechanism With the system, they are mainly: user and admin.

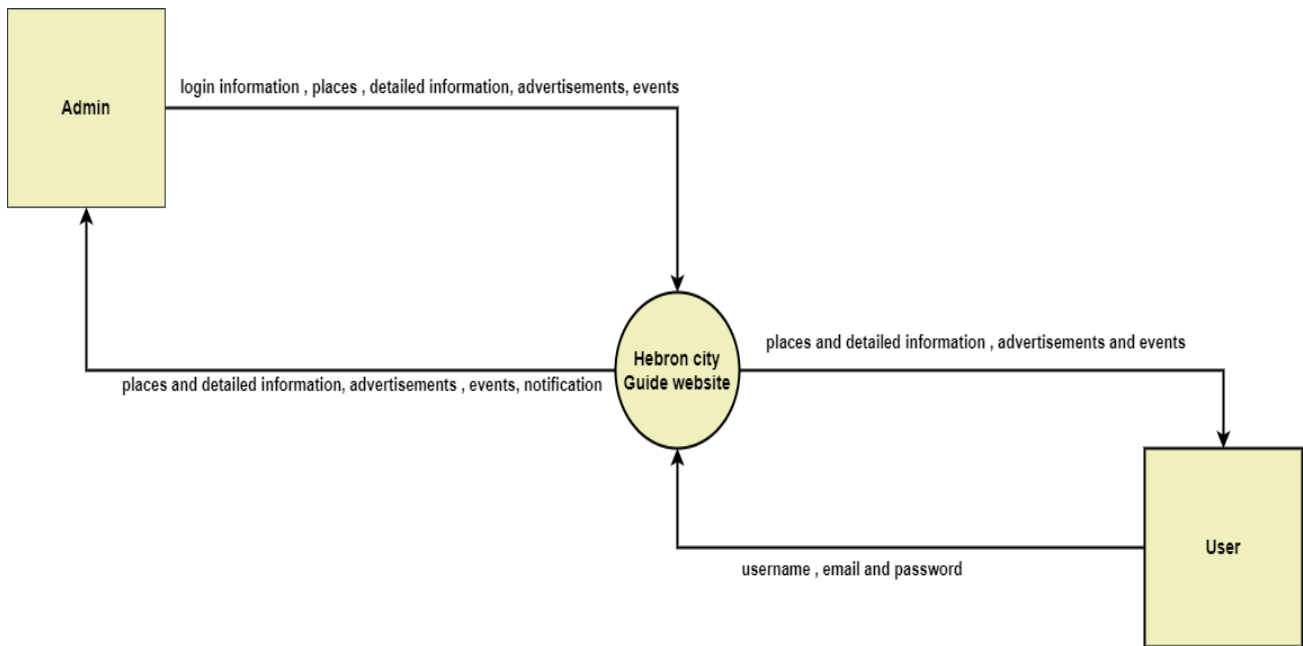


Figure 1.8 Context diagram

Chapter 2 Functional and nonfunctional requirements

2.1 Overview

In this chapter, we will discuss the functional requirements for admin's side, user's side and system's side, and the non-functional requirements of the project.

2.2 Actors

- User
- Admin

2.3 Functional requirement

2.3.1 User's side :

- Users are able to register to the website by providing personal information.
- Users can login to the website using email and password.
- Users are able to view all the places in Hebron city.
- Users are able to view all detailed information of the selected place.
- Users are able to add comments to specific places.
- Users are able to view comments posted by other users.
- Users are able to rate specific places.
- Users are able to view user ratings.
- Users are able to interact with Chabot to ask it any questions about the places ,detailed information and more.
- Users are able to view advertisements.
- Users are able to view events happening in Hebron city.
- Users are able to view their profile.
- Users are able to add advertisements by providing all correct information.
- Users are able to add new places by providing all correct information.
- Users are able to search for specific places.

2.3.2 Admin's side:

- The admin can login to the website using email and password.
- The admin is able to view all places in Hebron city.
- The admin is able to view all detailed information of the selected place.
- The admin is able to view notifications.
- The admin is able to check suggested places by users.
- The admin is able to view user comments.
- The admin is able to view user ratings.
- The admin is able to check suggested advertisements by users.
- The admin is able to view advertisements.
- The admin is able to add new advertisements to the website
- The admin is able to delete/update advertisement information.

- The admin is able to view events.
- The admin is able to add new events.
- The admin is able to delete/update events.
- The admin is able to add a new category.
- The admin is able to add new places to the website
- The admin is able to delete/update places.

2.3.3 System's Side :

2.3.3.1 Chatbot

1. Definition

A chatbot is a computer program that can simulate a conversation or chat with a user in natural language through messaging applications, website, or mobile applications and interact with users according to their input and should be available 24/7.

2. How does chatbot work?

Chatbot analyzes the inputted text and matches the text with predefined data called intents which are categorized to manage the conversation. The user utterance is tagged with one of these intents, even if what the user says stretches over two or more intents. Most chatbots will take the intent with the highest score and take the conversation down that avenue.

2.3.3.2 Rasa framework

Rasa is an open-source framework for building conversational AI chatbots and assistants. It provides tools and libraries for developing and deploying AI-driven, text-based chatbots that can handle natural language conversations with users. It also allows the user to train the model and add custom actions.[\[4\]](#)

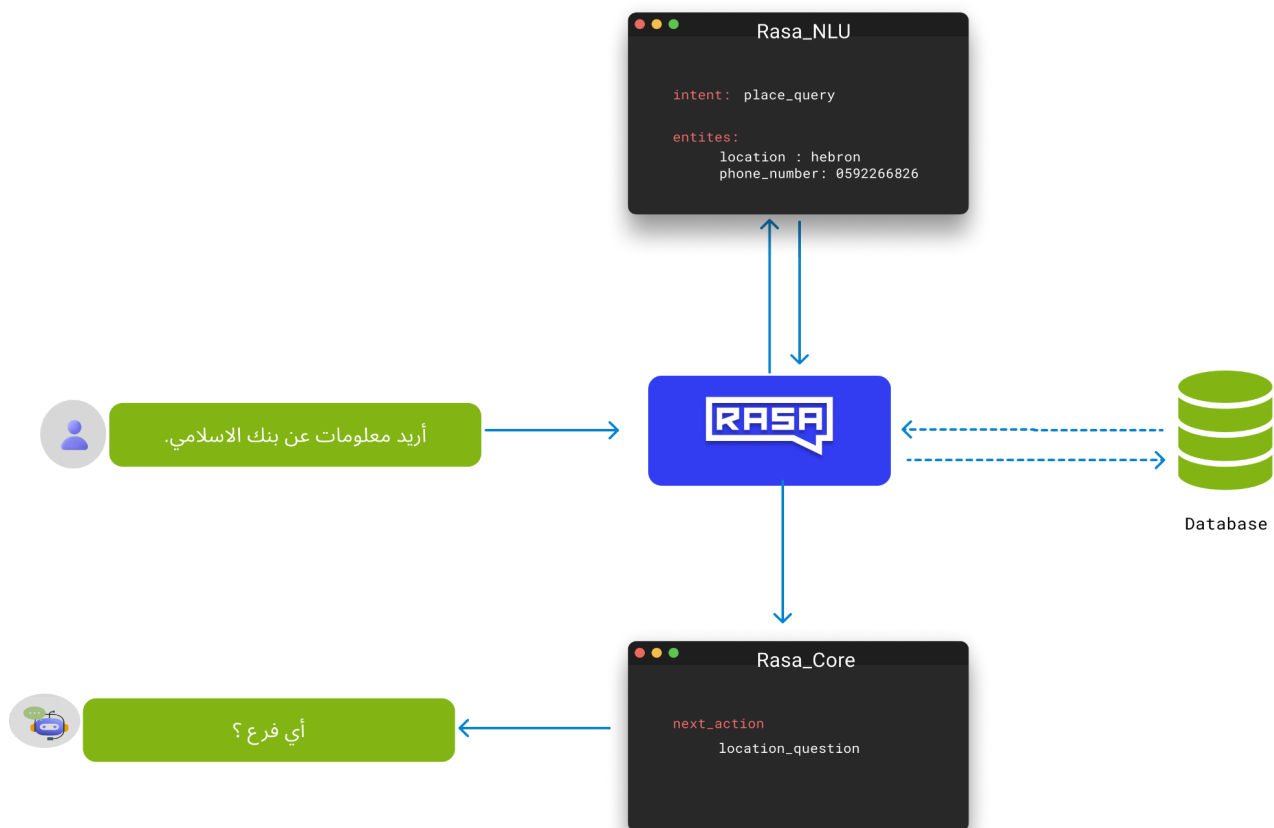


Figure 2.3.3.2 Rasa architecture

Rasa has two main components:

- **Rasa NLU** : Is an open-source natural language processing tool for intent classification (decides what the user is asking), extraction of the entity from the bot in the form of structured data and helps the chatbot understand what user is saying.
- **Rasa Core** : a chatbot framework with machine learning-based dialogue management which takes the structured input from the NLU and predicts the next best action using a probabilistic model.

In our project we will use the Rasa framework to build an AI chatbot that helps users to answer their questions about website, places , information of specific places and much more.

2.3.3.3 System side functional Requirements:

- The chatbot should be able to understand the user's intent and provide relevant responses.
- The chatbot should be able to provide detailed information about each place ,including location, social media accounts, and a brief description.
- The chatbot should be able to provide information about upcoming events or events happening on a specific date, including date, time, location, and a brief description.

2.4 Nonfunctional requirements

- The system is available for all users so it can be used at all times when the mobile phone/pc and internet connection are available.
- The system should be easy to use, provide convenient interfaces for the user, and enable users to access the system easily and quickly.
- The system can achieve a high level of reliability, ensuring that users have a consistent and dependable experience when accessing information about Hebron City.

2.5 Use-case diagram

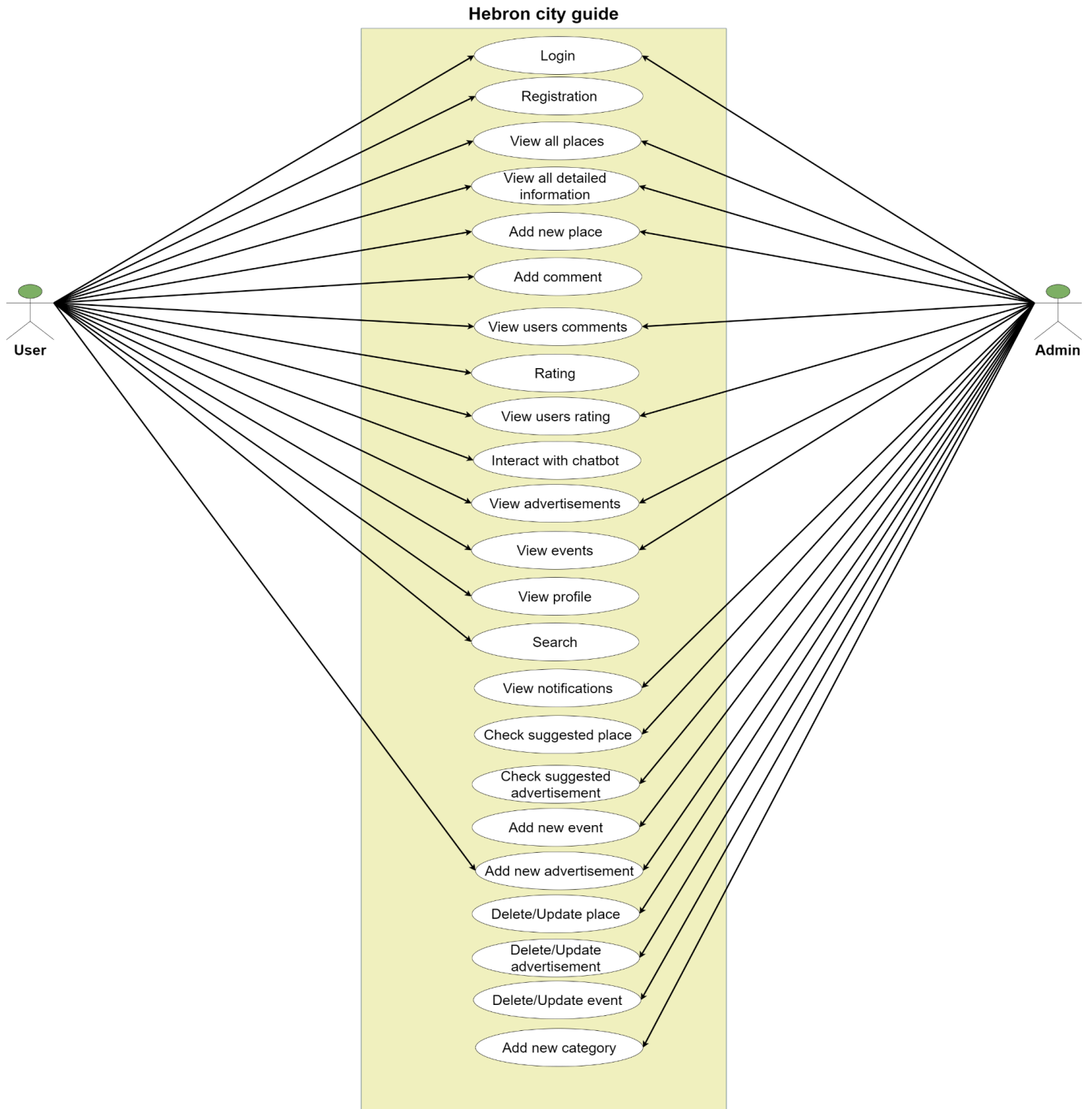


Figure 2.5 Use case diagram

2.6 User's functional requirements tables

Requirement	Registration
Actor	User
Objectives	Create a new account.
Precondition	The website is being open.
Scenario	<ol style="list-style-type: none"> 1. Open a website from any browser. 2. Choose the sign-up option. 3. Fill the required information: <ol style="list-style-type: none"> a. First name b. Last name c. Email d. Password e. Confirm password 4. Press the sign-up button.
Exception	<ol style="list-style-type: none"> 1. The user enters an existing email. 2. The user enters an invalid email. 3. The user enters a password with less than 8 characters. 4. Password and confirm password are not the same. 5. The user does not fill all the required fields. 6. No Internet connection.

Table 2.6.1 Registration

Requirement	Login
Actors	User , admin
Objectives	Login to the system.
Precondition	The user must register in the system.
Scenario	<ol style="list-style-type: none"> 1. Open the website from any browser. 2. Enter email. 3. Enter password. 4. Press on the login button.
Exception	<ol style="list-style-type: none"> 1. User is not registered yet. 2. Invalid email. 3. The password is incorrect. 4. No internet connection.

Table 2.6.2 Login

Requirement	View profile
Actors	User
Objectives	View the user profile.
Precondition	The user must register in the system and log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the profile option. 2. The view profile page will appear.
Exception	<ol style="list-style-type: none"> 1. User is not registered yet. 2. No internet connection.

Table 2.6.3 View profile

Requirement	View all places.
Actors	User, admin
Objectives	View the places in Hebron city.
Precondition	The website is open.
Scenario	<ol style="list-style-type: none"> 1. Open the website. 2. Choose the places option. 3. Choose the category in which the desired place is located. 4. All places belonging to the selected category will appear.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The place does not exist in the system

Table 2.6.4 View all places

Requirement	View all detailed information.
Actors	User, admin
Objectives	View detailed information of specific places.
Precondition	The website is open.
Scenario	<ol style="list-style-type: none"> 1. Open the website. 2. Choose the places option. 3. Choose the category in which the desired place is located. 4. All places belonging to the selected category will appear. 5. Choose the desired place. 6. The information about this place will appear.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The place does not exist in the system

Table 2.6.5 View all detailed information

Requirement	View advertisements.
Actors	User, admin
Objectives	View the advertisements that were uploaded to the website.
Precondition	The website is being open.
Scenario	<ol style="list-style-type: none"> 1. Choose the advertisements option. 2. The advertisement section will appear.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. There are no advertisements yet.

Table 2.6.6 View advertisements

Requirement	View events.
Actors	User, admin
Objectives	View the events that are happening in Hebron.
Precondition	The website is being open.
Scenario	<ol style="list-style-type: none"> 1. Choose the events option. 2. The event section will appear.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. There are no events yet.

Table 2.6.7 View events

Requirement	View comments posted by other users
Actors	User, admin
Objectives	View users' comments to take feedback.
Precondition	The website is open.
Scenario	<ol style="list-style-type: none"> 1. View selected place. 2. All users' comments will appear.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. There are no comments.

Table 2.6.8 View comments posted by other users

Requirement	View users rating
Actors	User, admin
Objectives	View users rating to take feedback .
Precondition	The website is open.
Scenario	<ol style="list-style-type: none"> 1. View selected place. 2. All users ratings will appear.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. There are no ratings.

Table 2.6.9 View users rating

Requirement	Search
Actors	User
Objectives	Users can search for specific places.
Precondition	The website is open.
Scenario	<ol style="list-style-type: none"> 1. Open the website. 2. Choose the search option. 3. Enter the name of the place. 4. Information about the place will appear.

Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The place doesn't exist.
-----------	---

Table 2.6.10 Search

Requirement	Add comment
Actors	User
Objectives	Users add comments to specific places.
Precondition	The user must register in the system and log in.
Scenario	<ol style="list-style-type: none"> 1. View selected place. 2. Add a new comment. 3. press the submit button. 4. The comment will be added.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The user is not logged in yet.

Table 2.6.11 Add comment

Requirement	Rating
Actors	User
Objectives	Users can rate the selected place.
Precondition	The user must register in the system and log in.
Scenario	<ol style="list-style-type: none"> 1. View selected place. 2. The user can rate the place between 1-5 stars. 3. press the submit button. 4. The rating will be added.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. User is not logged in yet.

Table 2.6.12 Rating

Requirement	Add new place
Actors	User
Objectives	Users add a new place that does not exist on the website.
Precondition	The user must register in the system and log in.
Scenario	<ol style="list-style-type: none"> 1. Choose Add place option. 2. The page will appear. 3. Fill in the correct information. 4. Press the submit button. 5. Then the form will be sent to the admin to verify the information and then add it to the website.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The user does not fill out all required fields 3. the place already exists.

Table 2.6.13 Add new place

Requirement	Add new advertisement.
Actors	User
Objectives	Users add new advertisements by providing all correct information.
Precondition	The user must register to the system and log in.
Scenario	<ol style="list-style-type: none"> 1. Choose Add new Advertisement option. 2. Add advertisement page will appear. 3. Fill in the correct information. 4. Press the submit button.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The user did not fill out all required fields. 3. The advertisement already exists.

Table 2.6.14 Add new advertisement

Requirement	Interact with Chabot
Actors	User
Objectives	Users ask questions then Chabot will answer them.
Precondition	The website is open.
Scenario	<ol style="list-style-type: none"> 1. Open the website. 2. Then it will appear on the home page. 3. The user asks the chatbot a question 4. Look at tables 2.8.1 , 2.8.2 and 2.8.3
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. Chabot malfunction.

Table 2.6.15 Interact with Chabot

2.7 Admin's functional requirements tables

Requirement	View notification
Actors	Admin
Objectives	View all notifications about the system.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose The notification option. 2. All notifications will appear.
Exception	<ol style="list-style-type: none"> 1. No internet connection.

Table 2.7.1 View notification

Requirement	Check suggested place
Actors	Admin
Objectives	Check the places that were suggested by the users to be added if its information is correct.

Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the notification option. 2. Choose the add new place form. 3. Validate the data. 4. Press the add button. 5. The place will be added to the website.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The place already exists.

Table 2.7.2 Check suggested place

Requirement	Check suggested advertisement
Actors	Admin
Objectives	Check the advertisements that were suggested by the users to be added if its information is correct.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the notification option. 2. Choose the add new advertisement form. 3. Validate the data. 4. Press the add button. 5. The advertisement will be added to the website.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The advertisement already exists.

Table 2.7.3 Check suggested advertisement

Requirement	Add new place
Actors	Admin
Objectives	Admin adds a new place that does not exist in the website.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the add new place option. 2. The add new place page will appear . 3. Fill all required information. 4. Press the add button. 5. The place will be added.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. the place already exists.

Table 2.7.4 Add place

Requirement	Delete/Update place
Actors	Admin
Objectives	Delete or update a specific place.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose a place. 2. press the delete/update button. 3. The place will be deleted/updated.
Exception	<ol style="list-style-type: none"> 1. No internet connection.

Table 2.7.5 Delete/Update place

Requirement	Add new advertisement.
Actors	Admin
Objectives	Admin add new advertisements by providing all correct information.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose Add new Advertisement option. 2. Add advertisement page will appear. 3. Fill all required information. 4. Press the add button. 5. The advertisement will be added.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The user did not fill out all required fields. 3. The advertisement already exists.

Table 2.7.6 Add new advertisement

Requirement	Delete/Update advertisement
Actors	Admin
Objectives	Delete/Update specific advertisement.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the advertisement. 2. Press the delete/update button. 3. The advertisement will be deleted/updated.
Exception	<ol style="list-style-type: none"> 1. No internet connection.

Table 2.7.7 Delete/Update advertisement

Requirement	Add new event
Actors	Admin
Objectives	Add new events to the website.

Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the add event option. 2. Fill all required information. 3. Press the add button. 4. The event will be added.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The event already exists.

Table 2.7.8 Add new event

Requirement	Delete/Update event
Actors	Admin
Objectives	Delete/Update specific event.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the event. 2. Press the delete/update button. 3. The event will be deleted/updated.
Exception	<ol style="list-style-type: none"> 2. No internet connection.

Table 2.7.9 Delete/Update event

Requirement	Add category
Actors	Admin
Objectives	Add a new category to the website.
Precondition	The admin must log in.
Scenario	<ol style="list-style-type: none"> 1. Choose the add category option. 2. Fill all required information. 3. Press the add button. 4. The category will be added.
Exception	<ol style="list-style-type: none"> 1. No internet connection. 2. The category already exists.

Table 2.7.10 Add category

2.8 System's functional requirements tables

Requirement	The chatbot should be able to provide detailed information about a specific place, including its location, social media accounts, and a brief description.
Actors	User
Objectives	Obtain comprehensive details about a specific place, such as the location, social media presence, and a description
Precondition	The user has initiated a conversation with the chatbot and requested information about a specific place.
Scenario	<ol style="list-style-type: none"> 1. The user asks "Tell me about polytechnic?" 2. The chatbot analyzes the user's input and recognizes that they are seeking information about a specific place. 3. The chatbot will respond "These are the information about polytechnic ..." 4. The chatbot will respond "Did that information help you? " 5. The user responds "That's great information, thank you" 6. The chatbot will respond "I'm glad I was able to help you"
Exception	Chatbot does not have the specific information requested by the user in its database.

Table 2.8.1 Information retrieval

Requirement	Understand User Intent and Provide Relevant Responses
Actors	User
Objectives	Analyze user input and match it to relevant information in the database and provide responses to users.
Precondition	The user has initiated a conversation with the chatbot and presented a query or request.
Scenario	<ol style="list-style-type: none"> 1. The user sends a message to the chatbot, expressing their information need or request (e.g., "I want restaurants located in [specific location]"). 2. The chatbot analyzes the user's message to determine their intent and recognizes that they are seeking information about a place in a specific location. 3. The chatbot provides a relevant response that addresses the user's intent, such as providing places near a specific location with additional details about each option.
Exception	The chatbot is unable to accurately determine the user's intent or does not have sufficient information to provide a relevant response.

Table 2.8.2 Understand User Intent and Provide Relevant Responses

Requirement	The chatbot should be able to provide information about upcoming events or events happening on a specific date, including date, time, location, and a brief description.
Actors	User
Objectives	Obtain comprehensive details about upcoming events or events happening on a specific date, such as the date, time, location, and a description of the event
Precondition	The user has initiated a conversation with the chatbot and requested information about upcoming events or events on a specific date.
Scenario	<ol style="list-style-type: none"> 1. The user asks for information about upcoming events or events on a specific date (e.g., "What events are happening in month 6 ?") 2. The chatbot provides a list of relevant upcoming events or events scheduled for the specified date. 3. The user asks a question to get more details about one of the listed events (e.g., "Can you tell me more about expo event?") 4. The chatbot provides additional, relevant information to address the user's question
Exception	The chatbot is unable to provide the requested information about specific upcoming events or events on a particular date, as it does not have that data available in the database.

Table 2.8.3 Provide information about upcoming events

Chapter 3 Architecture Design

3.1 Overview

In this chapter, we will discuss the system's architecture, database tables, and database mapping.

3.2 Architectural alternatives

1. Model-View-Controller (MVC) architecture:

Separates application into three interconnected parts:

- **Model:** Manages data and business logic.
- **View:** Handles user interface and presentation.
- **Controller:** Coordinates interactions between model and view. [\[2\]](#)

Pros:

- Improves code reusability and testability.
- Enhances separation of concerns.
- Simplifies UI development and maintenance.

Cons:

- Can become complex for large applications.
- Can involve additional code and code complexity when the data model and interactions are simple.

2. 3-tier architecture:

Separates application into three interconnected parts:

- **Presentation Layer:** Represents the user interface of the application and is responsible for presenting data to the user and receiving user input.
- **Application Layer:** Contains the business logic of the application and performs the core processing of the data.
- **Data Storage Layer:** Responsible for managing the data storage and retrieval in the system. [\[3\]](#)

Pros:

- Promotes modularity and loose coupling.
- Enhances maintainability and testability.
- Simplifies development and deployment.

Cons:

- The communication between layers in a distributed architecture can introduce additional latency and overhead, which may have an impact on the overall performance of the system.
- May require additional layers for complex scenarios.

After studying both the MVC and Three-Tier Architecture, we have chosen MVC as the preferred architectural pattern for our project. Several factors contributed to this decision. Firstly, MVC's clear separation of concerns aligns well with our project's requirements. Secondly, the widespread adoption of MVC across different frameworks and technologies is another significant advantage. By choosing MVC, we can leverage the existing ecosystem of tools, libraries, and frameworks that support this architectural pattern. Finally, The ability to reuse components across different parts of the application will also enhance productivity and reduce development time, benefiting our project in the long run.

3.3 Architecture

In our website application we choose an MVC architecture which is an architectural design pattern that organizes an application's logic into distinct components, each of which carries out a specific set of tasks. The components interact with each other to ensure that the application's functionality is delivered in a coordinated and consistent manner.

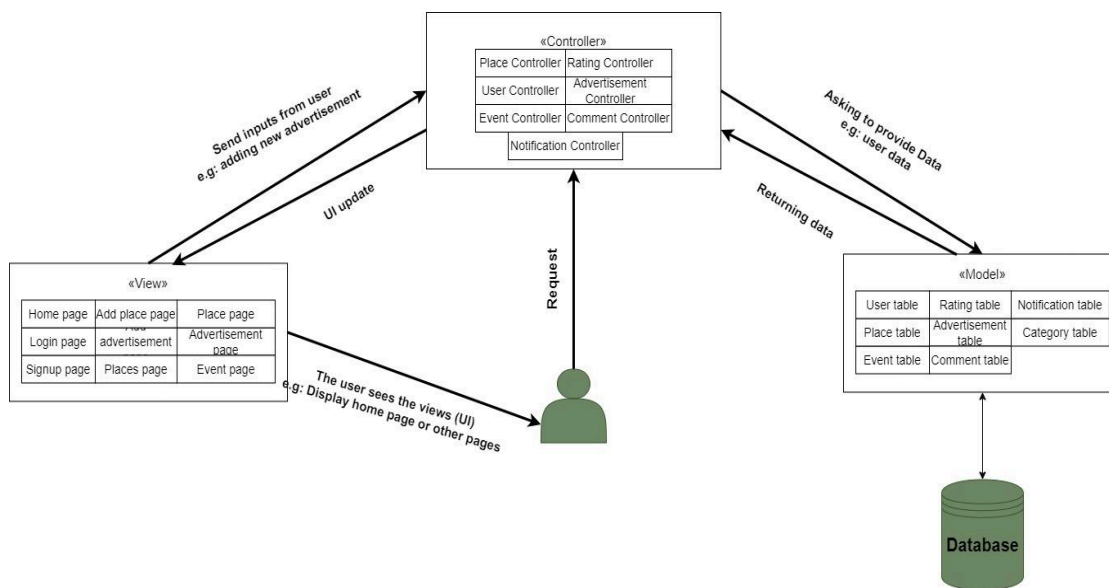


Figure 3.3 : MVC architecture

3.3.1 MVC Architecture's components

1. **Model**
2. **View**
3. **Controller**

3.3.2 MVC Architecture's components Description

1. **Model**

The Model represents the application's data and business logic. It encapsulates data manipulation, validation, and interaction with the application's data storage. It acts as a gateway between the data and the rest of the application, making it independent of how the data is presented. [2] In section 3.3 figure 3.3 Class diagram represent:

- Classes representing entities and their relationships.
- Data attributes.

2. View

The View is responsible for presenting data to the user in a user-friendly format. It represents the user interface (UI) and does not contain any business logic.[\[2\]](#) The View gets updated whenever the underlying data in the Model changes. It is often implemented as interfaces, specifying:

- Components for visual elements.
- Methods for displaying data.
- Mechanisms for user interaction.

Section 4.3 explains some of the user interface in our website.

3. Controller

The Controller acts as an intermediary between the Model and the View. It receives user input from the View, processes it, and interacts with the Model accordingly. It updates the Model based on the user's actions and instructs the View to update its display.[\[2\]](#)

We have several controllers such as :

- **Default controller:**
 - **Responsible for managing user-related operations.**
 - Some of its methods:**
 - user : Register/login user to the system.
 - profile: Enable users to display their profile.
 - index : View home page of the system.
- **Place controller:**
 - Manages operations related to places in Hebron.**
 - Some of its methods:**
 - addnewplace: Adds a new place to the system.
 - update: Modifies place details.
 - viewplace: Retrieves and displays place information.
 - delete_place: Delete specific place from the system.
- **Comment controller:**
 - Manages comment-related operations.**
 - Some of its methods:**
 - add_comment: Allows users to add a comment to a specific place.
 - update: Enables users to modify their previous comment for a place.
 - delete_comment: Enables users to delete their previous comment for a place.

- **Event controller:**
 - Manages event-related operations.**
 - Some of its methods:**
 - addnewevent: Adds a new event to the system.
 - update: Modifies event details.
 - delete_event: Admin deletes a specific event from the system.
- **Advertisement controller:**
 - Manages advertisement-related operations.**
 - Some of its methods:**
 - addnewAd: Adds a new advertisement to the system.
 - update: Modifies advertisement details.
 - delete_adv: Delete specific advertisement from the system.
- **Notification controller:**
 - Manages notification-related operations.**
 - Some of its methods:**
 - viewnotification: Retrieves and displays notifications.
 - add_user_adv: Add user suggested advertisement to the system.
 - add_user_place: Add user suggested place to the system.

3.4 Class Diagram

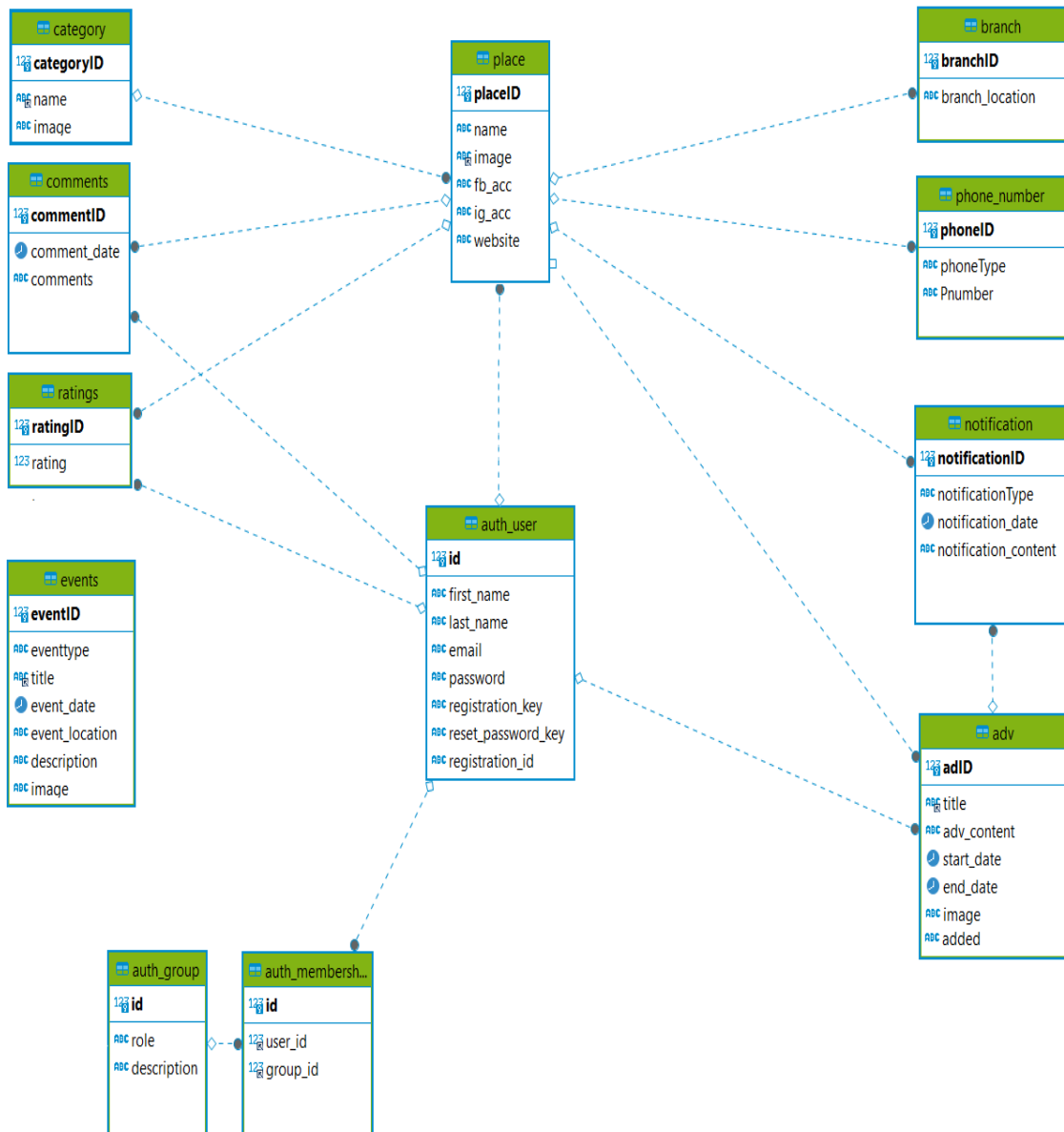


Figure 3.4 Class diagram

3.5 Database logical mapping

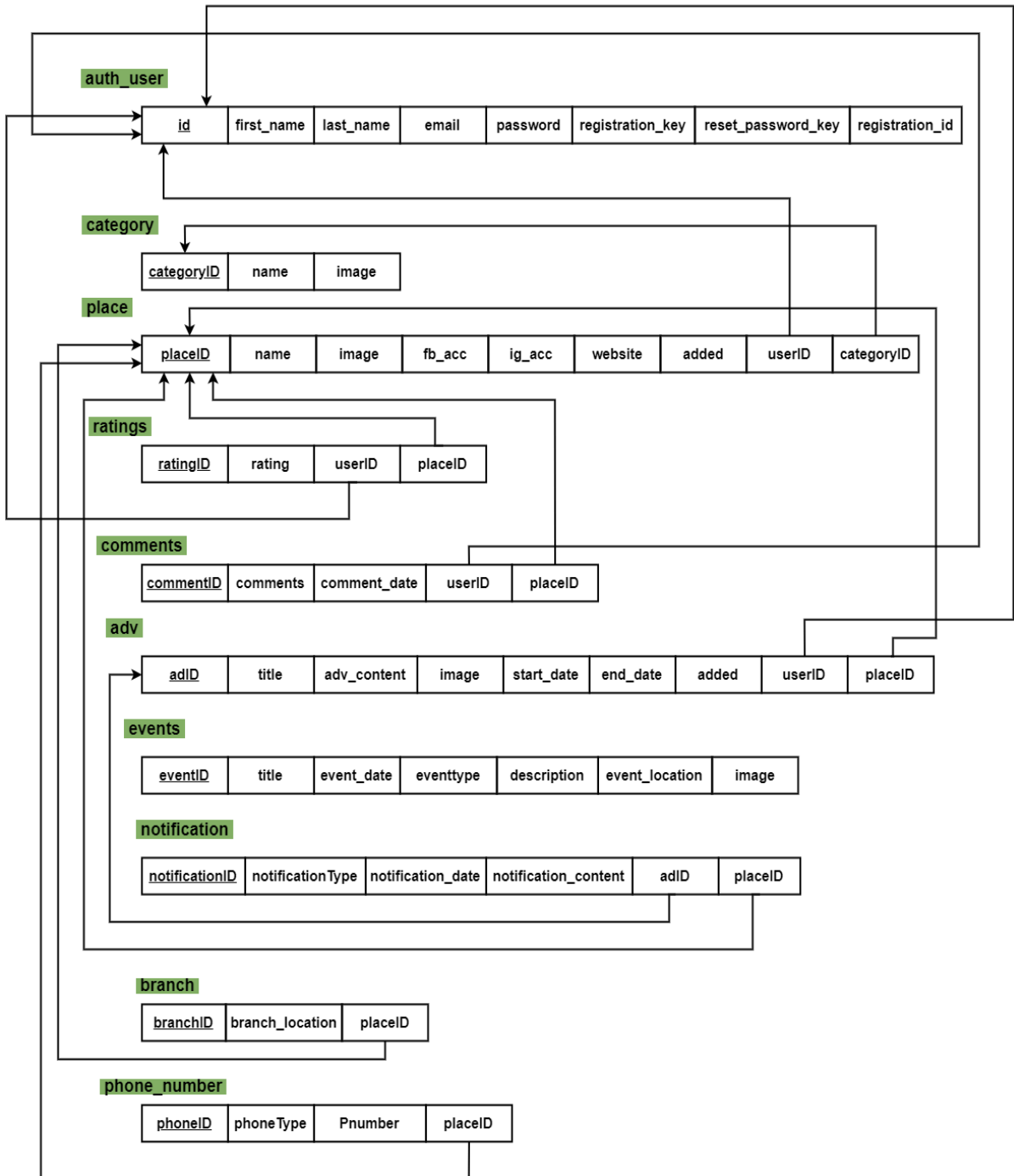


Figure 3.5 Database logical mapping

3.6 Database normalization

At this stage, the database that appeared in the mapping stage is modified by applying different types of normalization, including 1NF, 2NF, 3NF.

The database resulting from the mapping stage is ready in the 3NF format, which is common for databases, as each table has been converted to a relation, compound keys have been eliminated Also, multi-dependency has been eliminated.

3.7 Database description

1. auth_user
 - id: unique integer with auto increment for all users (PK).
 - first_name: string, the first name of the user.
 - last_name: string, the last name of the user.
 - email: string , unique and not-null.
 - password : string and not-null with 8 digits at minimum.
 - registration_key: string, It stores a randomly generated key that is sent to the user's email for verification.
 - reset_password_key: string, is used when a user requests a password reset.
 - registration_id: string, it stores registration-related information.
2. place
 - placeID: unique integer with auto increment for all places (PK).
 - name: string and not-null.
 - image: string, a picture of the place.
 - fb_acc: string, facebook account of specific place.
 - ig_acc: string, instagram account of specific place.
 - website: string, website page of specific place.
 - added: string, suggested user place.
 - categoryID: unique integer, represents the ID of the category (FK).
 - userID: unique integer, represents the ID of the user (FK).
3. category
 - categoryID: unique integer with auto increment for all categories (PK).
 - name: string and not-null.
 - image: string, a picture of the category.
4. ratings
 - ratingID: unique integer with auto increment (PK).
 - rating: integer, not null, rating given by a user to a specific place.
 - placeID: unique integer, represents the ID of the place (FK).
 - userID: unique integer, represents the ID of the user (FK).

5. comments

- commentID: unique integer with auto increment(PK).
- comment_date: Datetime, not-null, represents the date of the comment.
- comment: string, not-null, content of the comment given by a user to a specific place.
- placeID: unique integer, represents the ID of the place (FK).
- userID: unique integer, represents the ID of the user (FK).

6. adv

- adID: unique integer with auto increment(PK).
- title: string, not-null, represents the title of the advertisement.
- adv_content: string, not-null, information about the Advertisement.
- start_date: Datetime, the time the advertisement starts.
- end_date: Datetime, the time the advertisement ends.
- image: string, a picture of the Advertisement.
- added: string, suggested user advertisement.
- placeID: unique integer, represents the ID of the place that has the advertisement (FK).
- userID: unique integer, represents the ID of the user that suggests the advertisement (FK).

7. events

- eventID: unique integer with auto increment(PK).
- eventtype: string, not-null, represents the Type of the event.
- title: string, not-null, represents the title of the event.
- event_date: datetime, not-null, represents the date of the Event.
- event_location : string, not-null, location information for specific event.
- description: string, the content of the event.
- image: string, a picture of the event.

8. Notification

- notificationID: unique integer with auto increment(PK).
- notificationType: string, not-null, represents the Type of notification ("new place" or "advertisement added").
- notification_date: datetime, not-null, represents the date of the notification.
- notification_content: string, not-null, the content of the notification message.
- adID: unique integer, represents the ID of the advertisement (FK).
- placeID: unique integer, represents the ID of the place (FK).

9. branch

- branchID: unique integer with auto increment (PK).
- branch_location: string, the location of a specific place.
- placeID: unique integer, represents the ID of the place (FK).

10. phone_number

- phoneID: unique integer with auto increment (PK).
- phoneType: string, the type of the phone.
- Pnumber : integer, the content number of a specific place.
- placeID: unique integer, represents the ID of the place (FK).

Note: The `auth_user`, `auth_group`, `auth_membership`, `auth_permission`, `auth_event`, and `auth_cas` are authentication and authorization-related models that are a crucial part of the web2py infrastructure, providing a comprehensive and flexible system for managing user accounts, permissions, and security-related events within your web application.

3.8 Database Tables

Attribute	Type	PK	FK	Not-Null	Unique
id	int	yes		yes	yes
first_name	string				
last_name	string				
email	string				
password	string				
registration_key	string				
reset_password_key	string				
registration_id	string				

Table 3.8.1 `auth_user` table description

Attribute	Type	PK	FK	Not-Null	Unique
placeID	int	yes		yes	yes
name	string			yes	
image	string				
fb_acc	string				
ig_acc	string				
website	string				
added	string				
categoryID	int		yes		yes
userID	int		yes		yes

Table 3.8.2 Place table description

Attribute	Type	PK	FK	Not-Null	Unique
categoryID	int	yes		yes	yes
name	string			yes	
image	string				

Table 3.8.3 Category table description

Attribute	Type	PK	FK	Not-Null	Unique
ratingID	int	yes		yes	yes
rating	int			yes	
userID	int		yes		yes
placeID	int		yes		yes

Table 3.8.4 Ratings table description

Attribute	Type	PK	FK	Not-Null	Unique
commentID	int	yes		yes	yes
comments	string			yes	
comment_date	datetime			yes	
userID	int		yes		yes
placeID	int		yes		yes

Table 3.8.5 Comments table description

Attribute	Type	PK	FK	Not-Null	Unique
adID	int	yes		yes	yes
title	string			yes	
adv_content	string			yes	
start_date	datetime				
end_date	datetime				
image	string				
added	string				
userID	int		yes		yes
placeID	int		yes		yes

Table 3.8.6 Adv table description

Attribute	Type	PK	FK	Not-Null	Unique
eventID	int	yes		yes	yes
eventtype	string			yes	
title	string			yes	
event_date	datetime			yes	
event_location	string			yes	

description	string				
image	string				

Table 3.8.7 Events table description

Attribute	Type	PK	FK	Not-Null	Unique
notificationID	int	yes		yes	yes
notificationType	string			yes	
notification_date	datetime			yes	
notification_content	string			yes	
adID	int		yes		yes
placeID	int		yes		yes

Table 3.8.8 notification table description

Attribute	Type	PK	FK	Not-Null	Unique
branchID	int	yes		yes	yes
branch_location	string			yes	
placeID	int		yes		yes

Table 3.8.9 branch table description

Attribute	Type	PK	FK	Not-Null	Unique
phoneID	int	yes		yes	yes
phoneType	string			yes	
Pnumber	int			yes	
placeID	int		yes		yes

Table 3.8.10 phone_number table description

Chapter 4 Implementation

4.1 Overview

In this chapter, we will discuss the techniques used to build user interfaces and explain the basic user interfaces on the website.

4.2 Software environment and tools:

- **Bootstrap:**
It is a CSS framework that provides many ready-to-use designs and components.

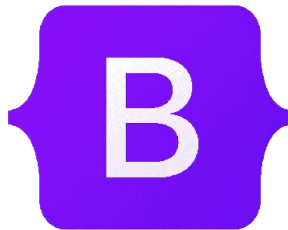


Figure 4.2.1 Bootstrap logo

- **Code Visual Studio:**
It is a code editor developed by Microsoft for the Linux , macOS and Windows operating systems. Includes support for debugging, Git and GitHub control, syntax highlighting, and smart code completion.

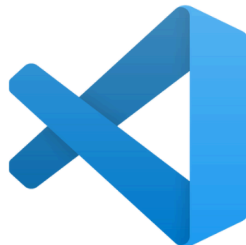


Figure 4.2.2 VS code logo

- **Web2py :**

Open-source web application framework written in the Python programming language. Web2py allows web developers to program dynamic web content using Python.



Figure 4.2.3 Web2py logo

- **Postman:**

Is an application that allows the testing of web APIs.



Figure 4.2.4 Postman logo

- **Rasa:**

Open-source tool that uses natural language understanding to develop AI-based chatbots. It provides a framework that can be used to create chatbots with minimal coding skills.

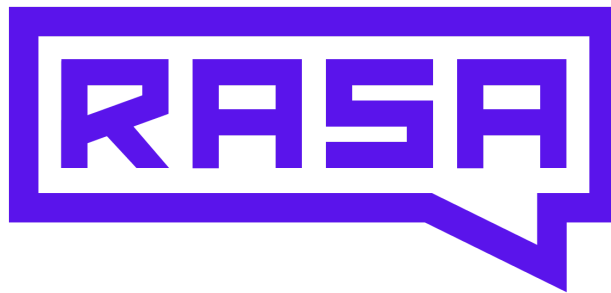


Figure 4.2.5 Rasa logo

- **MySQL:**

Open-source relational database management system (RDBMS).

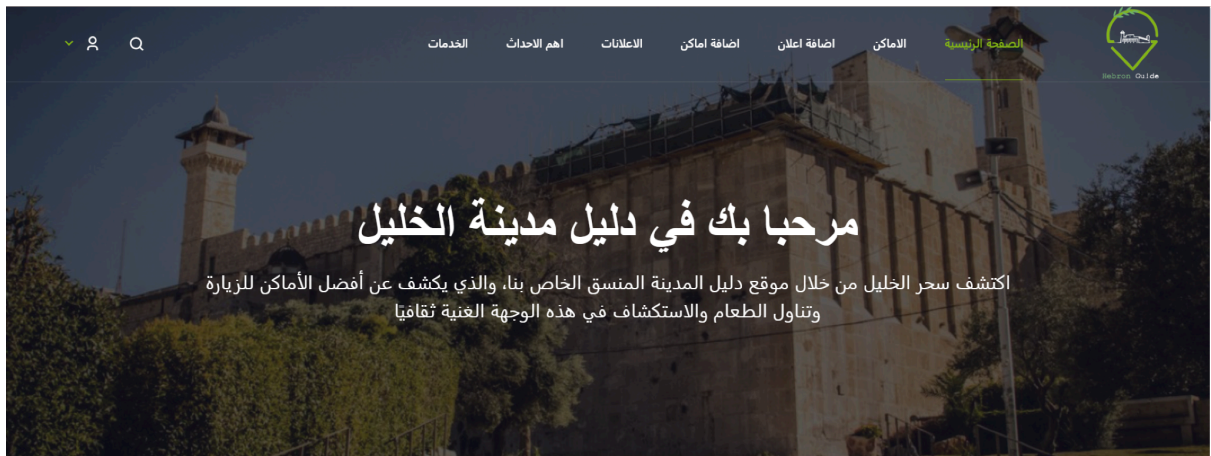


Figure 4.2.6 MySQL logo

4.3 User interface

4.3.1 Home page

The main page of the site displays information about the site and what are the most important services it provides. The most important events that occur in the city of Hebron are also displayed, as well as advertisements for various places. Through it, the user can communicate with the assistant to answer his questions.



من نحن

مرحبا بك في دليل مدينة الخليل

دليل مدينة الخليل هو عبارة عن منصة شاملة تعرض مجموعة متنوعة من المؤسسات في الخليل، بما في ذلك الجامعات والصيدليات والمستشفيات وغيرها. من خلال تقديم معلومات مفصلة حول كل فئة، يصبح هذا الموقع مصدراً لا غنى عنه لكل من السكان المحليين والزوار الذين يسعون إلى التنقل في المدينة بسهولة.



Figure 4.3.1 Home page

4.3.2 Registration page

The new user must fill out all information as shown. Which allows him to enjoy additional services such as adding advertisements and adding new places to the site.

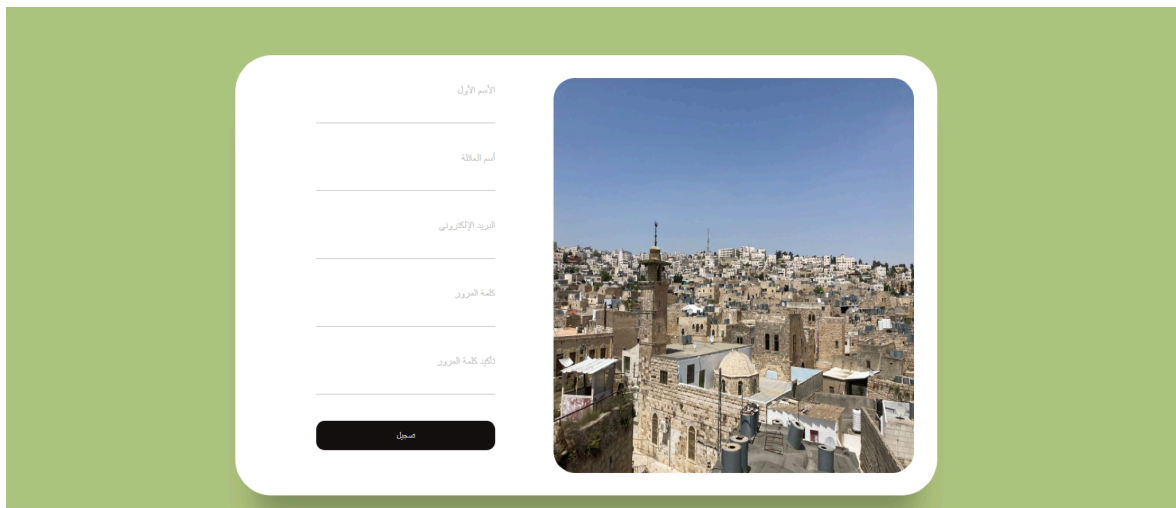


Figure 4.3.2 Registration page

4.3.3 Login page

Through this page, users can log in to the website.



Figure 4.3.3 Login page

4.3.4 Event section

Through it , users can stay informed about upcoming events in the city with a diverse range of event categories, including social, political, cultural, and more.



Figure 4.3.4 Event section

4.3.5 Advertisement section

Through it, the user can view special advertisements for different places in the city of Hebron.



Figure 4.3.5 Advertisement section

4.3.6 Category page

Through it, the main categories containing places in the city of Hebron are displayed.

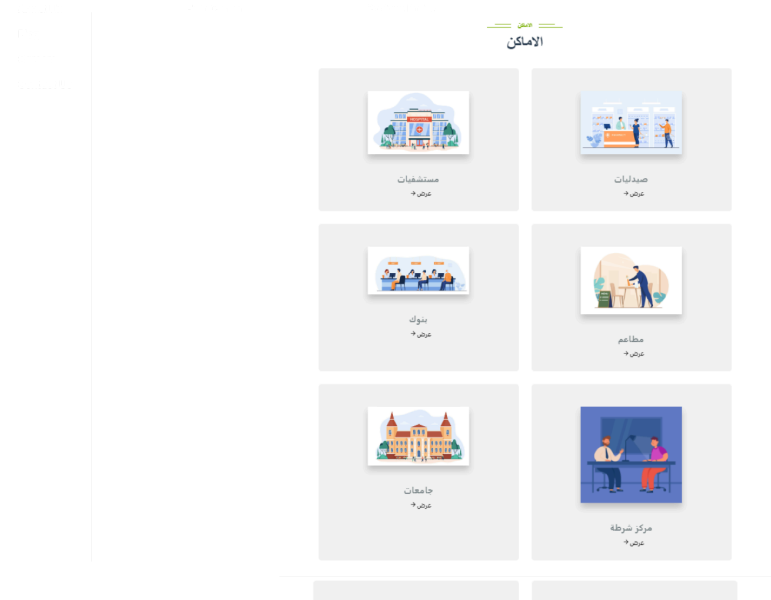
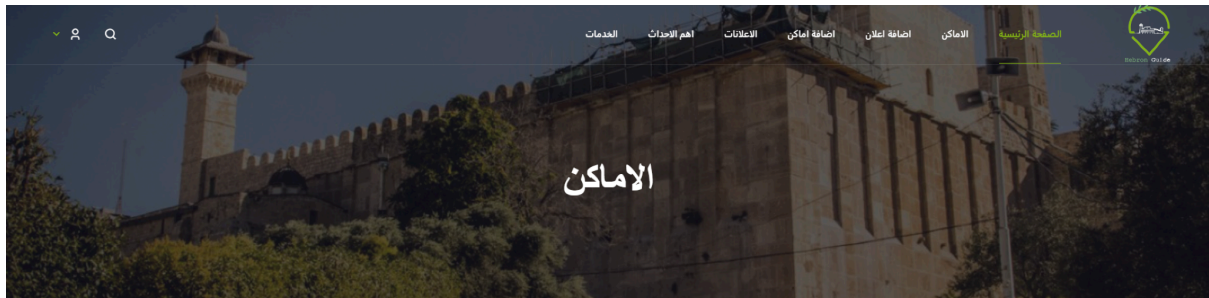


Figure 4.3.6 Category page

4.3.7 Places page

This page views a collection of places that fall under a specific category. Through it, users choose the specific place to display all information about it.

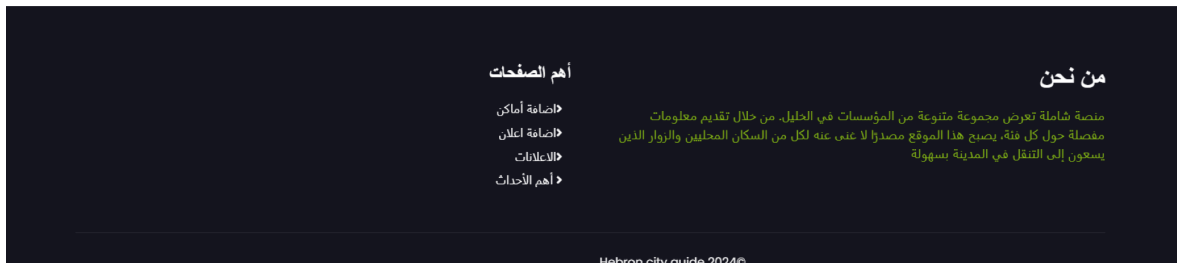


Figure 4.3.7 Places page

4.3.8 View place page

Through it, all information about the place is displayed, including the website, social networking sites, contact numbers and more. All reviews related to this place, including comments and ratings, are also displayed. It also allows registered users to add a comment and rate to the place.

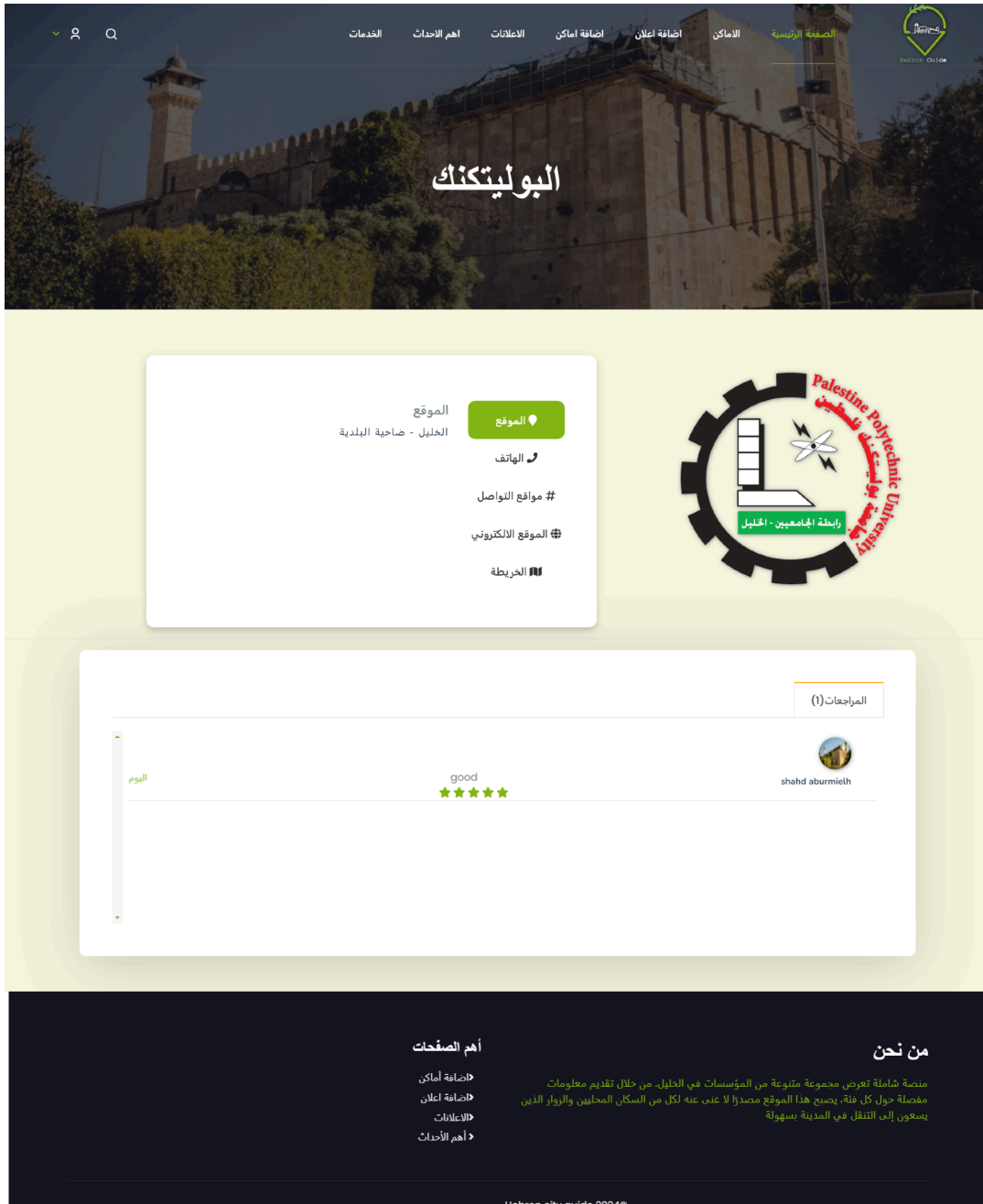


Figure 4.3.8 View place page

4.3.9 Add advertisement page

Through it, registered users are allowed to add a new advertisement to the site by filling out all the correct information for the advertisement.

The screenshot shows the 'Add Advertisement' page. At the top, there is a navigation bar with links for 'الخدمات' (Services), 'اهم الاحداث' (Important Events), 'الاعلانات' (Advertisements), 'اضافة اماكن' (Add Locations), 'اضافة اعلان' (Add Advertisement), 'الاماكن' (Locations), and 'الصفحة الرئيسية' (Home Page). The main heading is 'إضافة اعلانات' (Add Advertisements) and the sub-heading is 'إضافة إعلان جديد' (Add New Advertisement). The form contains the following fields:

- الحقول المطلوبة** (Required Fields):
- عنوان الاعلان** (Advertisement Title): A text input field.
- محتوى الاعلان** (Advertisement Content): A text area.
- وقت بدأ الاعلان** (Start Date): A date picker with the format 'mm/dd/yyyy --:-- --'.
- وقت انتهاء الاعلان** (End Date): A date picker with the format 'mm/dd/yyyy --:-- --' and a note 'يجب ان يكون اكبر من وقت بدأه' (Must be greater than the start date).
- اسم المكان** (Location Name): A text input field.
- اختار المكان** (Select Location): A dropdown menu.
- صورة الشعار** (Advertisement Image): A file upload field with a 'Choose File' button and the text 'No file chosen'.
- تأكيد** (Confirm): A green button at the bottom right.

Figure 4.3.9 Add advertisement page

4.3.10 Add place page

Through it, registered users are allowed to add a new place to the site by filling out all the correct information for the place.

The screenshot shows the 'Add Place' page on a website. The page has a dark header with navigation links: 'الاماكن', 'الصفحة الرئيسية', 'اضافة اعلان', 'اضافة اماكن', 'الاعلانات', 'اهم الاحداث', and 'الخدمات'. Below the header is a large image of a building with the word 'الاماكن' (Places) overlaid. Underneath the image is a sub-header 'اضافة مكان جديد' (Add New Place). The main content area is a form with the following fields:

- اسم المكان (Place Name) - Required field
- الفئات (Categories) - Dropdown menu
- الموقع (Location) - Required field
- رقم الهاتف (Phone Number)
- رقم المحمول (Mobile Number)
- صورة الشعار (Logo Image) - File upload section with 'No file chosen' and 'Choose File' buttons
- Facebook
- Instagram
- موقع إلكتروني (Electronic Site)

At the bottom right of the form is a green 'تأكيد' (Confirm) button.

Figure 4.3.10 Add place page

4.3.11 Profile page

Through it, the most important information related to the user is displayed, such as the number of advertisements, places, comments he added to the site, and his registration information.

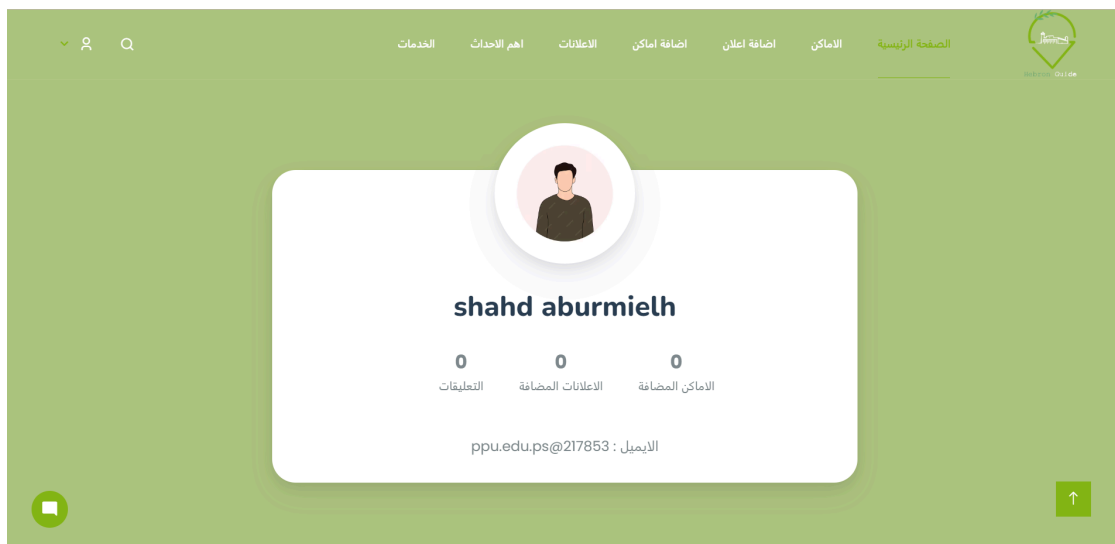


Figure 4.3.11 Profile page

4.3.12 Notification section

Through it, the system administrator can view all notifications that contain information about adding a place or adding a new advertisement by registered users.

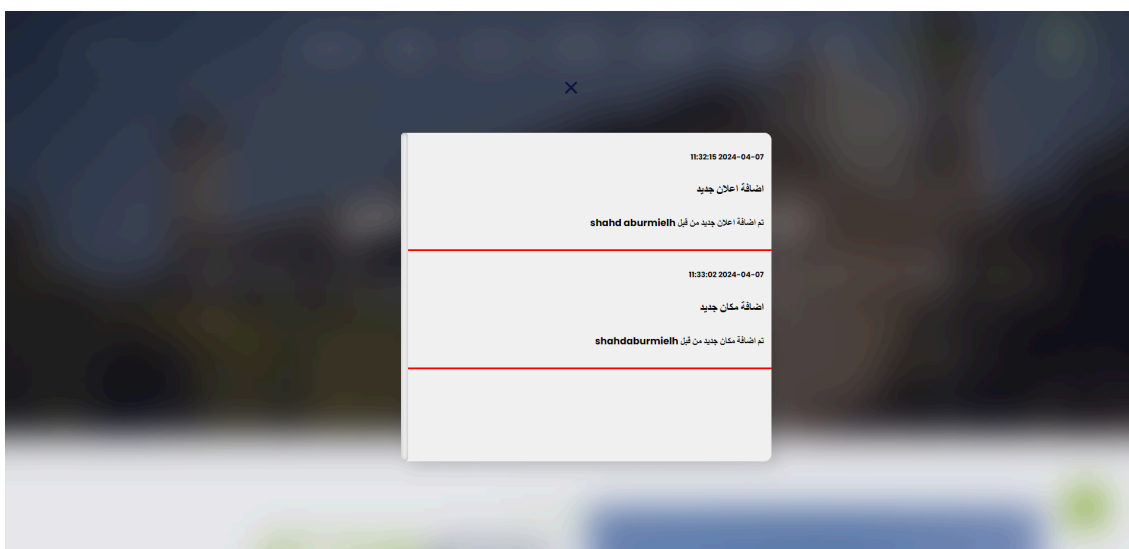


Figure 4.3.12 Notification section

4.3.13 Add category page

Through it, the system administrator can add a new category to the site by filling in its information

فئات

إضافة فئة جديد

تفاصيل الفئة

- الحقول المطلوبة
- اسم الفئة

صورة الشعار

No file chosen Choose File

تأكيد

Figure 4.3.13 Add category page

4.3.14 Add event page

Through it, the system administrator can add a new event to the site by filling in its information.

اهم الاحداث الاعلانات اضافة حدث اضافة فئات اضافة اماكن اضافة اعلان الاماكن الصفحة الرئيسية الاسم

اضافة احداث

الاسم

اضافة حدث جديد

تفاصيل الحدث

• الخقول المطلوبة

نوع الحدث

عنوان الحدث

وقت الحدث

موقع الحدث

وصف الحدث

صورة

No file chosen Choose File

تأكيد

Figure 4.3.14 Add event page

4.3.15 Chatbot Section

Through it, the user can conduct a direct chat by asking questions about the website , places, information about them, and more. Through it, questions are answered to help the user reach what he wants in the shortest time.

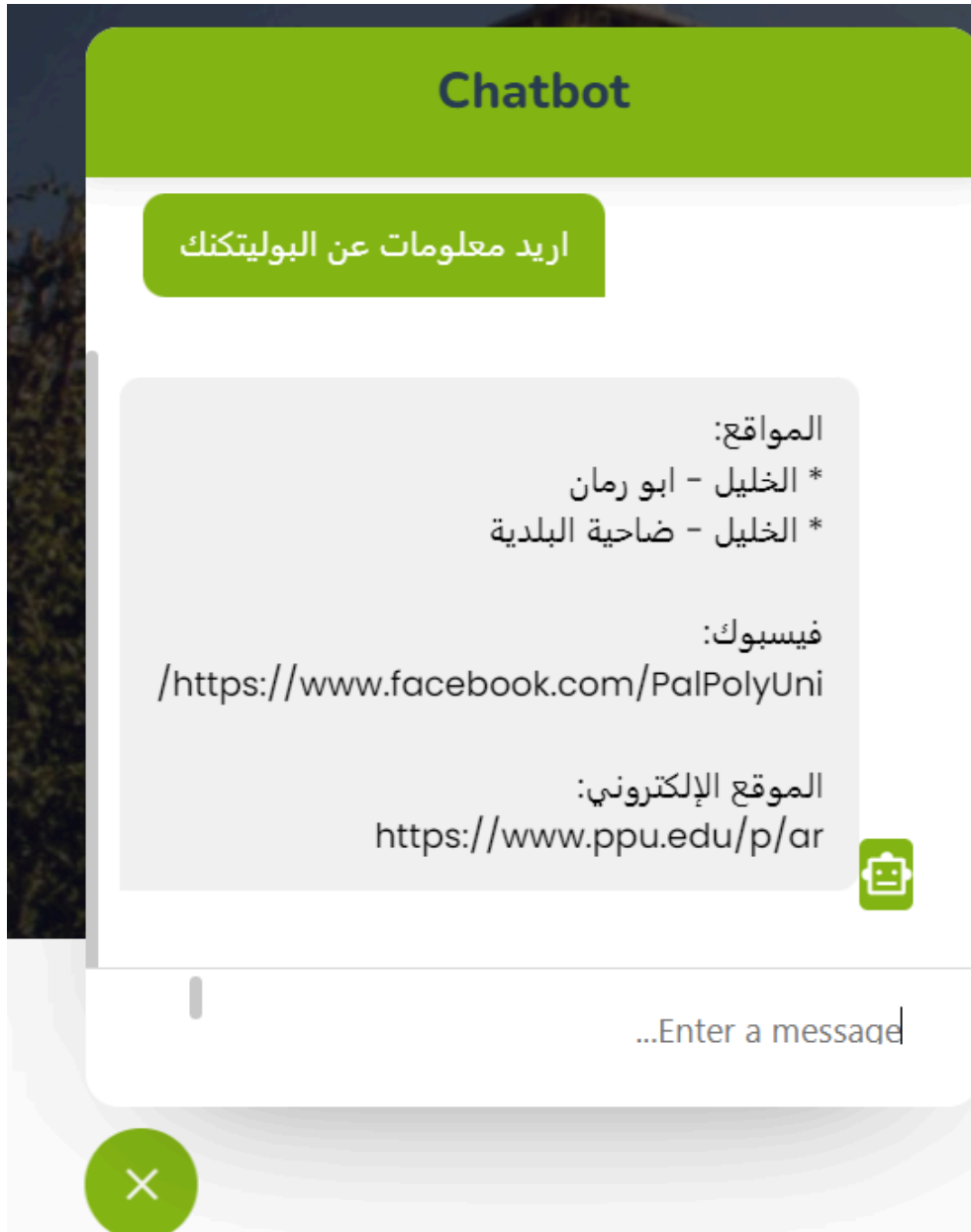


Figure 4.3.15 Chatbot section

4.4 Rasa framework files

The framework has a set of data structures that allows the user to input new data to train models, or even to generalize to unseen conversation paths the package is based on .yaml files. [5]

➤ nlu.yml:

- **Entities** : The important keywords that an assistant should take note of. For example, the message ‘My name is Maher’ has the name ‘Maher’ in it. An assistant should extract the name and remember it throughout the conversation to keep the interaction natural.
- **Intents** :In a given user message, the thing that a user is trying to convey or accomplish (e.g., greeting, specifying a location).
- **Utterances** : Anything a user says. A single utterance is an entire sentence passed as input to the Chatbot to intent. By definition, an utterance holds an intent and could potentially include one or more entities.

```
intent: place_information
examples: |
- [البوليتكنك] اخبرني عن (place)
- [البوليتكنك] اعطيني معلومات عن (place)
- [البوليتكنك] (place)
- [البوليتكنك] معلومات عن (place)
- [البوليتكنك] ما هي المعلومات المتاحة عن (place)
- [البوليتكنك] ممكن تعطيني التفاصيل عن (place)
- [البوليتكنك] اريد معرفة المزيد عن (place)
- [البوليتكنك] ممكن تشرح لي عن (place)
- [البوليتكنك] اعطني بعض المعلومات عن (place)

intent: place_location
examples: |
- [البوليتكنك] اريد موقع (place)
- [البوليتكنك] اين تقع (place)
- [البوليتكنك] وين (place)
- [البوليتكنك] موقع (place)
- [البوليتكنك] اريد مكان (place)
- [البوليتكنك] مكان (place)
- [البوليتكنك] وين موقع (place)
- [البوليتكنك] وين مكان (place)
```

Figure 4.4 Nlu.yml file content

- **Stories.yml** : is a representation of a conversation between a user and an AI assistant, converted into a specific format where user inputs are expressed as intents (and entities when necessary), while the assistant's responses and actions are expressed as action names.

```
! stories.yml x
data > ! stories.yml
1  version: "3.1"
2
3  stories:
4  - story: greeting
5    steps:
6    - intent: greet
7    - action: utter_greet
8
9  - story: ask for place comments
10  steps:
11  - intent: greet
12  - action: utter_greet
13  - intent: place_comments
14  - action: get_place_comments
15
16 - story: ask for place
17  steps:
18  - intent: greet
19  - action: utter_greet
20  - intent: place_information
21  - action: get_place_information
22
23 - story: ask for place around
24  steps:
25  - intent: greet
26  - action: utter_greet
27  - intent: place_around
28  - action: get_place_around
29
```

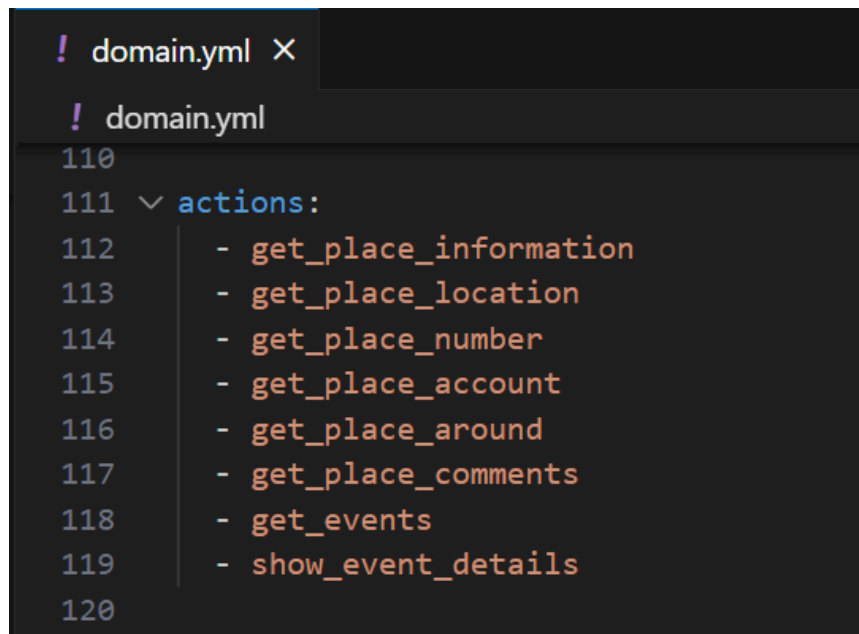
Figure 4.4 Stories.yml file content

- **Domain.yml** : The domain defines the universe in which your assistant operates. It specifies the intents, entities, slots, responses, forms, and actions your bot should know about. It also defines a configuration for conversation sessions.

```
! domain.yml x
! domain.yml
1  version: "3.1"
2
3  ✓ intents:
4  - greet
5  - goodbye
6  - affirm
7  - deny
8  - mood_great
9  - mood_unhappy
10 - bot_challenge
11 - thanks
12 - place_information
13 - place_location
14 - place_number
15 - place_account
16 - place_around
17 - place_comments
18 - events
19 - select_event
20
21 entities:
22 - place
23 - place_around
24 - acc_type
25 - category
26 - event_date
27 - event_id
28
```

Figure 4.4 Domain.yml file content

- **Actions** : After each user message, the model will predict an action that the assistant should perform next. This page gives you an overview of the different types of actions you can use.
 - **actions/actions.py**: All programming is here. The action class describes how the Chatbot fulfills various intents. Each action has access to the tracker, which contains data about the conversation, and implements a run function which holds the logic for carrying out said action.



```
! domain.yml X
! domain.yml
110
111  actions:
112     - get_place_information
113     - get_place_location
114     - get_place_number
115     - get_place_account
116     - get_place_around
117     - get_place_comments
118     - get_events
119     - show_event_details
120
```

Figure 4.4 Actions of rasa

Chapter 5 Testing

5.1 Overview

In the stage of testing the system, we make sure that the system works correctly without any problems, and we also make sure that the functional and non-functional requirements of the project are completed, and that the system works accurately and quickly in completing the tasks. The stage of checking the system comes after the design and implementation of the system. The method of checking the system will be explained in this chapter.

5.2 Validation

All information entered in all fields in the website is checked to ensure that the data entered by the user matches all conditions as follows:

- Customize the field to match the entry
- The process will not be executed if wrong data is entered.
- The operation is not performed if the fields are empty
- Ensure that there are actual users in the database.

5.3 System check

The system units were fully tested and ensured how they worked. The result of the examination was successful. The following tables review the testing we have done:

1. Register

#	Case	Input	Expected output	Obtained output	Pass/Fail
1	Empty fields	first name = last name = email = password = confirm password =	Error Message	As Expected	Pass
2	Correct information	first name = shahd last name = aburmielh email = shahd@gmail.com password = shahd1234. confirm password = shahd1234.	Successful register	As Expected	Pass
3	Email already exist	first name = shahd last name = aburmielh email = 217853@ppu.edu.ps	Error Message	As Expected	Pass

		password = shahd1234. confirm password = shahd1234.			
4	Invalid email	first name = shahd last name = aburmielh email = shahd.gmail.com password = shahd1234. confirm password = shahd1234.	Error Message	As Expected	Pass
5	Password does not matching system validation (contain at least 8 characters)	first name = shahd last name = aburmielh email = shahd.gmail.com password = 1234 confirm password = 1234	Error Message	As Expected	Pass
6	unmatched password between password and confirm password	first name = shahd last name = aburmielh email = shahd@gmail.com password = shahd1234. confirm password = shahd12345.	Error Message	As Expected	Pass

Table 5.3.1 Register process

Here are the register validation test case :

1. Case number 1 :

The screenshot shows a registration form with the following fields and error messages:

- الأسم الأول** (First Name): Error: **الاسم الأول لا يمكن أن يتفرقه فاصح** (First name cannot contain a space).
- اسم العائلة** (Last Name): Error: **اسم العائلة لا يمكن أن يتفرقه فاصح** (Last name cannot contain a space).
- البريد الإلكتروني** (Email): Error: **لا يمكن أن يكون البريد الإلكتروني فارغاً** (Email cannot be empty).
- كلمة المرور** (Password): Error: **كلمة السر لا يمكن أن تتفرقه فاصحة** (Password cannot contain a space).
- تأكيد كلمة المرور** (Confirm Password): Error: **كلمة السر لا يمكن أن تتفرقه فاصحة** (Confirm password cannot contain a space).

At the bottom of the form is a **مسجل** (Register) button. The background image shows a cityscape with a prominent minaret.

2. Case number 2 :


الاسم الأول

اسم العائلة

البريد الإلكتروني

كلمة المرور

تأكيد كلمة المرور



3. Case number 3 :

الاسم الأول


اسم العائلة

البريد الإلكتروني

البريد الإلكتروني موجود بالفعل

كلمة المرور

تأكيد كلمة المرور



4. Case number 4 :

الاسم الأول

اسم العائلة


البريد الإلكتروني

ليس بريدًا إلكترونيًا صالحًا

كلمة المرور

تأكيد كلمة المرور

تسجيل



5. Case number 5 :

الاسم الأول

اسم العائلة

البريد الإلكتروني


كلمة المرور

كلمة السر يجب أن تحتوي على 8 أحرف على الأقل وتتضمن
أرقامًا بحروف

تأكيد كلمة المرور

كلمة السر يجب أن تحتوي على 8 أحرف على الأقل وتتضمن
أرقامًا بحروف

تسجيل



6. Case number 6 :

الاسم الأول

✓

اسم العائلة

✓

البريد الإلكتروني

✓

كلمة المرور


✓

تأكيد كلمة المرور

❗

كلمة السر غير متطابقة

تسجيل



2. Login

#	Case	Input	Expected output	Obtained output	Pass/Fail
1	Empty fields	email = password =	Error Message	As Expected	Pass
2	Correct information	email = shahd@gmail.com password = shahd1234.	Successful login	As Expected	Pass
3	Invalid email	email = shahd.gmail.com password = shahd1234.	Error Message	As Expected	Pass

Table 5.3.2 Login process

Here are the login validation test cases :

1. Case number 1:



The screenshot shows a login form with the following elements:

- Header: ليس لديك حساب؟ سجل حساب جديد
- Field 1: البريد الإلكتروني (Email) with a red error icon and message: لا يمكن أن يكون البريد الإلكتروني فارغاً
- Field 2: كلمة المرور (Password) with a red error icon and message: كلمة السر لا يمكن أن تكون فارغة
- Button: تسجيل الدخول (Login)
- Footer: نسيت كلمة المرور (Forgot Password)

The background image shows a cityscape with a prominent stone building and a minaret under a clear blue sky.

2. Case number 2 :

ليس لديك حساب؟ سجل حساب جديد

البريد الإلكتروني

shahd@gmail.com

كلمة المرور

.....

سجل الحساب

نسيت كلمة المرور



3. Case number 3 :

ليس لديك حساب؟ سجل حساب جديد

البريد الإلكتروني

shahd2@gmail.com

البريد الإلكتروني غير موجود

كلمة المرور

.....

سجل الحساب

نسيت كلمة المرور



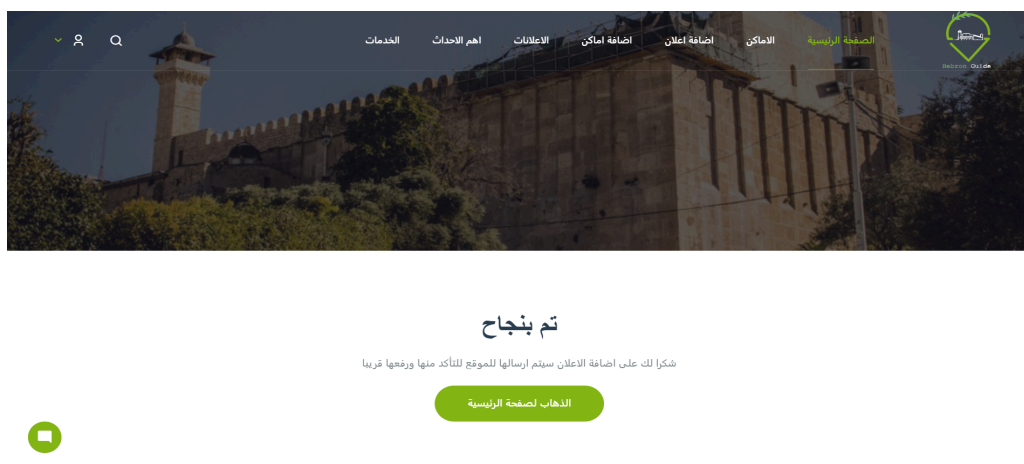
3. Add new advertisement

#	Case	Input	Expected output	Obtained output	Pass/Fail
1	Correct Information	fill all fields with correct information.	Successful	As Expected	Pass
2	Existed advertisement	fill all fields of information.	Error Message	As Expected	Pass

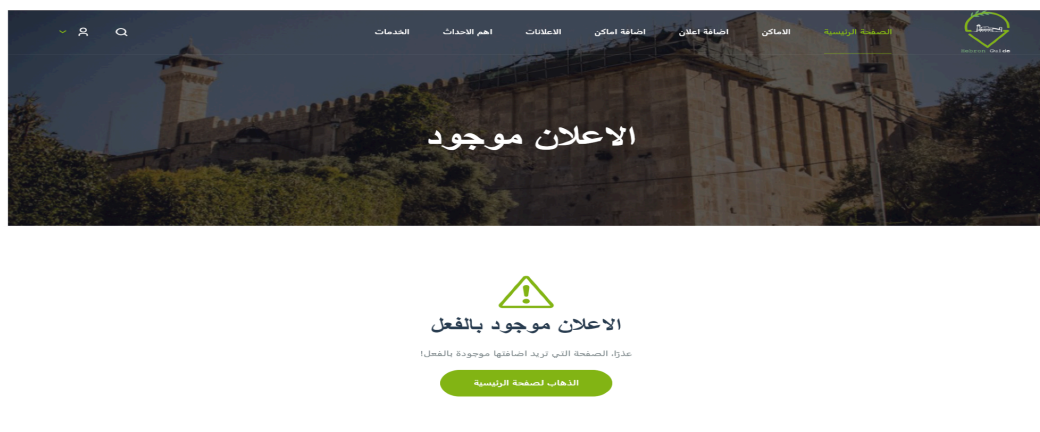
Table 5.3.3 Add new advertisement process

Here are the add advertisement validation test cases :

1. Case number 1 :



2. Case number 2:



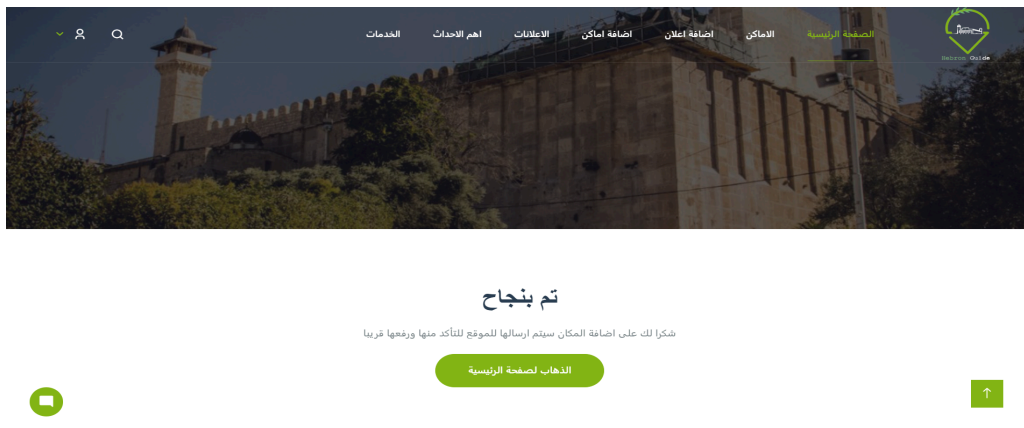
4. Add new place

#	Case	Input	Expected output	Obtained output	Pass/Fail
1	Correct Information	fill all fields with correct information.	Successful	As Expected	Pass
2	Existed place	fill all fields of information.	Error Message	As Expected	Pass

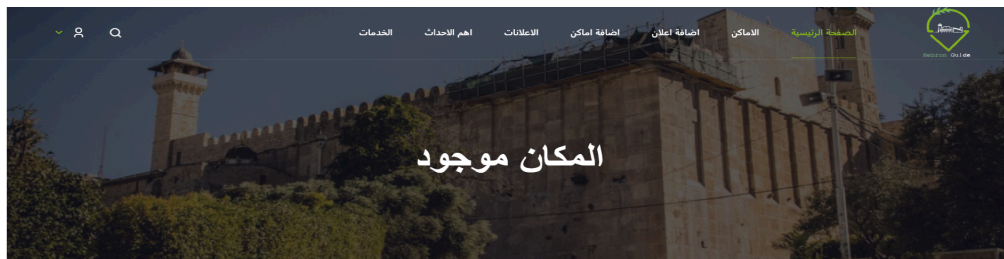
Table 5.3.4 Add new place process

Here are the add place validation test cases :

1. Case number 1 :



2. Case number 2 :



5. Chatbot test

#	Case	Input	Expected output	Obtained output	Pass/Fail
1	Greeting	"message" : "مرحبا"	Successful	As Expected	Pass
2	Asking for specific place	"message" : "أخبرني عن "البوليتكنك"	Successful	As Expected	Pass
3	Asking for place accounts	"message" : "اريد مواقع "التواصل ل البوليتكنك"	Successful	As Expected	Pass
4	Asking for place content numbers	"message" : "اريد ارقام "التواصل ل البوليتكنك"	Successful	As Expected	Pass
5	Asking for place location	"message" : "اريد موقع "البوليتكنك"	Successful	As Expected	Pass
6	Asking for places in specific location and category	"message" : "اريد مطاعم في" " رأس الجورة"	Successful	As Expected	Pass
7	Asking for place comments	"message" : "اريد أهم" "المراجعات ل البوليتكنك"	Successful	As Expected	Pass
8	Asking for upcoming events	"message" : "أخبرني عن" "الاحداث القادمة في الخليل"	Successful	As Expected	Pass
9	Asking for upcoming events of specific month	"message" : "أخبرني عن" "الاحداث القادمة في شهر 6"	Successful	As Expected	Pass

Table 5.3.5 Chatbot test

In this part we used Postman application to test the various functionalities and uses cases of chatbot. Here are the Chatbot test cases:

- Case number 1 :

```
1 {
2   "message": "مرحبا"
3 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 62 ms Size: 502 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "recipient_id": "default",
4     "text": "مرحبا بك في دليل مدينة الخليل. كيف يمكنك مساعدتك؟"
5   }
6 ]
```

- Case number 2 :

```
1 {
2   "message": "اخبرني عن البوليتكنك"
3 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 2.18 s Size: 756 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "recipient_id": "default",
4     "text": "إنستغرام: https://www.facebook.com/PalPolyUni/\n\nفيسبوك: [الخليل - ابورمان, 'الخليل - ضاحية البلدية': الموقع]\n\nالموقع الإلكتروني: https://www.ppu.edu/p/ar"
5   }
6 ]
```

- Case number 3 :

```
1 {
2   "message": "اريد مواقع التواصل البوليتكنك"
3 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 2.26 s Size: 394 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "recipient_id": "default",
4     "text": "إنستغرام: https://www.facebook.com/PalPolyUni/\n\nفيسبوك: None"
5   }
6 ]
```

- Case number 4 :

```
1 {
2   "message": "اريد ارقام التواصل البوليتكنك"
3 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 2.25 s Size: 434 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "recipient_id": "default"
```

- Case number 5 :

```

1 {
2   "message": "أريد موقع البوليتكنك"
3 }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 2.21 s Size: 528 B Save as example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "recipient_id": "default",
4     "text": "الخليل - صاحبة البلدية - المواقع:"
5   },
6   {
7     "recipient_id": "default",
8     "text": "الخليل - ايورمان -"
9   }
10 ]

```

- Case number 6 :

```

1 {
2   "message": "أريد مطاعم في رأس الجورة"
3 }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 2.25 s Size: 1.75 KB Save as example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "recipient_id": "default",
4     "text": "الاماكن الموجودة في رأس الجورة في فئة مطاعم:"
5   },
6   {
7     "recipient_id": "default",
8     "text": "بونجورن\النشام\nالذيرة\nليتشر\nبلاك مكدونالدز\nالدولار\nطابون الريفان\nزوارن\nمستر بكري\nبيت الباسمين\nسيجافرينو\nمطعم\nحكاية بحر\nكعبير\nnoj restaurant\nمطعم القديم\nمطعم قصر الباشا - الخليل\nمطعم وكافيه فيلان\nدوز بيرغر\nحليويات بردايبين\nقيعين\nمطعم وكافيه اورنج\nمشاري القمة\nومشاري غيث"
9   }
10 ]

```

- Case number 7 :

```

1 {
2   "message": "اريد اهم المراجعات ل اليوليتكنك"
3 }

```

Body Cookies Headers (6) Test Results 200 OK 2.20 s 464 B Save as example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "recipient_id": "default",
4     "text": "جيد جدا\n:تعليقات المستخدمين عن اليوليتكنك"
5   }
6 ]

```

- Case number 8 :

```

1 {
2   "message": "اخبرني عن الاحداث القادمة في الخليل"
3 }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 2.21 s Size: 973 B Save as example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "recipient_id": "default",
4     "text": "ايام اليوليتكنك - اكميو الخليل - الاحداث القادمة",
5     "buttons": [
6       {
7         "title": "اعرض التفاصيل عن اكميو الخليل",
8         "payload": "/show_event_details?event_id=27"
9       },
10      {
11        "title": "اعرض التفاصيل عن ايام اليوليتكنك",
12        "payload": "/show_event_details?event_id=28"
13      }
14    ]
15  }
16 ]

```

- Case number 9:

```

1 {
2   "message": "اخبرني عن الاحداث القادمة في شهر 6"
3 }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 2.20 s Size: 743 B Save as example

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "recipient_id": "default",
4     "text": " : الاحداث التي تحدث في شهر 6 "
5   },
6   {
7     "recipient_id": "default",
8     "text": " اكميو الخليل - "
9   },
10  {
11    "recipient_id": "default",
12    "buttons": [
13      {
14        "title": "اظهر التفاصيل عن اكميو الخليل",
15        "payload": "/show_event_details?event_id=27"
16      }
17    ]
18  }
19 ]

```

Chapter 6 Conclusion and Future work

6.1 Conclusion

In the end, we have developed a website that helps users navigate Hebron City with ease. This Hebron City Guide website enables users to view detailed information about various places within the city, including local attractions, businesses, and services. Additionally, users can stay informed about upcoming events and advertisements relevant to their interests. It also empowers users to contribute by adding new places and advertisements. The aim of this project is to make this platform a reference for accessing information about the city of Hebron for both visitors and residents. By offering comprehensive and up-to-date details, we hope to enhance the user experience and promote greater engagement with the city's offerings.

6.2 Future work

In the future, we look forward to adding important features to the system, such as expanding the content and information available on the website, ensuring it remains comprehensive and up-to-date. This will involve updating the details on places, events, and advertisement as well as adding new content as the city evolves. Also, we will be working on enhancing the conversational abilities of the virtual assistant, allowing it to engage in more natural dialogues with users. This includes improving its ability to understand and interpret complex queries, as well as its capacity to provide personalized recommendations and suggestions based on the user's preferences and previous interactions.

6.3 References

- [1]: R. Kawaneen and M. Maharmeh, "Dalil Al-Katroni Tajari wa Harfi Khas bi Madyinat Al-Khalil" [Electronic Commercial and Artisan Guide for Hebron City], Bachelor's thesis, College of Information Technology and Computer Engineering, Palestine Polytechnic University, Hebron, Palestine, 2011. [Online]. Available: <https://scholar.ppu.edu/handle/123456789/7667>.
- [2]: C. Subramanya, "Application Architecture — Understanding MVC (Model-View-Controller)," Medium, 2023. [Online]. Available: <https://subramanya-c.medium.com/application-architecture-understanding-mvc-model-view-controller-fda0879839e1>.
- [3]: M. Shrestha, "What is the 3-Tier Architecture?" Medium, 2020. [Online]. Available: <https://medium.com/@shrestha.matina.20/what-is-the-3-tier-architecture-4520522e0720>.
- [4] B. Reddy, "What Is RASA?," Medium, 30-Aug-2023. [Online]. Available: <https://medium.com/@bharathreddy028/what-is-rasa-c6f2790a38b7>.
- [5]: SanchariGr, "Rasa Documentation," Medium, 2024. [Online]. Available: <https://rasa.com/docs/rasa/training-data-format>.