Eötvös Loránd University

Faculty of Informatics

Department of Algorithms and Applications

Global Museum Management Application: Enhancing Tourist Experience and City Cultural Promotion

Supervisor:

Dr. Szabó László Ferenc

Associate Professor, PhD

Author:

Murad Huseynov

Computer Science BSc

Budapest, 2024

EÖTVÖS LORÁND UNIVERSITY
FACULTY OF INFORMATICS

Thesis Registration Form

Student's Data:
Student's Name: Huseynov Murad
Student's Neptun code: RX15MW

Course Data:
Student's Major: Computer Science BSc

I have an internal supervisor

Internal Supervisor's Name: *Dr. Szabó László Ferenc*

Supervisor's Home Institution:	***Department of Algorithms and Applications***
Address of Supervisor's Home Institution:	***1117, Budapest, Pázmány Péter sétány 1/C.***
Supervisor's Position and Degree:	*Associate Professor and Chair, PhD with Habilitation*

Thesis Title: Global Museum Management Application: Enhancing Tourist Experience and City Cultural Promotion

Topic of the Thesis:
(Upon consulting with your supervisor, give a 150-300-word-long synopsis os your planned thesis.)

In the era of digital transformation, the intersection of technology and cultural experiences is increasingly crucial. This thesis proposes the development of a comprehensive web application designed to revolutionize the management and experience of museums on a global scale. This application, leveraging React.js for the frontend, Node.js for the backend, and Firebase for database management, will serve three primary user groups: administrators, tourists, and cities.

The application aims to streamline the process of managing museum data and enhance the tourist experience. Administrators will have the capability to add, modify, or delete city and tourist profiles, as well as manage a dynamic list of countries. The system will facilitate easy user registration, offering distinct signup pages for tourists and cities, requiring specific personal data and official information, respectively.

For cities, the application will allow the registration of museums, including detailed descriptions, available visiting days, ticket limitations, and categorization of tickets (adult, student, minor, and senior). Tourists, on the other hand, will have the ability to book tickets, view past bookings, and receive tickets with unique QR codes and optional PDF tickets via email.

The application's structure ensures efficiency in managing museum visits and provides an intuitive interface for users. It will not only facilitate the tourism experience but also aid cities in promoting and managing their cultural assets more effectively. The optional features, such as museum reviews and ratings, are considered additional enhancements that may be implemented post the main development phase.

This thesis aims to deliver a solution that not only simplifies museum management but also enriches the cultural experience for tourists worldwide, fostering a more connected and accessible global cultural landscape.

Budapest, 2023. 11. 14.

Acknowledgments

First of all, I would like to thank my supervisor, Dr. Szabó László Ferenc, for his continuous support throughout the semester. His all-important guidance and constant feedback have been a vital foundation of my research journey.

My 4-year undergraduate experience has been full of ups and downs, and to my family, who always motivated me no matter what and shared their constant love and support, I just want to say "Thank you!" I am who I am today thanks to you, your encouragement, and the sacrifices you have made for me.

To my friends, especially the ones who were together with me in my toughest moments in the past 4 years, I am beyond grateful for your belief in me and your help to overcome the challenges I faced. I cherish our fellowship and all of the memories we share together.

To conclude, it would not be possible to achieve my goals in this pursuit without my beloved ones. This work is not only the outcome of constant dedication but also a testament to the incredible support system I am fortunate to have. I am thoroughly grateful for every moment of encouragement.

Contents

Chapter 1

Introduction

1.1 Motivation

All in all, tourism has always been a growing industry. Especially after COVID-19 lockdowns were lifted globally, people have tended to explore new places and cultures more than ever. Museums are considered cornerstones of cultural attractions, holding a significant place in tourism, and benefiting from the elevation of the industry. Personally, I enjoy traveling and prefer to visit as many museums as possible in a city or town I visit; sometimes, an interesting museum is my reason for traveling to a particular destination. However, searching for museums can often be time-consuming as there is no centralized platform where users can browse countries and cities, list museums, acquire information, and book tickets. Even though popular museums have their own websites, this requires users to search for each museum individually, which an unfeasible approach most of the times.

From another perspective, cities could greatly benefit from the centralization of their museums in a single interface. On the one hand, this would boost the popularity of obscure museums and contribute to an equal distribution of tourists, consequently increasing overall demand for a destination. On the other hand, such a system could potentially help cities manage their marketing more effectively and reduce financial costs as museum websites and tracking would no longer be handled separately.

1.2 Summary and Description of the Project

This project aims to address all the issues mentioned above by providing a unique platform that groups museums under the cities in which they are located. To achieve the desired functionalities, the application allows registrations for cities and tourists. Additionally, there is an administrative login that has access to adding, modifying, and deleting city and tourist accounts. As a city registration should be done by an authority or an official from the local government, this registration can be successfully completed after adding a unique city ID that can be generated only by an administrator to keep the security policies.

1.2.1 City Registration

When logged in as a city, the following screens and functionalities will be accessible:

1. **City Dashboard:** This screen is the main screen for a city that welcomes the user after a successful login. Here, users can view already added museums as individual museums cards. They can also use a search bar to look for a specific museum. This page allows users to direct to pages for adding, editing, and deleting museums, and also navigate to pages dedicated to individual museums that will also be visible to tourists.

2. **Add Museum:** This is a 3-step comprehensive screen for registering museums. The first step is about entering textual data, such as museums name, short summary and history in two separate fields, operating times, address, phone number, website, and a cover image for the museum card. The application allows cities to divide tickets according to four social and age-related groups: adult, student, senior, minor. The second step requires providing ticket prices for each category and choosing a currency. By default, the application creates 30-minute slots between the opening and closing times and there is a limit that each tourist can book a ticket for. The final step is about setting open days, opening and closing times, and defining the slot limit.

3. **Edit Museum:** This screen can be used to edit an already added museum. For simplicity, the page has been designed to look exactly like Add Museum screen, except that the fields containing information regarding the museum will already be filled with the current data.

4. **Museum Details:** This screen is dedicated to viewing individual museums. The textual information that has been added during museum registration will be illustrated here. Apart from this, there are 3 buttons for including museum images on the page, particularly for the summary and history sections. To avoid overwhelming these sections with too many images, there is a last button to add images to a carousel. The screen also shows the address on a map where the museum location is pinpointed. This page is available for tourists as well, but city functionalities will not be available to them, while tourists will have their custom functions.

5. **Delete Museum:** This functionality allows for deletion of a museum in case it is closed or no longer operating to keep the tourists updated.

6. **Settings:** This page can be used to update registration data, passwords, or can be used to delete the account.

1.2.2 Tourist Registration

When logged in as a tourist, the following screens and functionalities will be accessible:

1. **Tourist Dashboard:** This page is the entry point for tourist logins. Users can see a list of countries that contains at least one registered city. They can click on a country to see the list of cities and consecutively, click on a city to view a list of museums. Users can click on a museum to access the details of museum, also click on *Show Weather* button to display weather statistics of the city. The tourist dashboard also contains a search bar that can be used for finding specific countries and/or cities.

2. **View Bookings:** Tourists can view their bookings if they are logged in. By default, the bookings will be sorted by date in descending order, but this can be changed to ascending order. Additionally, bookings can be sorted by museum names alphabetically. When clicked on a booking, the details will be shown in a separate page, including a unique QR code.

3. **Display City Weather:** This Page includes a 3-hour weather forecast for the next 5 days, displaying both actual and feels-like temperatures. To provide

a detailed insight, the reports are listed individually per hour, and the whole information is also summarized graphically.

4. **Museum Details:** The core structure of this page is similar to the *Museum Details* page on the city side, but tourists cannot access the buttons for adding visuals. Instead, they can see the images added by city registrars. Additionally, tourists can access pages for booking tickets, showing busy slot diagrams, adding, and viewing reviews from here.

5. **Book Ticket:** Booking a ticket is a 4-step process. Initially, users are required to choose at least one ticket from one of the categories (adult, student, senior, minor). To reach maximum user convenience, total and base prices per category are shown in the side. Second step is about selecting one of the available dates from calendar. Moving on, users will see the available time slots and number of free places in each slot. After choosing a slot, users can also select the option to receive a PDF ticket via email. The final step is checking the booking summary and completing the booking.

6. **Show Busy Slots:** To support tourists in planning their visits, it is possible to display which days of the week are busier than others through bar charts. There are two bar charts: one illustrates the number of bookings only. On the contrary, the other graph displays the data about total booked slots.

7. **Add Review:** Each tourist can provide feedback on the visited museums as well as give an estimate rating using stars. There are 5 stars in total where 0 star is the worst, and 5 stars is the best possible rating that can be given to a museum.

8. **View All Reviews:** All of comments provided to a museum are gathered on a single page where the name of the reviewer, the comment and rating are shown. The comments can be sorted by the day published or by the given stars.

9. **Settings:** Similar to the *Settings* page in city side, this page can be used for updating registration data, passwords, or to delete the account.

Chapter 2

System Requirements

2.1 Hardware Requirements

In order to run Global Museum Management application at basic level, users are required to have a device with the following parameters:

1. **Processor:** For smooth operations, it is recommended to have Intel Core i5 or similar processor that has at least 2.0 GHz speed.

2. **Memory:** In order to install the necessary dependencies, and run the application successfully, the device should have at least 10 GB of free disk space.

3. **RAM:** To handle multitasking effectively, a minimum of 2 GB RAM is needed.

2.2 Software Requirements

In terms of software requirements, the following specifications are essential:

1. **Operating System:** Windows 10 or later, macOS 10.0 or later should be installed.

2. **Browser:** As a web application, Global Museum Management runs in a browser. Thus, users need to have a modern browser installed on their devices.

The application has been created with React.js on front-end, Firebase and Node.js for back-end and database management. Before starting the application users should make sure that these tool have been installed. A step-by-step installation guide is provided in the next section.

2.3 Installation Guide

In order to prepare the environment to start using Global Museum Management application, please follow the steps below. The installation guide assumes non of the tools and dependencies has been installed in advance, so users can skip some instructions in case they have been fulfilled already. It is always recommended to install the latest versions of the software to avoid version-related issues.

1. Install Node.js from:

   `https://nodejs.org/en/download` [6]

2. An Integrated Development Environment (IDE) that supports Node.js is required to run the application. Visual Studio Code is one of the optimal options that has also been used in the development process. Install Visual Studio Code from:

   `https://code.visualstudio.com/download` [10]

3. The application is publicly available as a GitHub repository. To clone the repository, Git should be installed locally. Install Git from:

   `https://git-scm.com/downloads` [3]

4. Clone the application repository from GitHub:

   (a) Open a terminal or command prompt.

   (b) Navigate to directory that the application repository will be located in.

   (c) Clone the repository by running the command:

   ```
   git clone https://github.com/muradhuseynov1/museum-app
   ```

5. Install Dependencies:

   (a) On the project folder (global-museum-management/), run the following command:

   ```
   npm install
   ```

(b) The project folder contains 3 sub-folders, which are "museum-app1", "node-server", "functions". Navigate to each folder and run the same command:

```
npm install
```

As there are four package.json files in total for each service and tool, the command will install all the dependencies listed in these JSON files.

6. Final step is to run the application. For this, users should navigate to the project folder (global-museum-management/) using their terminal or command prompt, and run the following command:

```
npm start
```

The Global Museum Management application will open shortly after running the command, on user's default browser as a new tab.

2.4 Dependencies

Dependencies are essential software packages and libraries that the Global Museum Management application requires to provide its services. These services include the usage of external APIs, Firebase tools, and configuring the layout of the application.

The following dependencies have been used to develop the Global Museum Management application:

- "@date-io/date-fns": "3̂.0.0"
- "@emotion/react": "1̂1.11.4"
- "@emotion/styled": "1̂1.11.5"
- "@mui/lab": "5̂.0.0-alpha.167"
- "@mui/material": "5̂.15.15"
- "@mui/system": "5̂.15.15"
- "@mui/x-data-grid": "7̂.1.0"
- "@mui/x-date-pickers": "6̂.19.6"

- "country-list": "$\hat{2}$.3.0"
- "currency-codes": "$\hat{2}$.1.0"
- "date-fns": "$\hat{3}$.6.0"
- "express": "$\hat{4}$.18.3"
- "leaflet": "$\hat{1}$.9.4"
- "node-fetch": "$\hat{3}$.3.2"
- "react-leaflet": "$\hat{4}$.2.1"
- "react-leaflet-cluster": "$\hat{2}$.1.0"
- "react-material-ui-carousel": "$\hat{3}$.4.2"
- "react-select": "$\hat{5}$.8.0"
- "react-virtualized": "$\hat{9}$.22.5"
- "react-window": "$\hat{1}$.8.10"
- "recharts": "$\hat{2}$.12.3"
- "cors": "$\hat{2}$.8.5"
- "nodemailer": "$\hat{6}$.9.8"
- "pdfkit": "$\hat{0}$.14.0"
- "qrcode": "$\hat{1}$.5.3"
- "@date-io/dayjs": "$\hat{3}$.0.0"
- "@mui/icons-material": "$\hat{5}$.14.19"
- "@mui/x-date-pickers": "$\hat{6}$.19.9"
- "@testing-library/jest-dom": "$\hat{5}$.17.0"
- "@testing-library/react": "$\hat{1}$3.4.0"
- "@testing-library/user-event": "$\hat{1}$3.5.0"

- "dayjs": "$\hat{1}$.11.10"
- "emoji-flag": "$\hat{1}$.1.0"
- "firebase": "$\hat{1}$0.4.0"
- "i18n-iso-countries": "$\hat{7}$.7.0"
- "material-ui-flags": "$\hat{1}$.2.4"
- "qrcode.react": "$\hat{3}$.1.0"
- "react": "$\hat{1}$8.2.0"
- "react-country-flag": "$\hat{3}$.1.0"
- "react-datepicker": "$\hat{4}$.25.0"
- "react-dom": "$\hat{1}$8.2.0"
- "react-firebase-hooks": "$\hat{5}$.1.1"
- "react-router-dom": "$\hat{6}$.16.0"
- "react-scripts": "5.0.1"
- "react-slick": "$\hat{0}$.30.2"
- "slick-carousel": "$\hat{1}$.8.1"
- "tss-react": "$\hat{4}$.9.7"
- "uuid": "$\hat{9}$.0.1"
- "web-vitals": "$\hat{2}$.1.4"
- "firebase-admin": "$\hat{1}$2.1.0"
- "firebase-functions": "$\hat{5}$.0.1"

Chapter 3

User Documentation

3.1 Login Screen

Login page is the first page that welcomes users after the successful launching of the application. If a user has valid account, he/she can log in using email and password that was used in registration process. The application is consisted of 3 roles: Tourist, City and System Administrator, and the login process navigates to different dashboards based on the role.

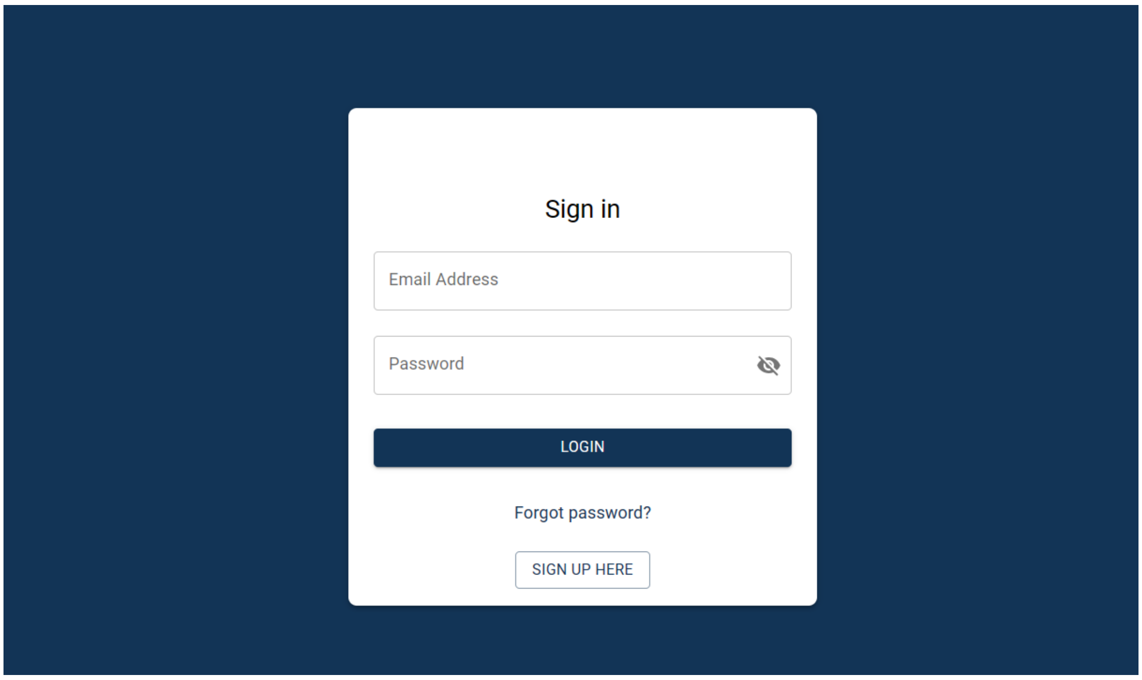
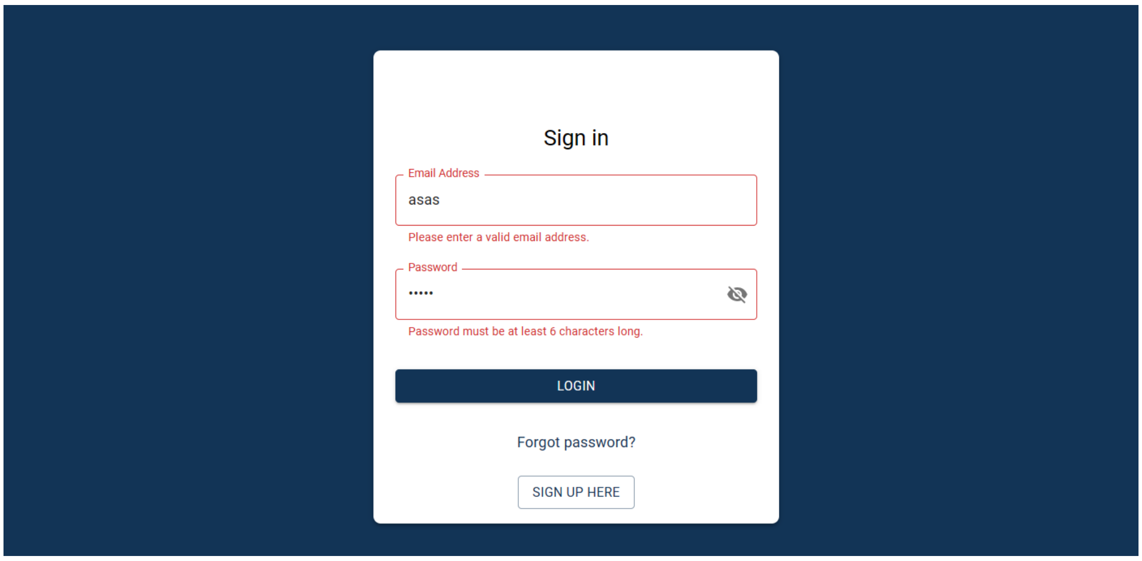


Figure 3.1: Login screen

If the entered email or password is wrong or does meet the format requirements, users will get relevant error messages.

1. **Login attempt with wrong credentials:**



asas

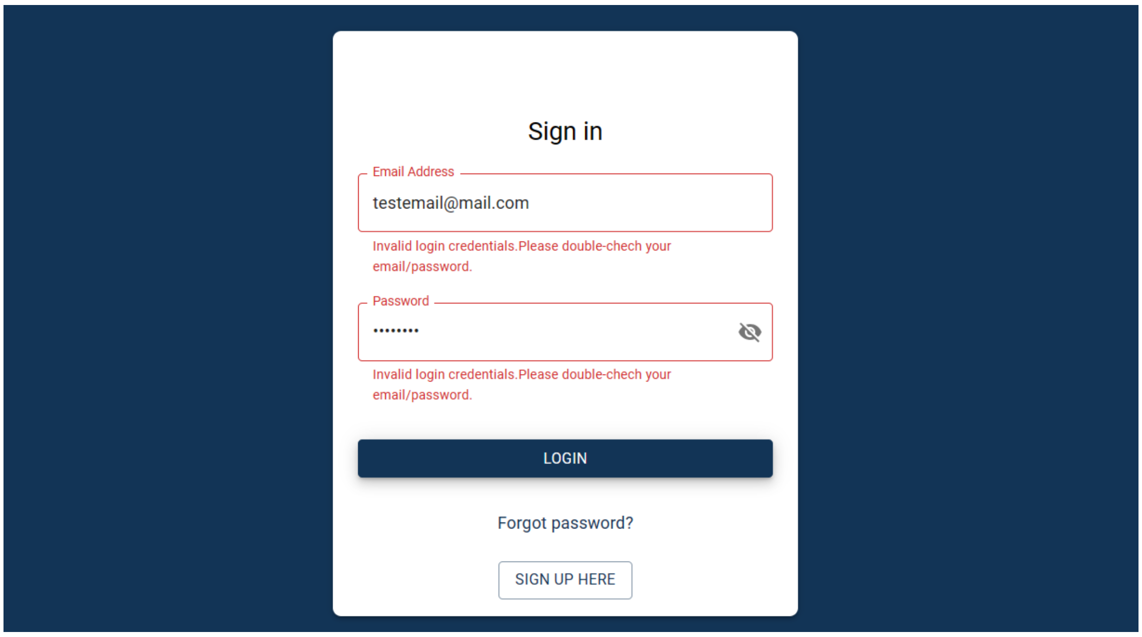
Please enter a valid email address.

•••••

Password must be at least 6 characters long.

Figure 3.2: Login page validations for wrong credentials

2. **Login attempt with incorrect format of email and password:**



testemail@mail.com

Invalid login credentials.Please double-chech your email/password.

••••••••

Invalid login credentials.Please double-chech your email/password.

Figure 3.3: Login page validations for incorrect format

3.2 Forgot Password

Users have availability to reset their passwords. They can access the Forgot Password page to achieve this functionality by clicking on the "Forgot Password?" button on the Login screen.

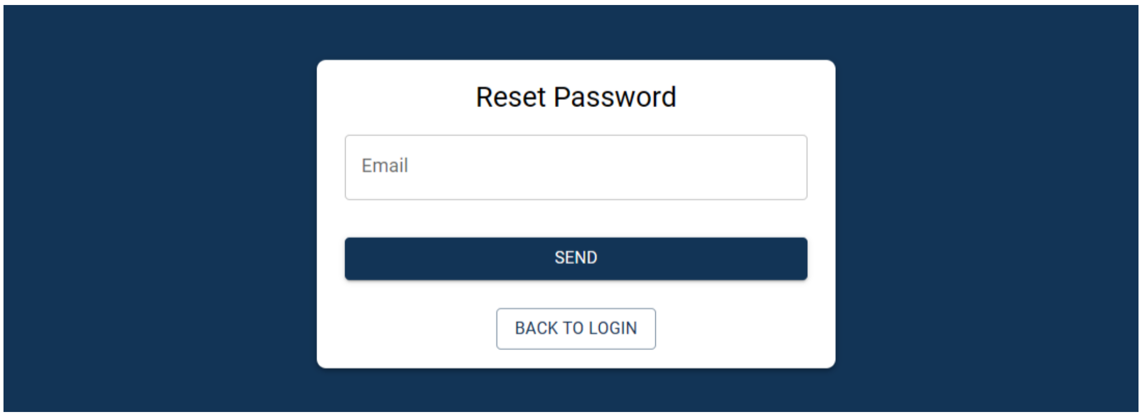
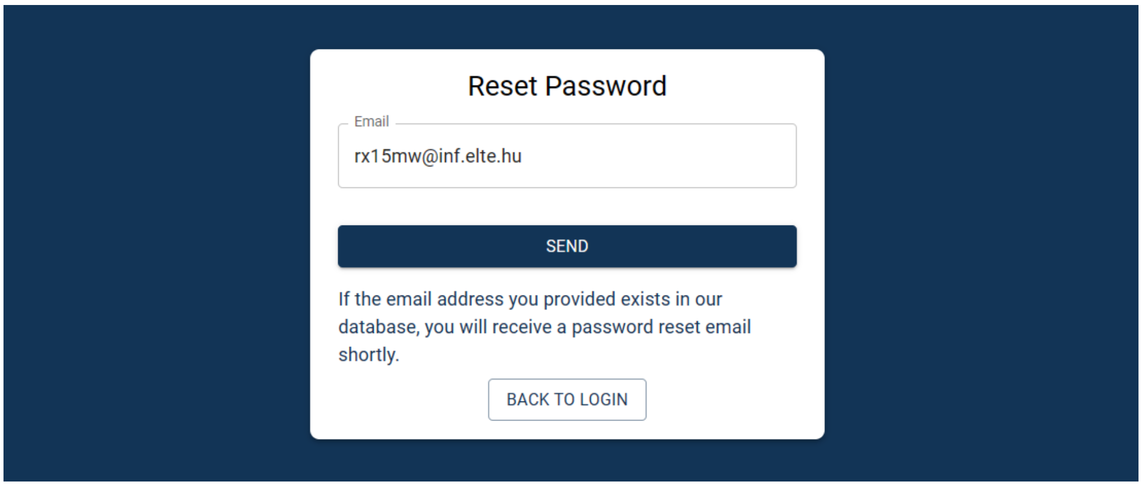


Figure 3.4: Forgot Password page

After entering a valid email address and clicking on the "Send" button, the system will output a notification regarding the procedure.

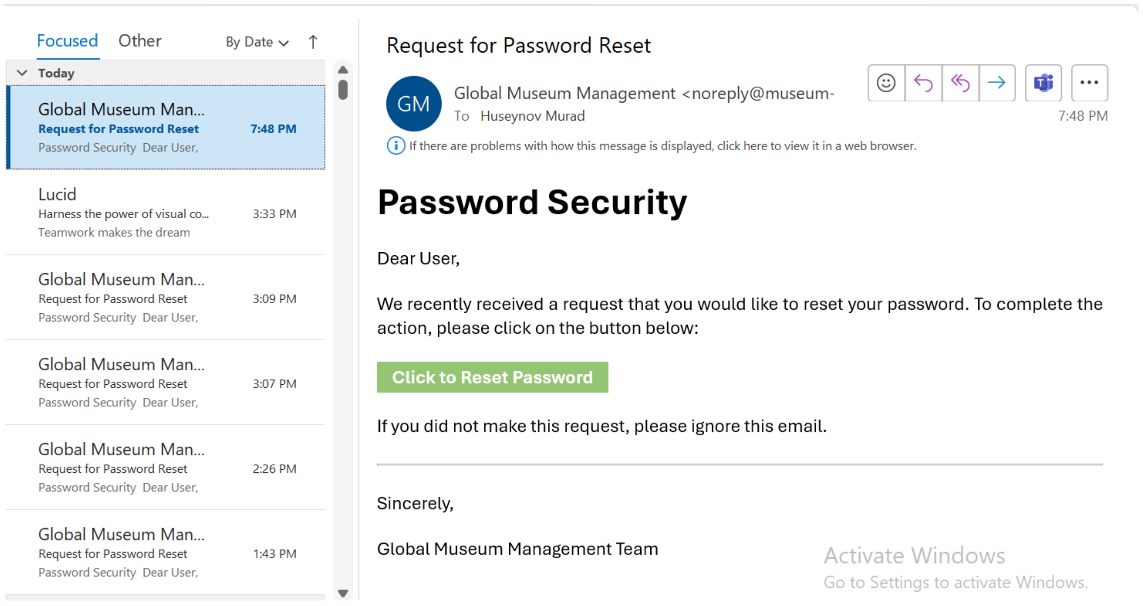


rx15mw@inf.elte.hu

If the email address you provided exists in our database, you will receive a password reset email shortly.

Figure 3.5: Forgot Password page after entering an email address

Shortly, a password reset email will be sent to the provided email address.


Focused Other By Date

Today

Global Museum Man...
Request for Password Reset **7:48 PM**
Password Security Dear User,

Lucid
Harness the power of visual co... 3:33 PM
Teamwork makes the dream

Global Museum Man...
Request for Password Reset 3:09 PM
Password Security Dear User,

Global Museum Man...
Request for Password Reset 3:07 PM
Password Security Dear User,

Global Museum Man...
Request for Password Reset 2:26 PM
Password Security Dear User,

Global Museum Man...
Request for Password Reset 1:43 PM
Password Security Dear User,

Request for Password Reset

GM Global Museum Management <noreply@museum-
To Huseynov Murad
7:48 PM

If there are problems with how this message is displayed, click here to view it in a web browser.

Password Security

Dear User,

We recently received a request that you would like to reset your password. To complete the action, please click on the button below:

Click to Reset Password

If you did not make this request, please ignore this email.

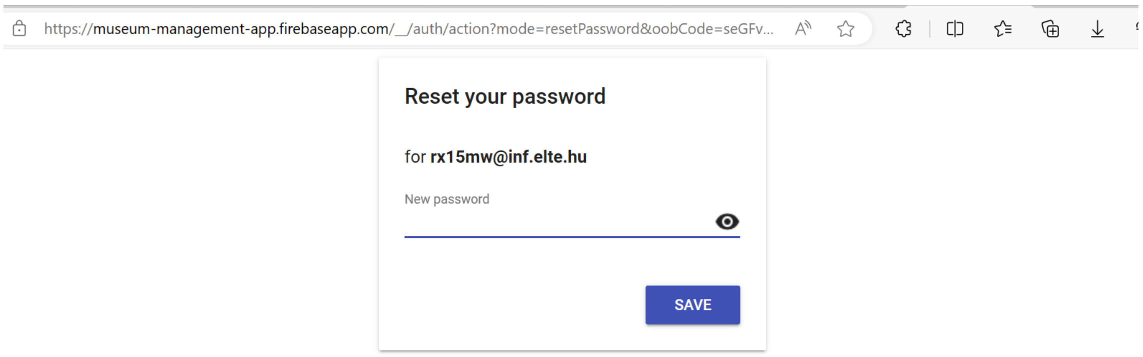
Sincerely,

Global Museum Management Team

Activate Windows
Go to Settings to activate Windows.

Figure 3.6: Password reset email

This email contains a link for choosing a new password. After a successful reset, users can use their new password to log in.


https://museum-management-app.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=seGFv...

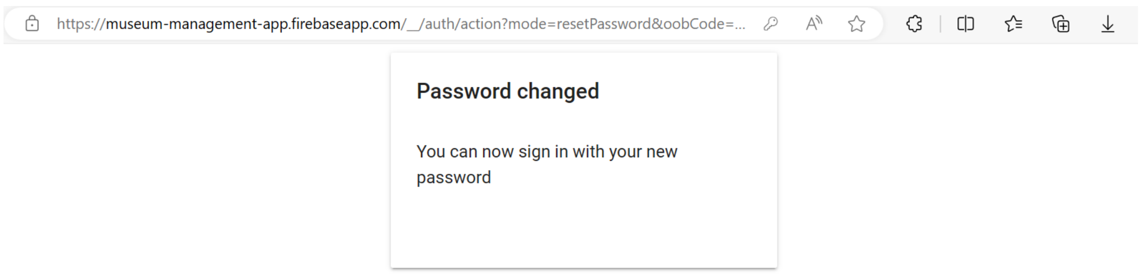
Reset your password

for **rx15mw@inf.elte.hu**

New password

SAVE

Figure 3.7: Content of password reset link


https://museum-management-app.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=...

Password changed

You can now sign in with your new password

Figure 3.8: Password update using link

3.3 Registration

To register a new account users need to click on "SIGN UP HERE" button on the login page. Afterwards, it will be required to choose a role, which can be either city or tourist.

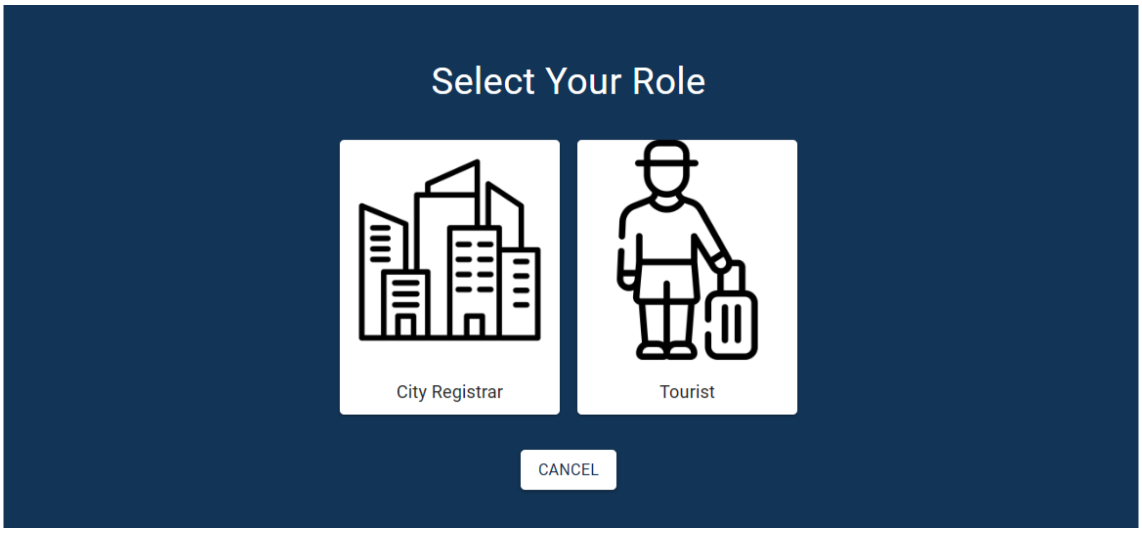


Figure 3.9: Role selection page

If the chosen role is city, a 2-step registration page will open. The first part is about providing registrar-related information.

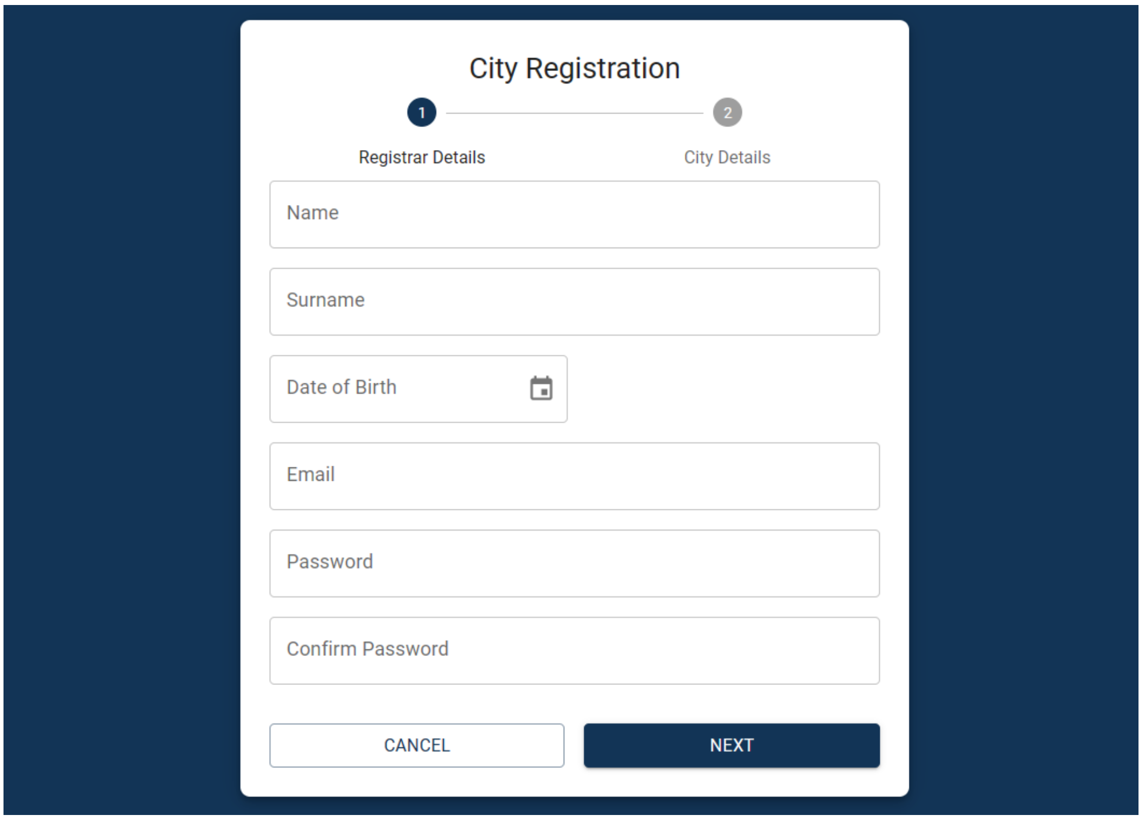


Figure 3.10: First part of city registration

The second part is about adding city-related information. City name field should be typed in manually, while country name can be chosen from a drop-down list. City ID is a special field which is crucial for city registrations, as the signup cannot be completed without properly filling in this field. In order to avoid unauthorized enrolling, a unique city ID is generated by system administrator and delivered to the city registrar. This ID can be used for one registration only.

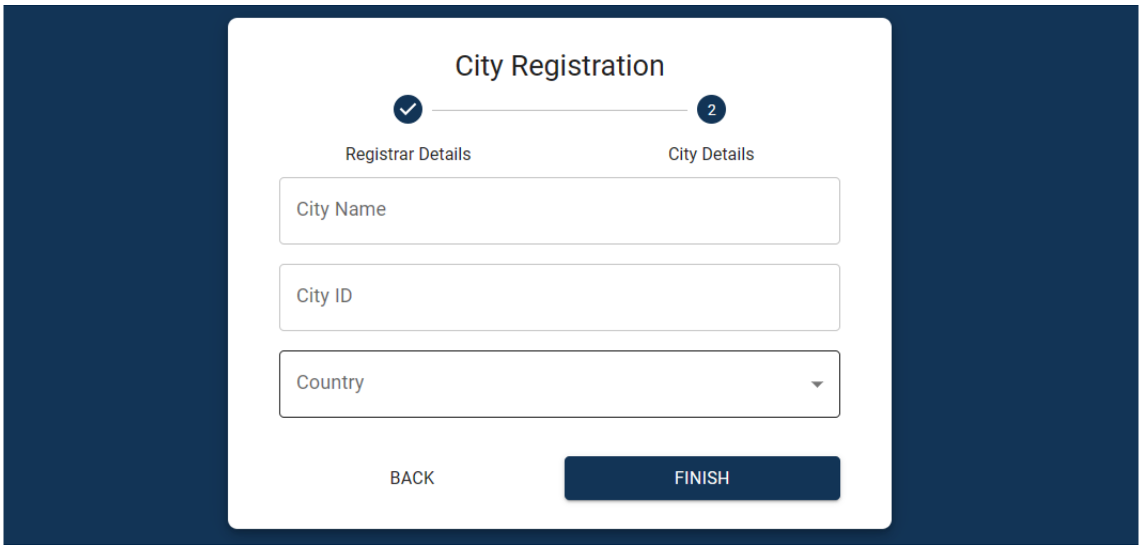


Figure 3.11: Second part of city registration

If the selected role is tourist, a form, requiring basic personal data and credentials, will open. After filling in all the fields properly, a tourist account can register successfully.

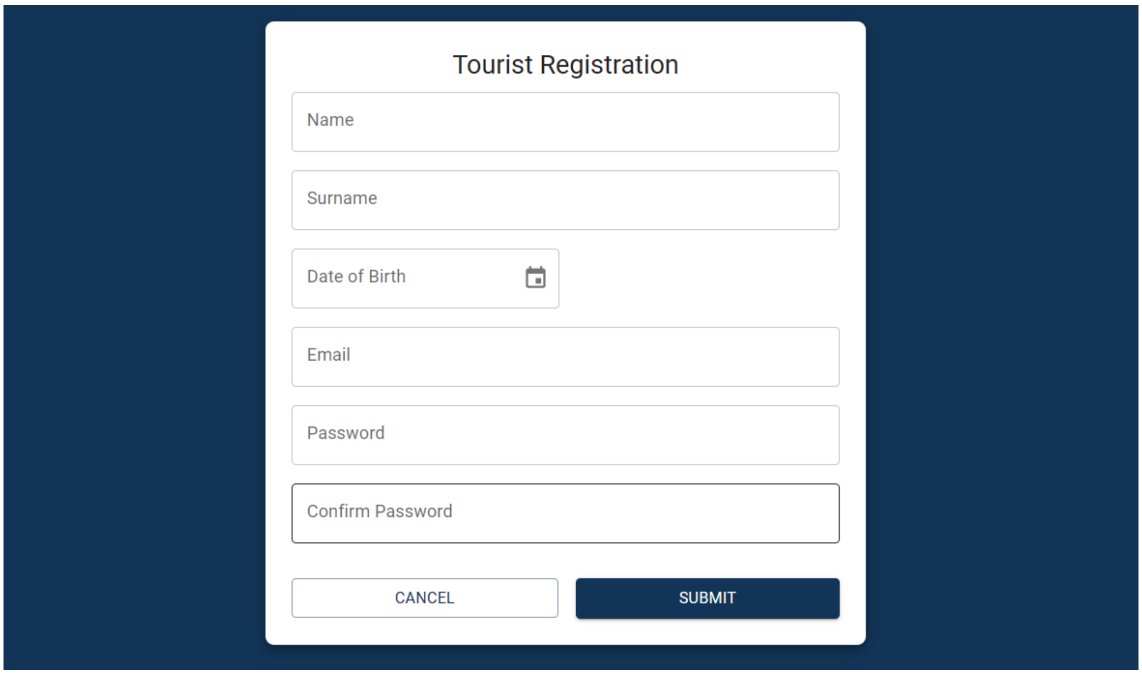
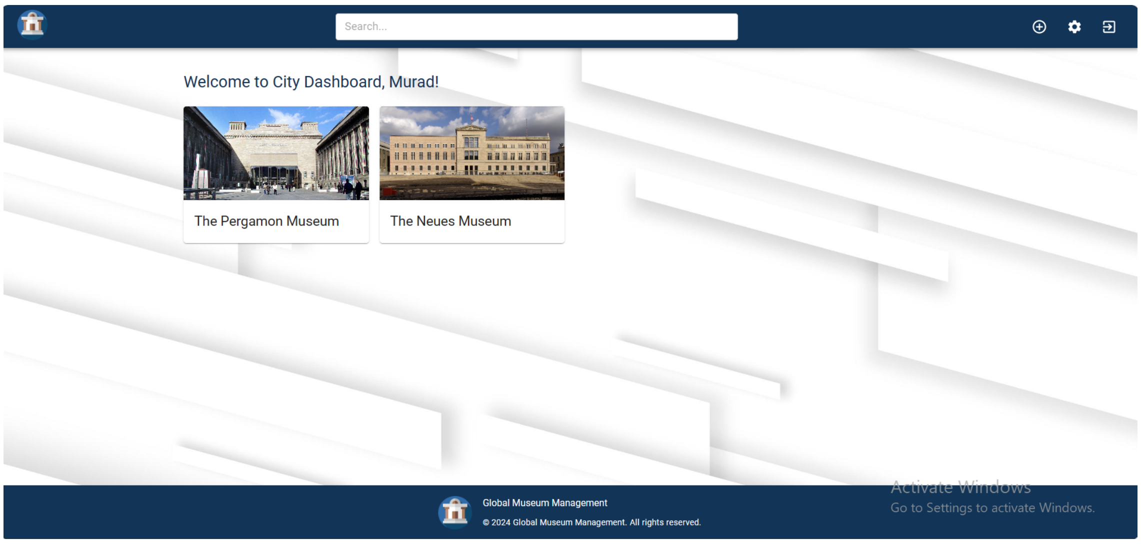


Figure 3.12: Tourist registration form

3.4 City Account

3.4.1 City Dashboard

If a successful login is made with credentials that belong to a city role, the city dashboard will be shown. The main feature of this page is that, it will show all the museums added by the city registrar.



Search...
Welcome to City Dashboard, Murad!
The Pergamon Museum
The Neues Museum
Global Museum Management
© 2024 Global Museum Management. All rights reserved.
Activate Windows
Go to Settings to activate Windows.

Figure 3.13: City dashboard

In each museum card there are relevant icons for editing and deleting museums. If the delete icon is clicked, a dialog will open to require a confirmation from the user about the deletion process.

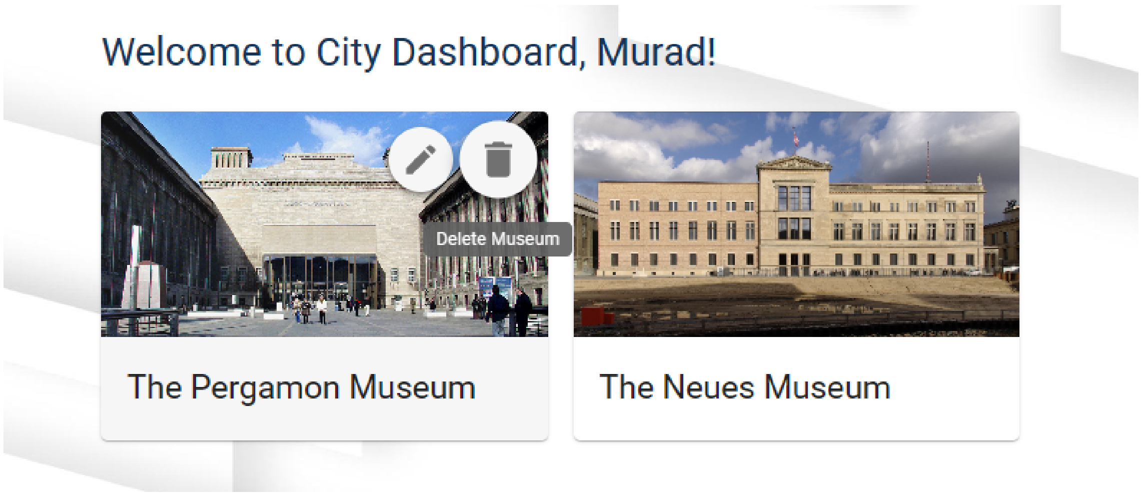
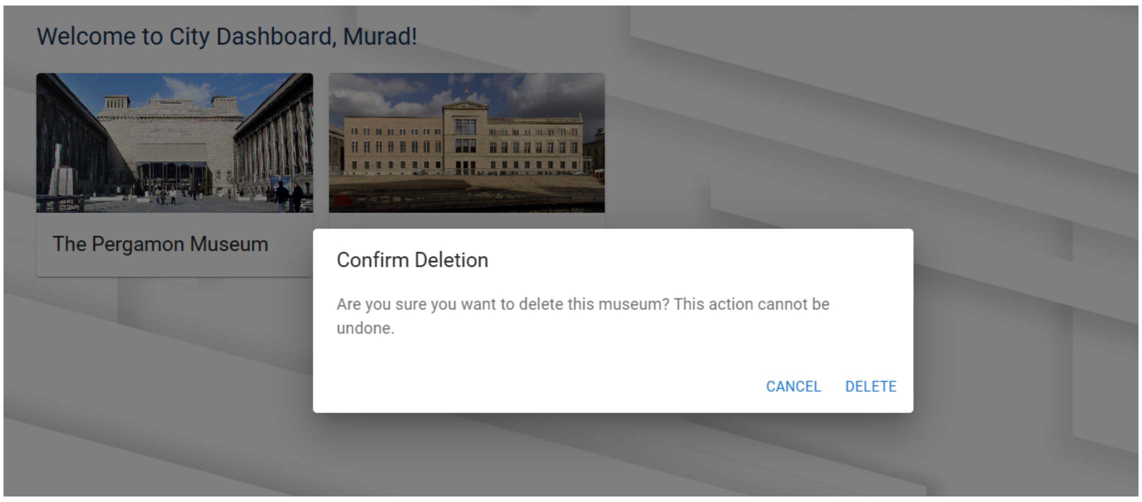


Figure 3.14: Edit and Delete icons on museum cards



Confirm Deletion

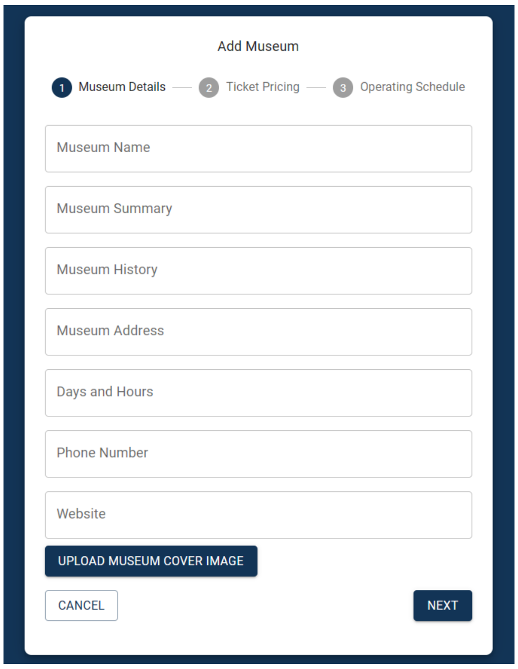
Are you sure you want to delete this museum? This action cannot be undone.

Figure 3.15: Confirmation dialog for museum deletion

The multi-functional navigation bar can be used for searching museums in city dashboard. Also, the application's logo on the left side of the navigation bar is clickable and directs the screen back to city dashboard. Meanwhile, users can click on the icons on right-hand side of the navigation bar to add a new museum, go to settings page, or log out in the specified order.

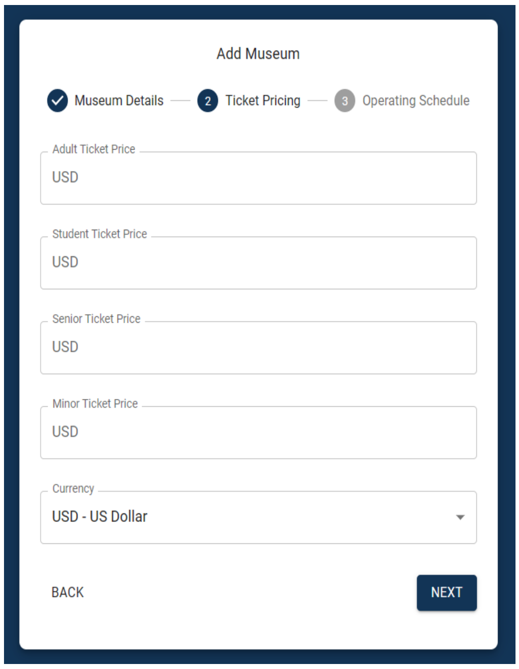
3.4.2 Add Museum and Edit Museum

Adding a museum is consisted of 3 parts. The first part is about adding textual information regarding the museum such as, museum's name, summary, history, address, opening days and hours, phone number and website. Additionally, a cover image is uploaded, however this image is only displayed on the museum card for city registrars. The second card part is dedicated to choosing desired currency and providing prices per available categories. For user convenience, there are 4 possible categories which are adults, students, seniors and minors.


Add Museum

Museum Name
Museum Summary
Museum History
Museum Address
Days and Hours
Phone Number
Website

Figure 3.16: First phase of adding a museum


Add Museum

Adult Ticket Price: USD
Student Ticket Price: USD
Senior Ticket Price: USD
Minor Ticket Price: USD
Currency: USD - US Dollar

Figure 3.17: Second phase of adding a museum

The third and final phase of adding a museum is registering operating days and hours. In the top of the page, users will see a calendar that can be used for adding or removing dates manually. This functionality is necessary for managing special days and holidays when the museum will not operate. For museums that operate on a regular basis, the application provides two options, which are "Daily" and "Weekly". In both options, it is required to enter start and end dates, however in the latter choice, users will also see days of the week listed and only the selected days of the week will be chosen on the calendar.

After selecting days, users will also be required to enter opening and closing times, and select a slot limit. By default, the application will create 30-minute slots between the opening and closing times entered, and the slot limit refers to the maximum number of places per slot. After adding all the details, and clicking on the "Submit" button, the museum will be registered and visible both on city and tourist sides.

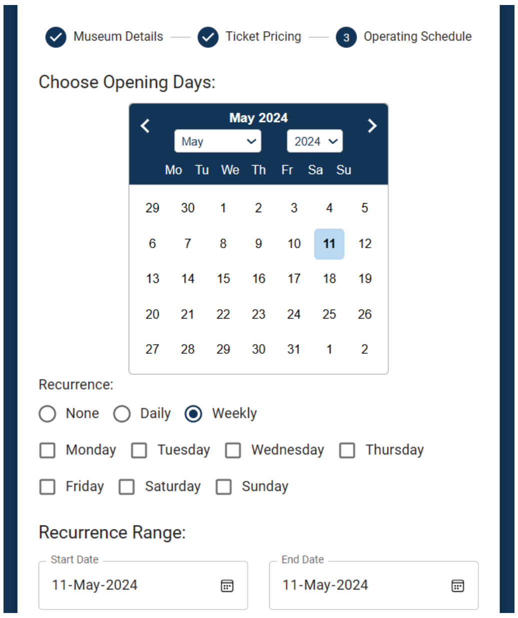


Figure 3.18: (Recurrent) calendar on the third phase of adding museum

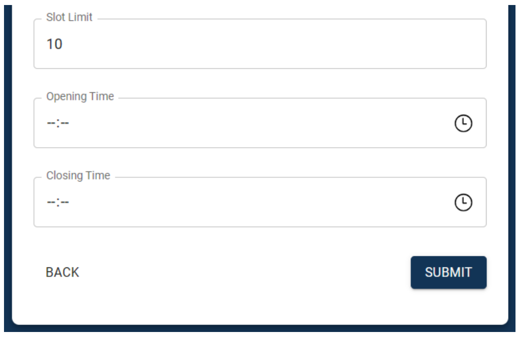


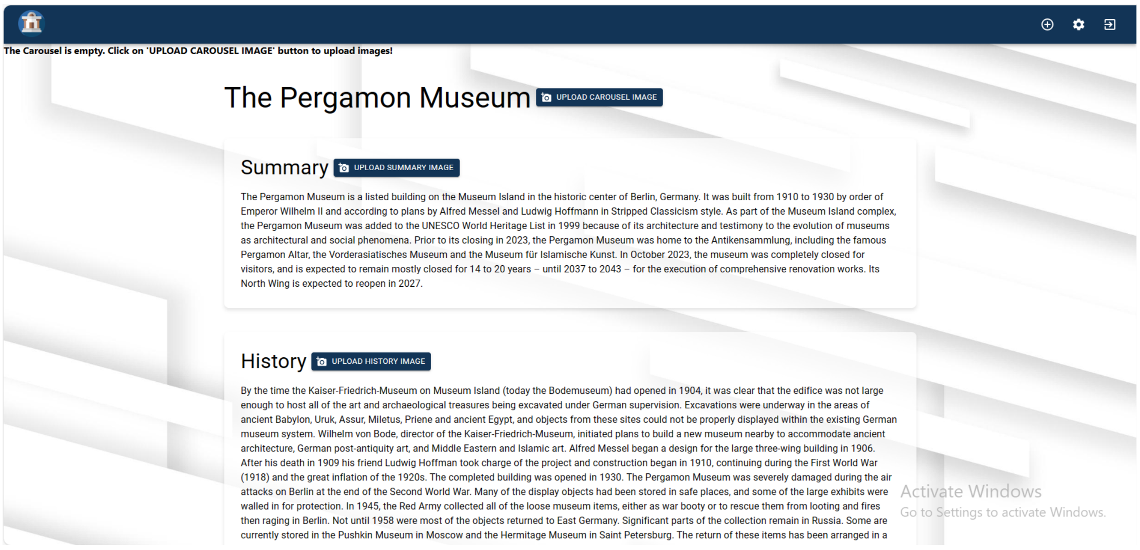
Figure 3.19: Opening/Closing times and slot generation on the third phase of adding museum

As mentioned previously, in order to edit a museums, users need to click on edit icon which on each every museum card. The navigated page will have the same interface as in adding museum page with all the fields being filled in with current data.

3.4.3 Customizable Museum Details page for city role

If users click on a museum card, a page dedicated to that particular museum will be shown. The textual information that was provided during museum adding process will be displayed in this page. Additionally, the address of museum will be pinpointed on a map. Specific to the city accounts, there will be 3 buttons, called "Upload Carousel Image", "Upload Summary Image" and "Upload History Image". The purpose of these buttons is to add section-specific images, and provide additional visuals in a carousel by not overwhelming "Summary" and "History" sections.

Users can delete already added images by click on "delete" icon which is present on each visual element. Also, if users want to view the images in "Summary" and "History" sections individually, they can click on the image and it will open in a separate browser tab.


The Carousel is empty. Click on 'UPLOAD CAROUSEL IMAGE' button to upload images!

The Pergamon Museum

Summary

The Pergamon Museum is a listed building on the Museum Island in the historic center of Berlin, Germany. It was built from 1910 to 1930 by order of Emperor Wilhelm II and according to plans by Alfred Messel and Ludwig Hoffmann in Stripped Classicism style. As part of the Museum Island complex, the Pergamon Museum was added to the UNESCO World Heritage List in 1999 because of its architecture and testimony to the evolution of museums as architectural and social phenomena. Prior to its closing in 2023, the Pergamon Museum was home to the Antikensammlung, including the famous Pergamon Altar, the Vorderasiatisches Museum and the Museum für Islamische Kunst. In October 2023, the museum was completely closed for visitors, and is expected to remain mostly closed for 14 to 20 years – until 2037 to 2043 – for the execution of comprehensive renovation works. Its North Wing is expected to reopen in 2027.

History

By the time the Kaiser-Friedrich-Museum on Museum Island (today the Bodemuseum) had opened in 1904, it was clear that the edifice was not large enough to host all of the art and archaeological treasures being excavated under German supervision. Excavations were underway in the areas of ancient Babylon, Uruk, Assur, Miletus, Priene and ancient Egypt, and objects from these sites could not be properly displayed within the existing German museum system. Wilhelm von Bode, director of the Kaiser-Friedrich-Museum, initiated plans to build a new museum nearby to accommodate ancient architecture, German post-antiquity art, and Middle Eastern and Islamic art. Alfred Messel began a design for the large three-wing building in 1906. After his death in 1909 his friend Ludwig Hoffman took charge of the project and construction began in 1910, continuing during the First World War (1918) and the great inflation of the 1920s. The completed building was opened in 1930. The Pergamon Museum was severely damaged during the air attacks on Berlin at the end of the Second World War. Many of the display objects had been stored in safe places, and some of the large exhibits were walled in for protection. In 1945, the Red Army collected all of the loose museum items, either as war booty or to rescue them from looting and fires then raging in Berlin. Not until 1958 were most of the objects returned to East Germany. Significant parts of the collection remain in Russia. Some are currently stored in the Pushkin Museum in Moscow and the Hermitage Museum in Saint Petersburg. The return of these items has been arranged in a

Figure 3.20: City specific museum details page containing image upload buttons

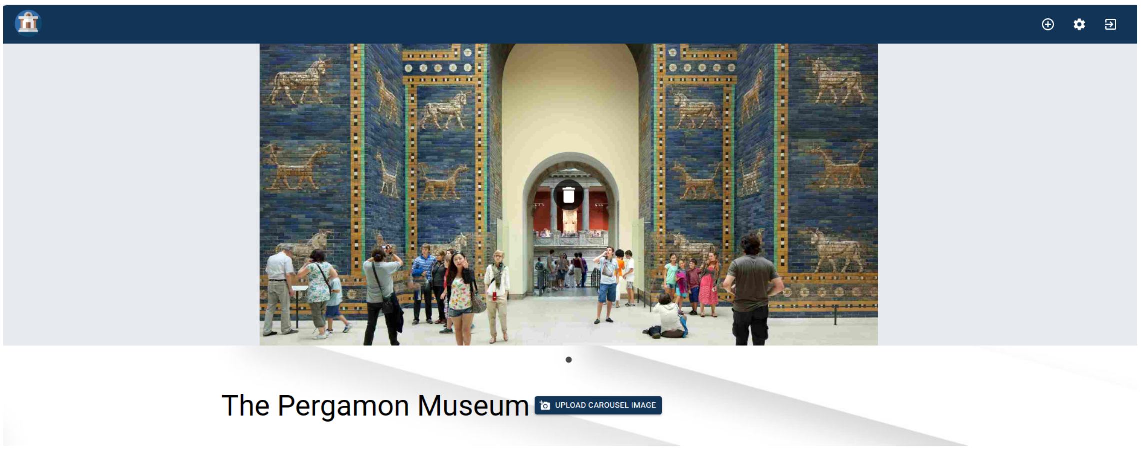
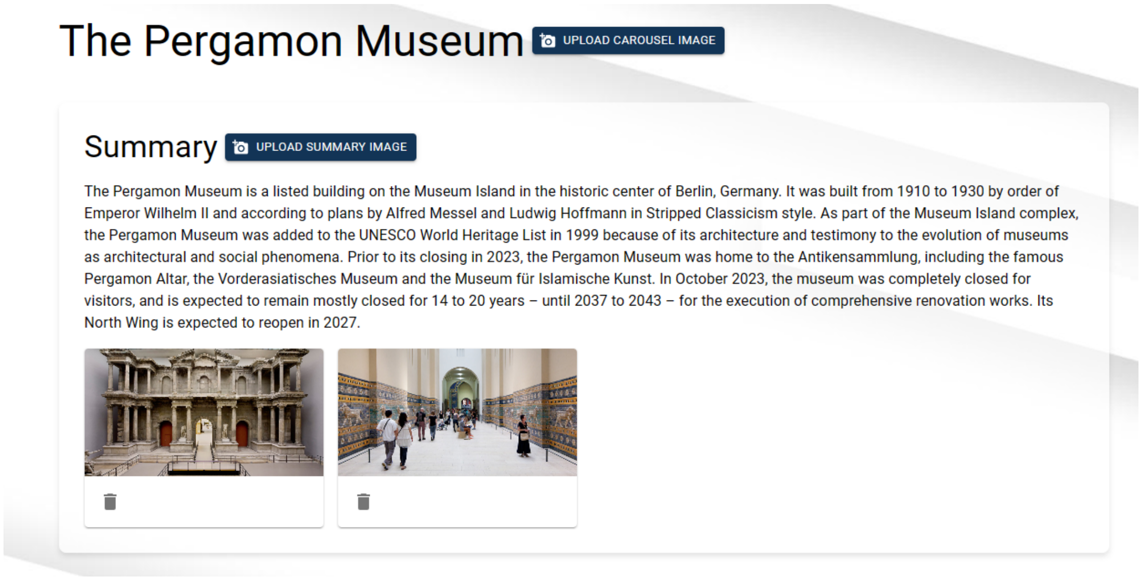


Figure 3.21: Carousel image added by city registrar



The Pergamon Museum

Summary

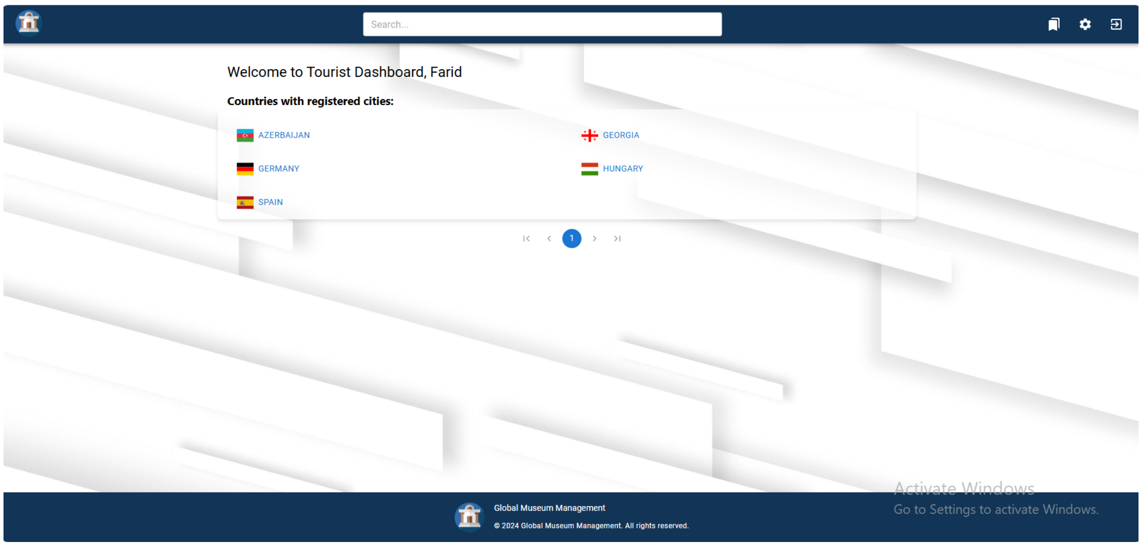
The Pergamon Museum is a listed building on the Museum Island in the historic center of Berlin, Germany. It was built from 1910 to 1930 by order of Emperor Wilhelm II and according to plans by Alfred Messel and Ludwig Hoffmann in Stripped Classicism style. As part of the Museum Island complex, the Pergamon Museum was added to the UNESCO World Heritage List in 1999 because of its architecture and testimony to the evolution of museums as architectural and social phenomena. Prior to its closing in 2023, the Pergamon Museum was home to the Antikensammlung, including the famous Pergamon Altar, the Vorderasiatisches Museum and the Museum für Islamische Kunst. In October 2023, the museum was completely closed for visitors, and is expected to remain mostly closed for 14 to 20 years – until 2037 to 2043 – for the execution of comprehensive renovation works. Its North Wing is expected to reopen in 2027.

Figure 3.22: Image added in History section by city registrar

3.5 Tourist Account

3.5.1 Tourist Dashboard

If a successful login is made with credentials that belong to a tourist role, the tourist dashboard will be shown. Initially, a list of countries will appear where at least one city has been registered. Once a country is clicked, the list of cities will appear.



Welcome to Tourist Dashboard, Farid

Countries with registered cities:

AZERBAIJAN	GEORGIA
GERMANY	HUNGARY
SPAIN	

Global Museum Management

© 2024 Global Museum Management. All rights reserved.

Figure 3.23: Tourist dashboard with the list of countries

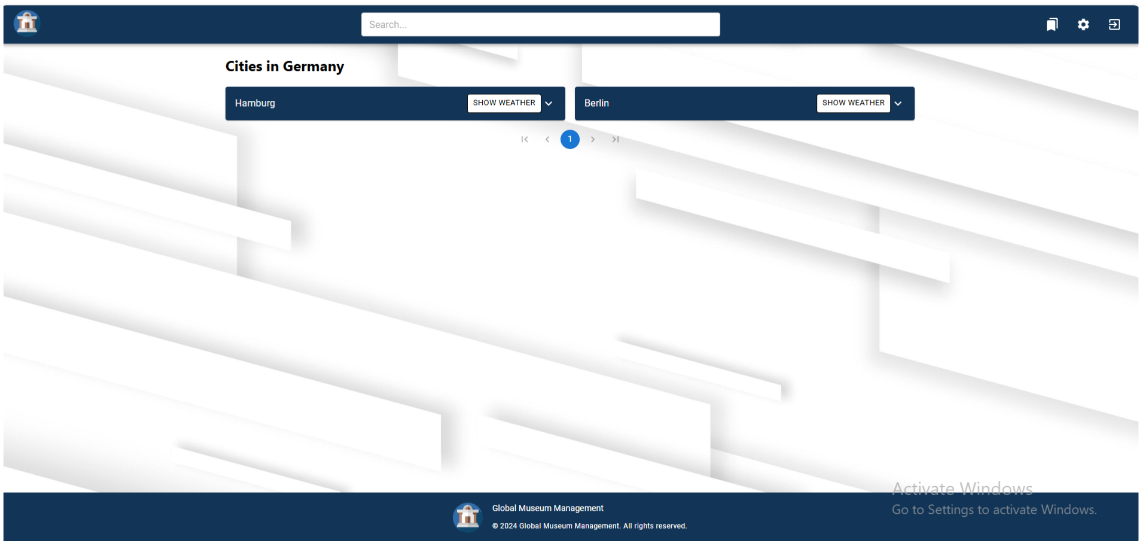


Figure 3.24: List of cities after a country name is clicked

Users can click on a city name to see the list of museum. Additionally, it is possible to view weather report for each city by clicking on "Show Weather" button which is located on the right-side of every city name. The report shows graphical and textual statistics of each city for the upcoming 5 days with 3-hour intervals.

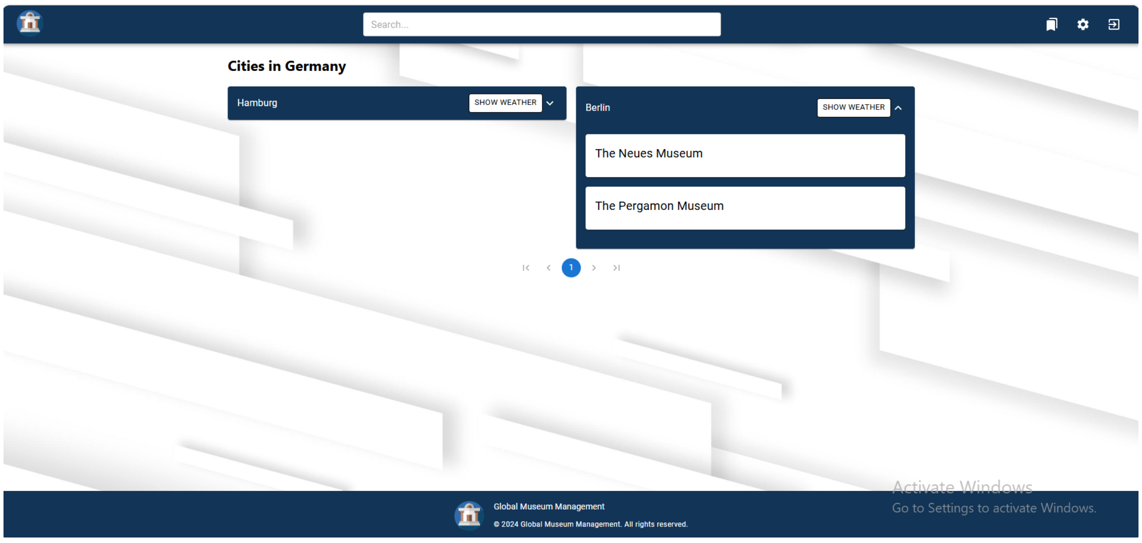
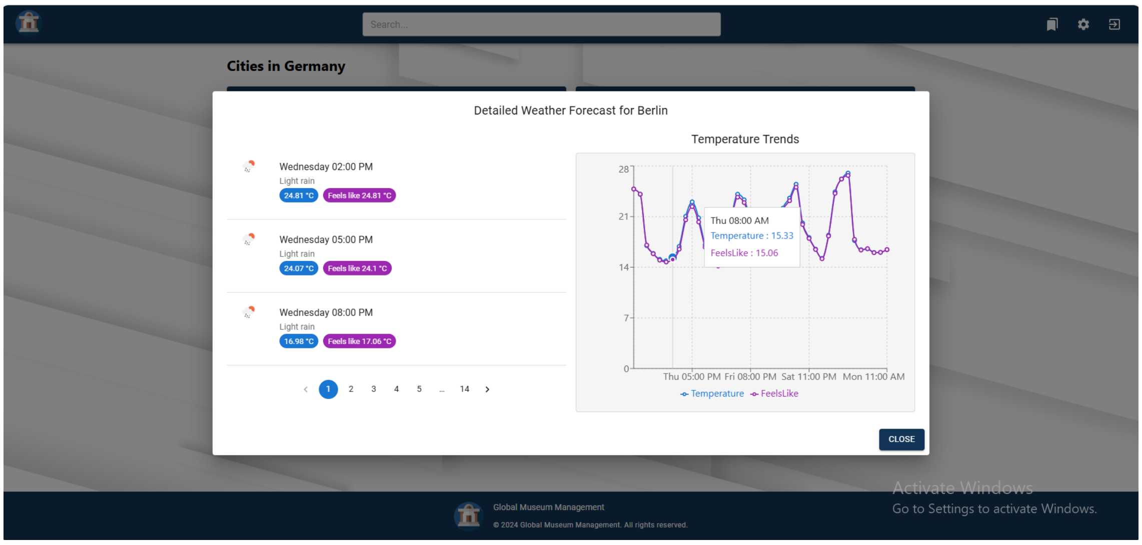


Figure 3.25: List of museums when a city name is clicked



Detailed Weather Forecast for Berlin

Wednesday 02:00 PM
Light rain
24.81 °C Feels like 24.81 °C

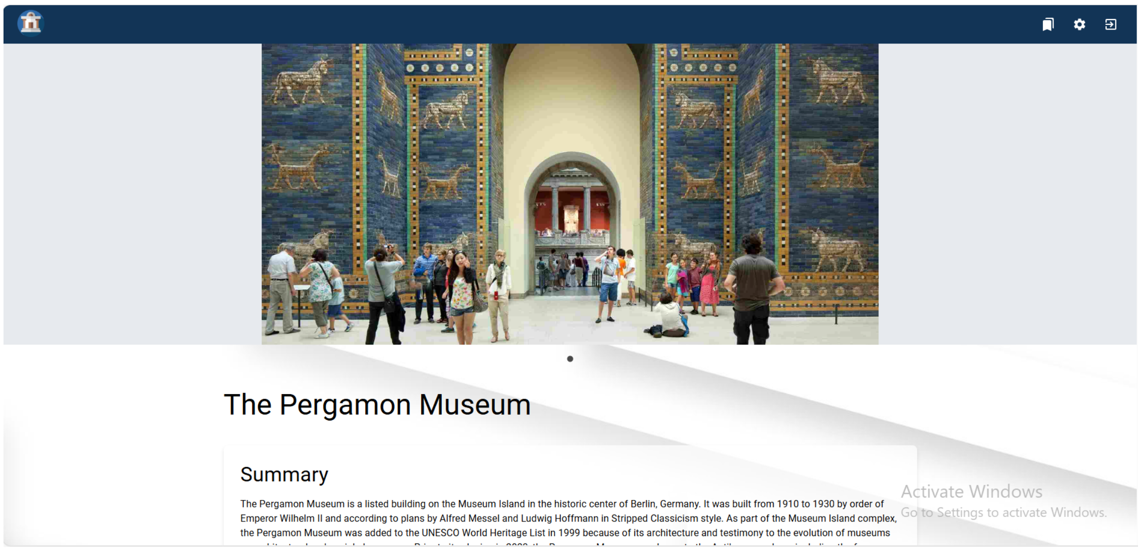
Wednesday 05:00 PM
Light rain
24.07 °C Feels like 24.1 °C

Wednesday 08:00 PM
Light rain
16.98 °C Feels like 17.06 °C

Figure 3.26: Weather report

3.5.2 Museum Details page for tourist role

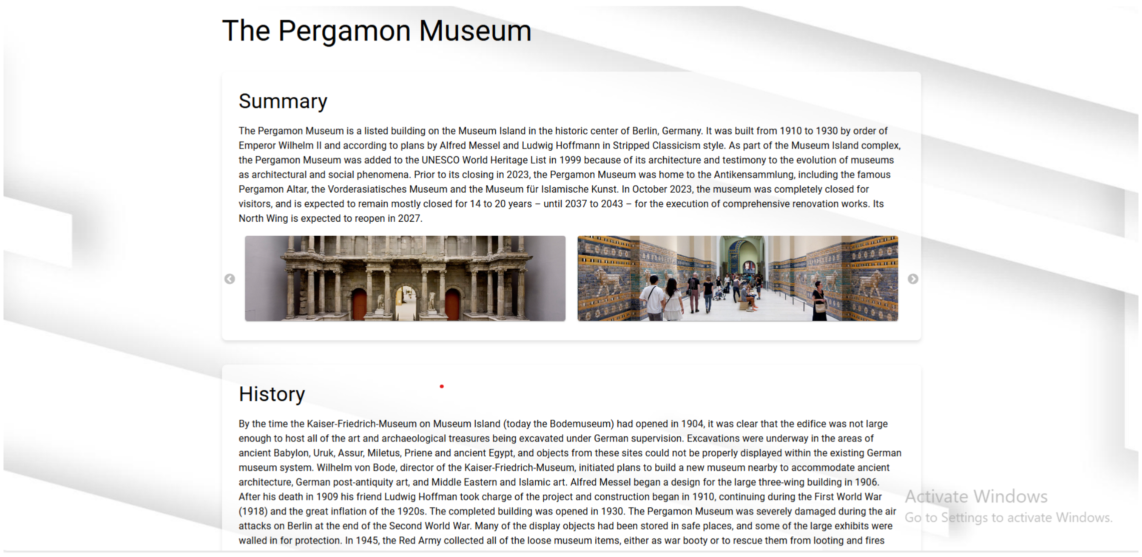
If users click on a museum name in the tourist dashboard, they will be redirected to a page where the selected museum's details are shown. The interface and main functionalities on this page are nearly the same as those in the city side with extra tourist-specific options. Moreover, tourists cannot add or delete visuals in this page, but see those visuals that have added by city registrar.


The Pergamon Museum

Summary

The Pergamon Museum is a listed building on the Museum Island in the historic center of Berlin, Germany. It was built from 1910 to 1930 by order of Emperor Wilhelm II and according to plans by Alfred Messel and Ludwig Hoffmann in Stripped Classicism style. As part of the Museum Island complex, the Pergamon Museum was added to the UNESCO World Heritage List in 1999 because of its architecture and testimony to the evolution of museums

Figure 3.27: Museum carousel visible by tourists


The Pergamon Museum

Summary

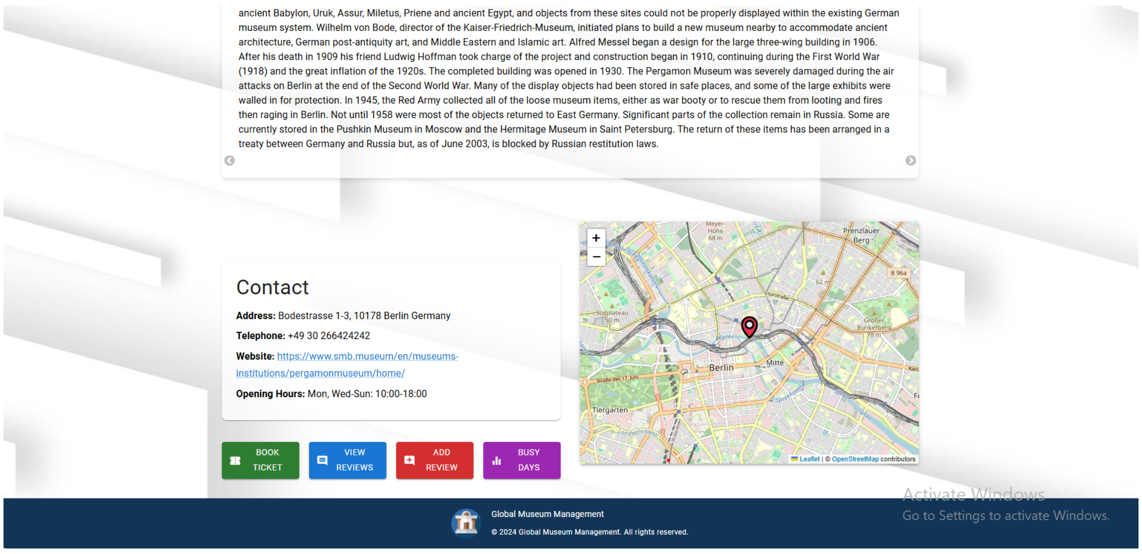
The Pergamon Museum is a listed building on the Museum Island in the historic center of Berlin, Germany. It was built from 1910 to 1930 by order of Emperor Wilhelm II and according to plans by Alfred Messel and Ludwig Hoffmann in Stripped Classicism style. As part of the Museum Island complex, the Pergamon Museum was added to the UNESCO World Heritage List in 1999 because of its architecture and testimony to the evolution of museums as architectural and social phenomena. Prior to its closing in 2023, the Pergamon Museum was home to the Antikensammlung, including the famous Pergamon Altar, the Vorderasiatisches Museum and the Museum für Islamische Kunst. In October 2023, the museum was completely closed for visitors, and is expected to remain mostly closed for 14 to 20 years – until 2037 to 2043 – for the execution of comprehensive renovation works. Its North Wing is expected to reopen in 2027.

History

By the time the Kaiser-Friedrich-Museum on Museum Island (today the Bodemuseum) had opened in 1904, it was clear that the edifice was not large enough to host all of the art and archaeological treasures being excavated under German supervision. Excavations were underway in the areas of ancient Babylon, Uruk, Assur, Miletus, Priene and ancient Egypt, and objects from these sites could not be properly displayed within the existing German museum system. Wilhelm von Bode, director of the Kaiser-Friedrich-Museum, initiated plans to build a new museum nearby to accommodate ancient architecture, German post-antiquity art, and Middle Eastern and Islamic art. Alfred Messel began a design for the large three-wing building in 1906. After his death in 1909 his friend Ludwig Hoffman took charge of the project and construction began in 1910, continuing during the First World War (1918) and the great inflation of the 1920s. The completed building was opened in 1930. The Pergamon Museum was severely damaged during the air attacks on Berlin at the end of the Second World War. Many of the display objects had been stored in safe places, and some of the large exhibits were walled in for protection. In 1945, the Red Army collected all of the loose museum items, either as war booty or to rescue them from looting and fires

Figure 3.28: Museum sections visible by tourists

In the end of page, where museum contact details and pinpointed location are visible, tourists will see 4 buttons which are custom to tourist accounts.


ancient Babylon, Uruk, Assur, Miletus, Priene and ancient Egypt, and objects from these sites could not be properly displayed within the existing German museum system. Wilhelm von Bode, director of the Kaiser-Friedrich-Museum, initiated plans to build a new museum nearby to accommodate ancient architecture, German post-antiquity art, and Middle Eastern and Islamic art. Alfred Messel began a design for the large three-wing building in 1906. After his death in 1909 his friend Ludwig Hoffman took charge of the project and construction began in 1910, continuing during the First World War (1918) and the great inflation of the 1920s. The completed building was opened in 1930. The Pergamon Museum was severely damaged during the air attacks on Berlin at the end of the Second World War. Many of the display objects had been stored in safe places, and some of the large exhibits were walled in for protection. In 1945, the Red Army collected all of the loose museum items, either as war booty or to rescue them from looting and fires then raging in Berlin. Not until 1958 were most of the objects returned to East Germany. Significant parts of the collection remain in Russia. Some are currently stored in the Pushkin Museum in Moscow and the Hermitage Museum in Saint Petersburg. The return of these items has been arranged in a treaty between Germany and Russia but, as of June 2003, is blocked by Russian restitution laws.

Contact

Address: Bodestrasse 1-3, 10178 Berlin Germany

Telephone: +49 30 266424242

Website: https://www.smb.museum/en/museums-institutions/pergamonmuseum/home/

Opening Hours: Mon, Wed-Sun: 10:00-18:00

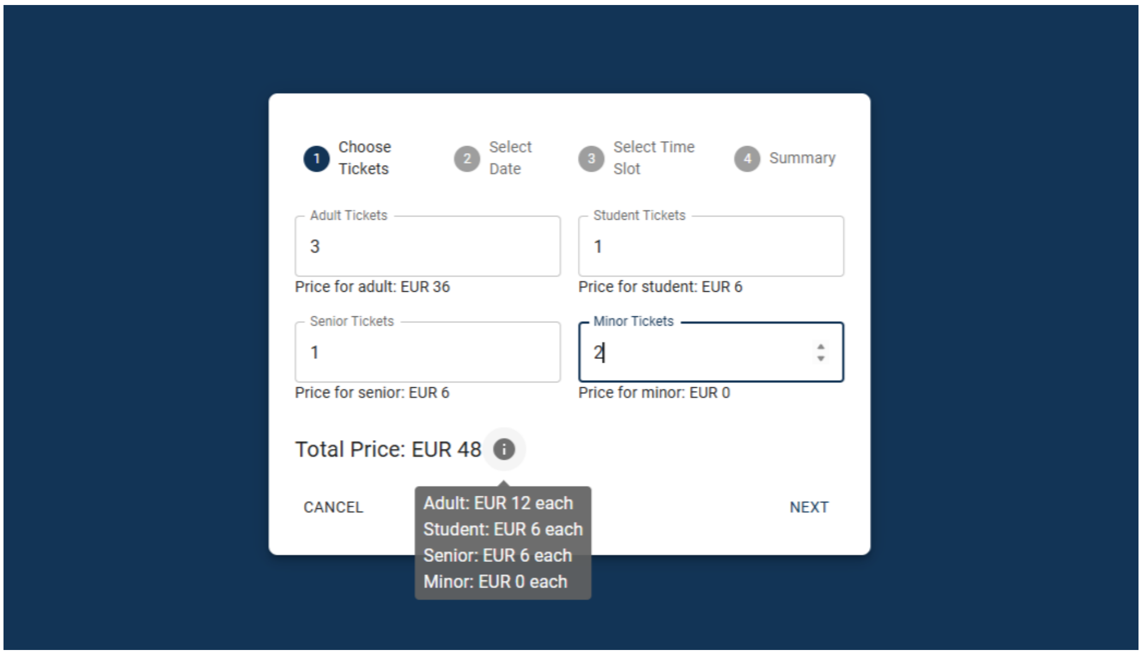
Global Museum Management

© 2024 Global Museum Management. All rights reserved.

Figure 3.29: Tourist-specific buttons

Book ticket

The "Book ticket" button can be used for booking a ticket for the selected museum. This is 4-step process where tourists will start by selecting the desired number of tickets per category. For user convenience, base prices per category will be shown, and users will see the total price when choosing more tickets.



Adult Tickets: 3
Price for adult: EUR 36

Student Tickets: 1
Price for student: EUR 6

Senior Tickets: 1
Price for senior: EUR 6

Minor Tickets: 2
Price for minor: EUR 0

Total Price: EUR 48

Adult: EUR 12 each
Student: EUR 6 each
Senior: EUR 6 each
Minor: EUR 0 each

Figure 3.30: Choosing ticket for tourist booking

Subsequently, users will see a calendar where they should choose one of the available days, and proceed to the third step.

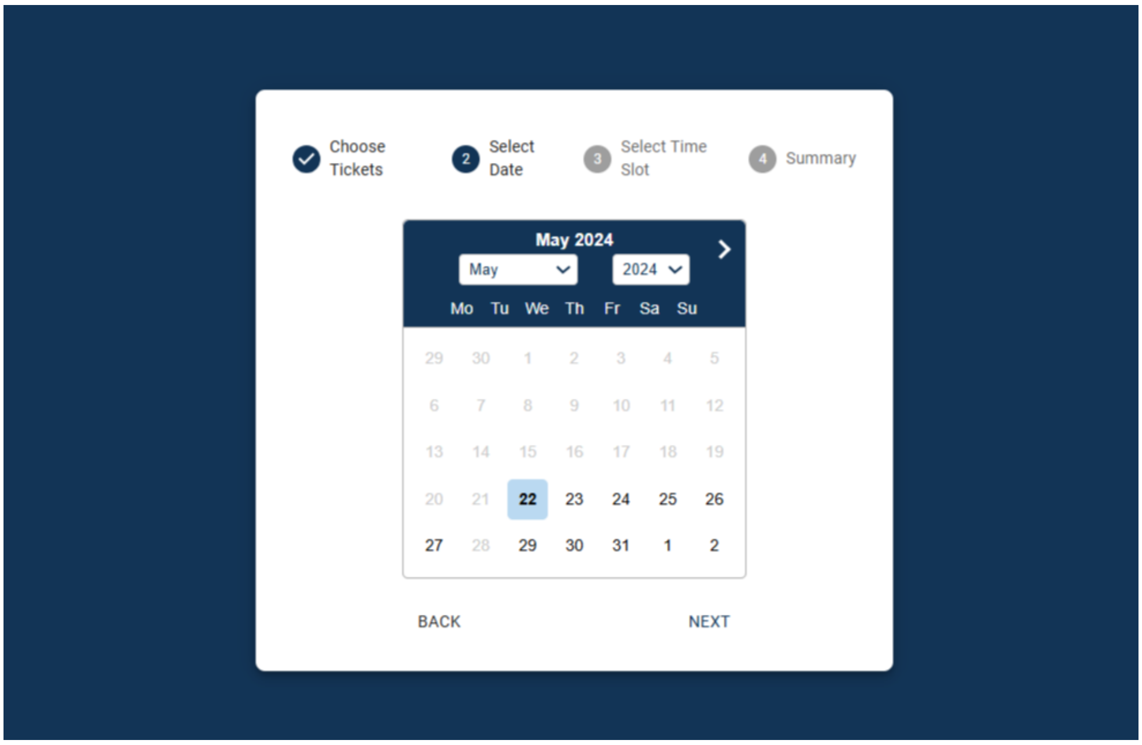


Figure 3.31: Selecting a day for tourist booking

After selecting a date, it is required to decide on a time slot. Users can see all the possible slots listed together with the number available places for each slot. In this step, there is also a checkbox that allows users to receive a PDF ticket, summarizing the booking.

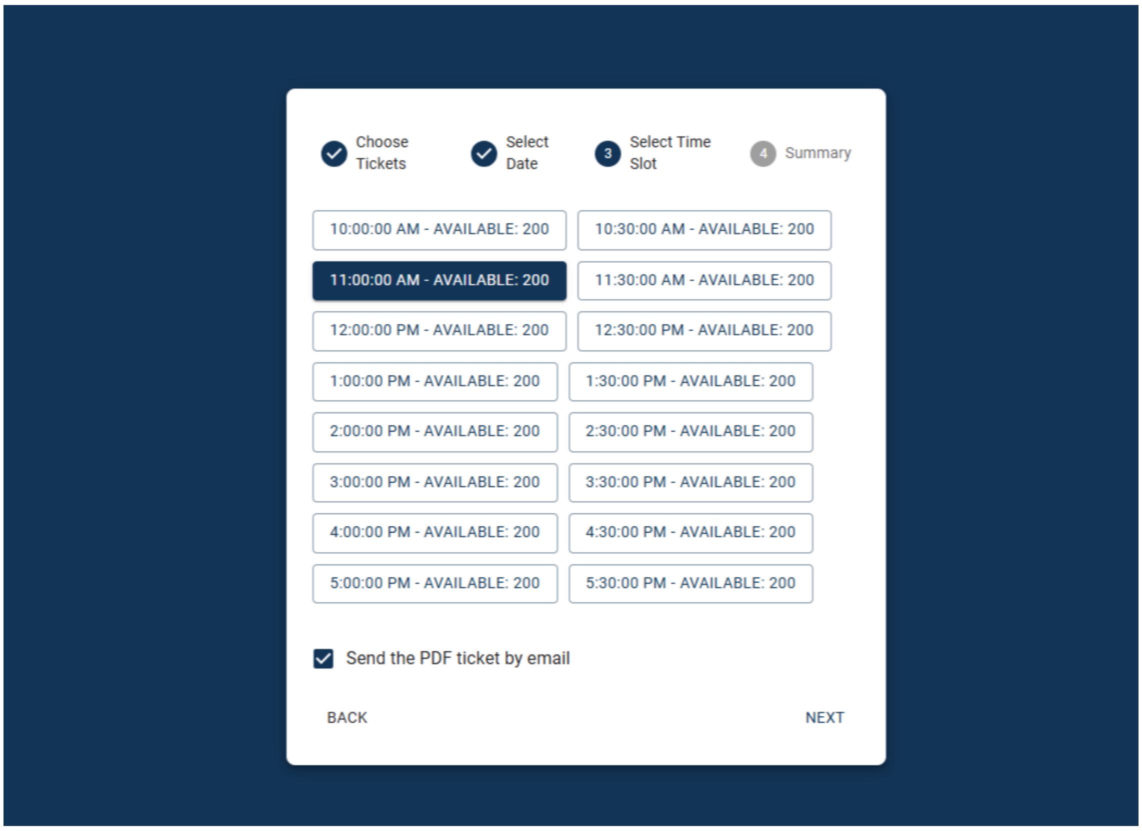
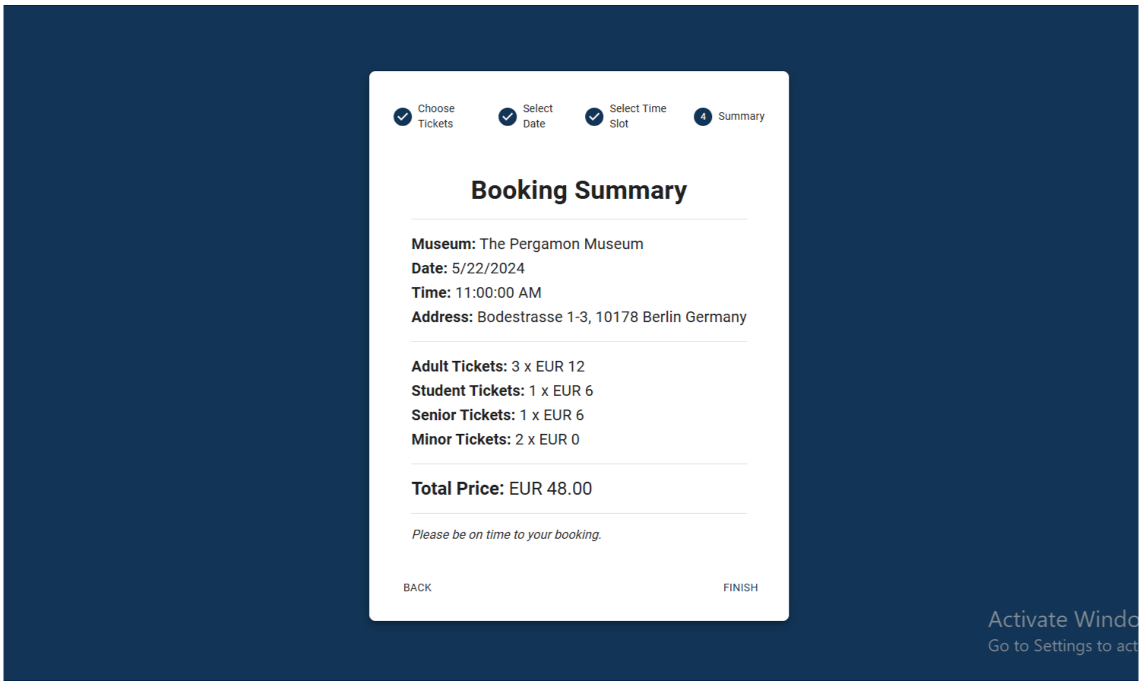

Choose Tickets
Select Date
Select Time Slot
Summary
10:00:00 AM - AVAILABLE: 200
10:30:00 AM - AVAILABLE: 200
11:00:00 AM - AVAILABLE: 200
11:30:00 AM - AVAILABLE: 200
12:00:00 PM - AVAILABLE: 200
12:30:00 PM - AVAILABLE: 200
1:00:00 PM - AVAILABLE: 200
1:30:00 PM - AVAILABLE: 200
2:00:00 PM - AVAILABLE: 200
2:30:00 PM - AVAILABLE: 200
3:00:00 PM - AVAILABLE: 200
3:30:00 PM - AVAILABLE: 200
4:00:00 PM - AVAILABLE: 200
4:30:00 PM - AVAILABLE: 200
5:00:00 PM - AVAILABLE: 200
5:30:00 PM - AVAILABLE: 200
Send the PDF ticket by email
BACK
NEXT

Figure 3.32: Selecting a time slot for tourist booking

Finally, users can see the booking summary, and if all the details are correct, the booking process can be completed. The optional PDF ticket will be sent to the tourist's email address shortly after the submission.



Booking Summary

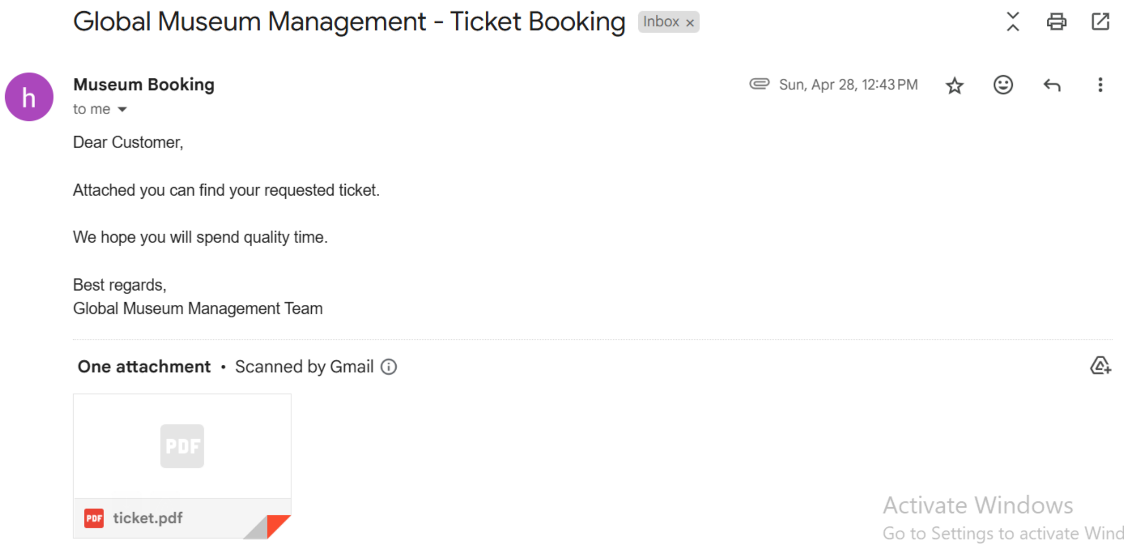
Museum: The Pergamon Museum
Date: 5/22/2024
Time: 11:00:00 AM
Address: Bodestrasse 1-3, 10178 Berlin Germany

Adult Tickets: 3 x EUR 12
Student Tickets: 1 x EUR 6
Senior Tickets: 1 x EUR 6
Minor Tickets: 2 x EUR 0

Total Price: EUR 48.00

Please be on time to your booking.

Figure 3.33: Summary of booking before completion


Global Museum Management - Ticket Booking

Museum Booking
to me
Sun, Apr 28, 12:43 PM

Dear Customer,

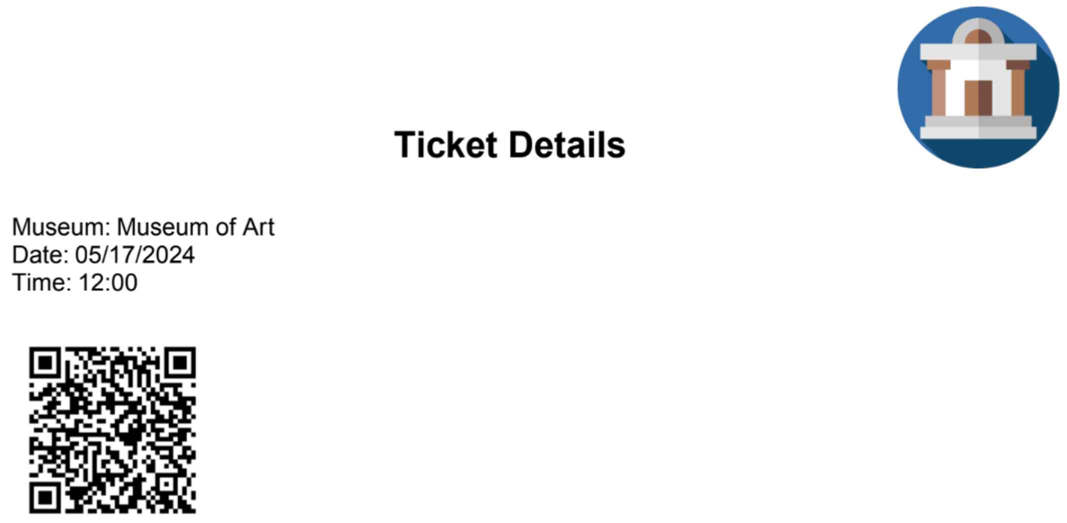
Attached you can find your requested ticket.

We hope you will spend quality time.

Best regards,
Global Museum Management Team

One attachment • Scanned by Gmail

Figure 3.34: Email for a PDF ticket



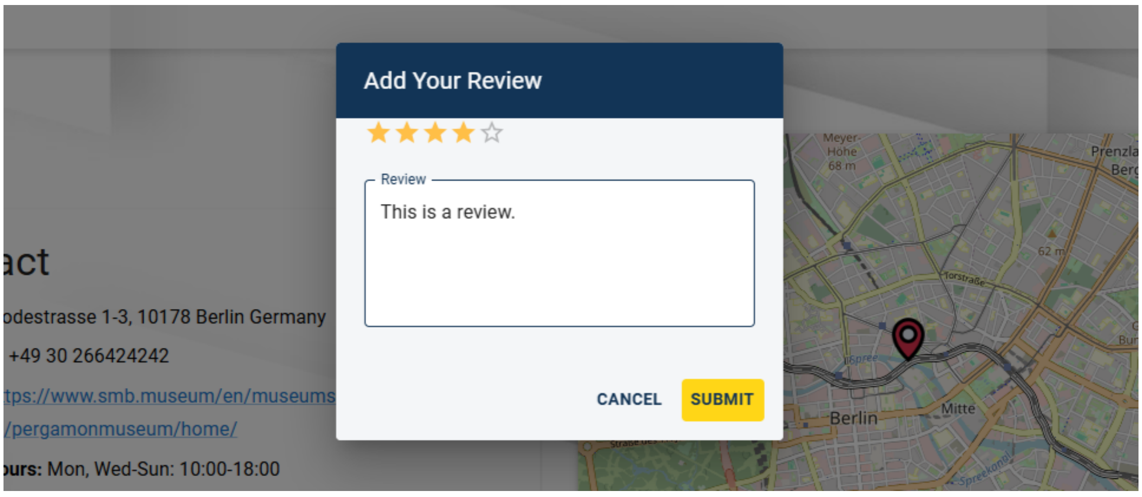
Ticket Details

Museum: Museum of Art
Date: 05/17/2024
Time: 12:00

Figure 3.35: Content of a PDF ticket

Add review

The "Add review" button can be used to add a review to the selected museum. Users can also rate the museum with a 5-star rating system.



Add Your Review

Review

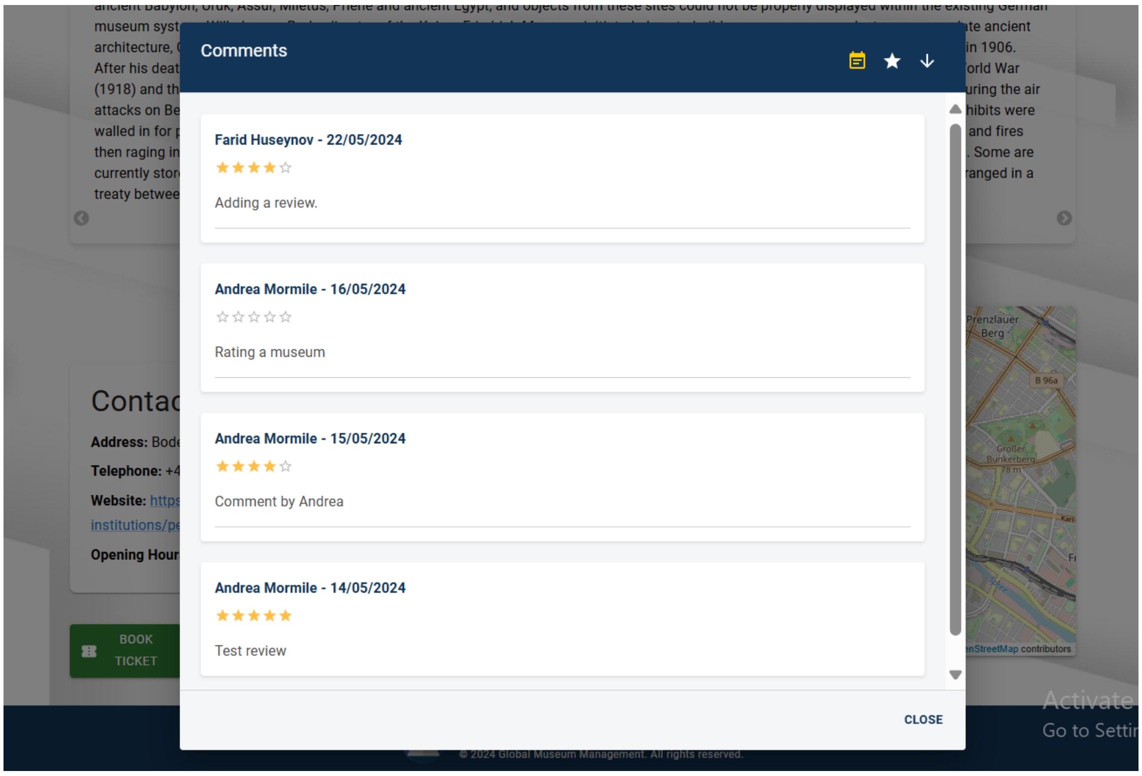
This is a review.

CANCEL
SUBMIT
act
odestrasse 1-3, 10178 Berlin Germany
+49 30 266424242
tps://www.smb.museum/en/museums
/pergamonmuseum/home/
ours: Mon, Wed-Sun: 10:00-18:00
Berlin
Mitte

Figure 3.36: Add review page

View reviews

The "View reviews" button can be used to view all the reviews provided for a museum. Each review contains the name of the reviewer, the date the review was added, and the rating given. Additionally, the reviews can be sorted by the given rating or by the date that the review has been added, both in ascending and descending order.



Comments

Farid Huseynov - 22/05/2024
★★★★☆
Adding a review.

Andrea Mormile - 16/05/2024
☆☆☆☆☆
Rating a museum

Andrea Mormile - 15/05/2024
★★★★☆
Comment by Andrea

Andrea Mormile - 14/05/2024
★★★★★
Test review

CLOSE

Figure 3.37: View reviews page

Busy days diagram

The "Busy days" button can be used for illustrating graphical data about the number of bookings and slots per days of the week. There are two diagrams available: "Overall view" and "Slots view". The first diagram shows the data about the number of bookings by neglecting the number of occupied slots. However, the latter diagram focuses only on the number of slots occupied.

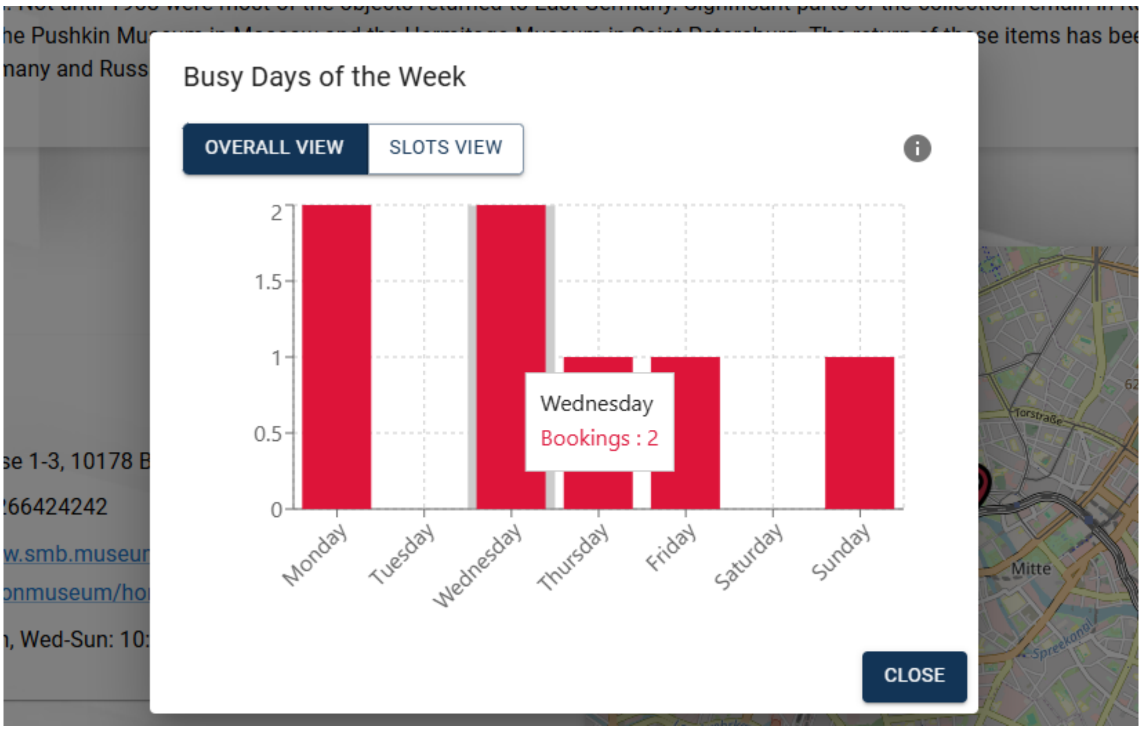


Figure 3.38: "Overall view" diagram

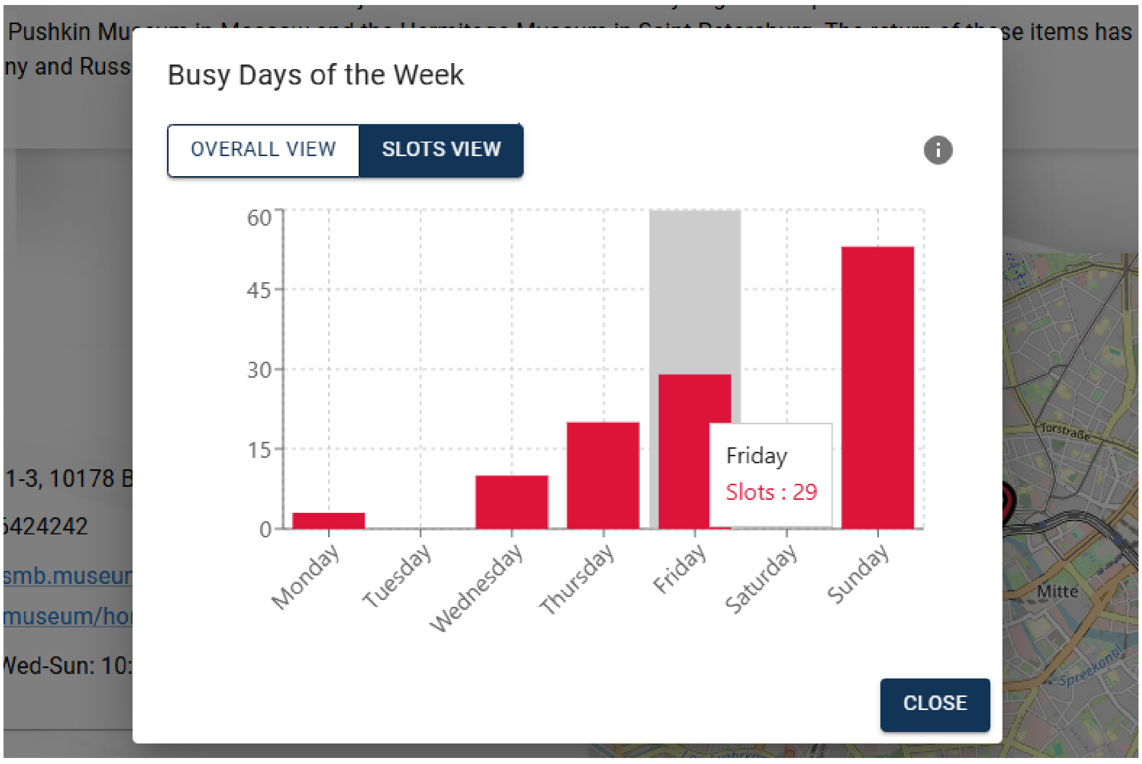
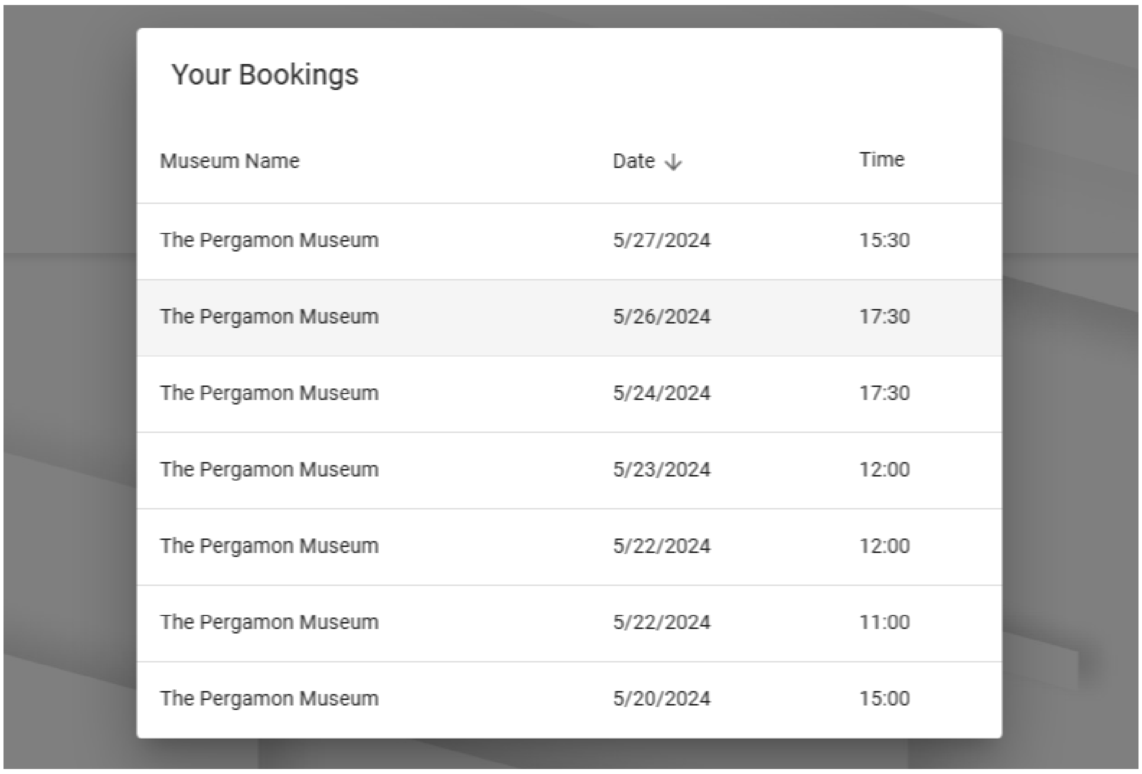


Figure 3.39: "Slots View" diagram

3.5.3 List of bookings and tickets details

Users can also see all of their bookings by click on the booking icon located on the navigation bar. When the icon is clicked, the list bookings will appear. By default, the list is sorted in descending order by date, however it can be sorted in both ascending and descending order, either by date or names of museums.



Your Bookings

Museum Name	Date ↓	Time
The Pergamon Museum	5/27/2024	15:30
The Pergamon Museum	5/26/2024	17:30
The Pergamon Museum	5/24/2024	17:30
The Pergamon Museum	5/23/2024	12:00
The Pergamon Museum	5/22/2024	12:00
The Pergamon Museum	5/22/2024	11:00
The Pergamon Museum	5/20/2024	15:00

Figure 3.40: List of all bookings

In order to view the details of a particular booking, users can select a row which will open the details of the booking in a separate page.

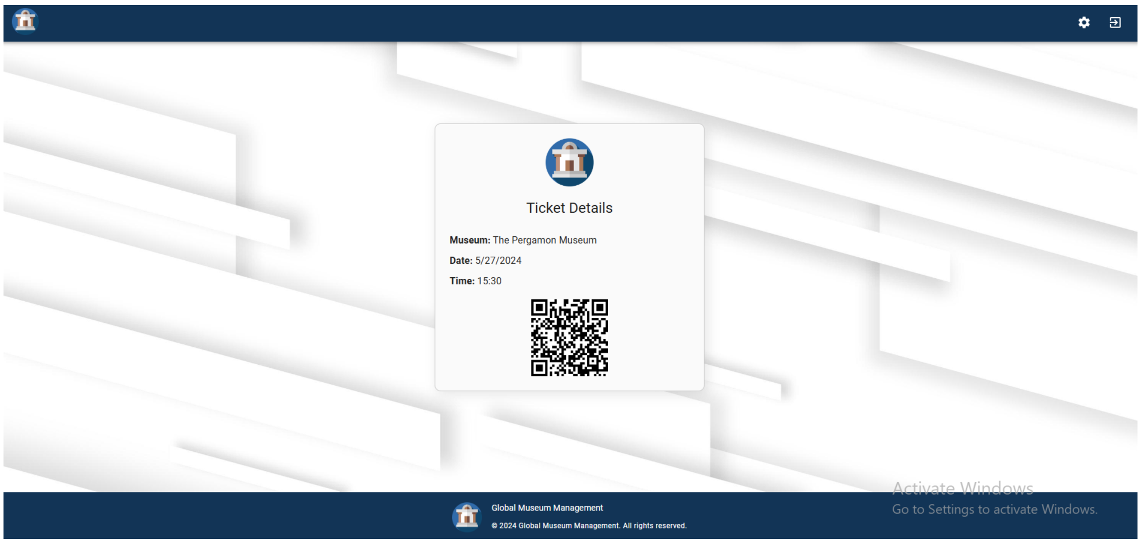
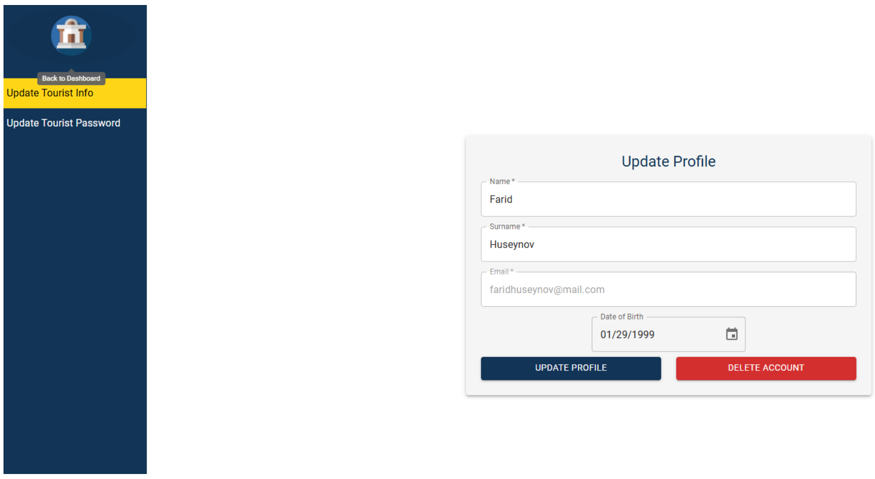

Ticket Details
Museum: The Pergamon Museum
Date: 5/27/2024
Time: 15:30
Global Museum Management
© 2024 Global Museum Management. All rights reserved.
Activate Windows
Go to Settings to activate Windows.

Figure 3.41: Details of a particular booking

3.6 Settings

The settings page is common for both tourist and city accounts, and can be opened by clicking on the gear icon on the navigation bar. The page contains contains a drawer that has options for updating profile information and password. Also, users can click on the application logo to redirect back to account dashboards. In the profile updating page, users can modify their name, surname and date of birth, however they cannot modify their email address. Also, there is an button for deleting accounts permanently in this page, and users need to enter their account password to complete the deletion process.



Update Profile

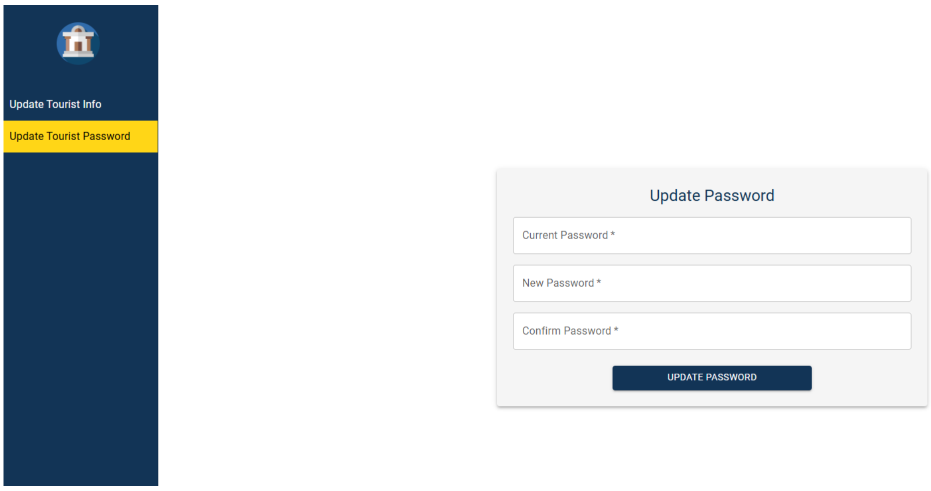
Name *
Farid

Surname *
Huseynov

Email *
faridhuseynov@mail.com

Date of Birth
01/29/1999

Figure 3.42: Updating profile info in settings page



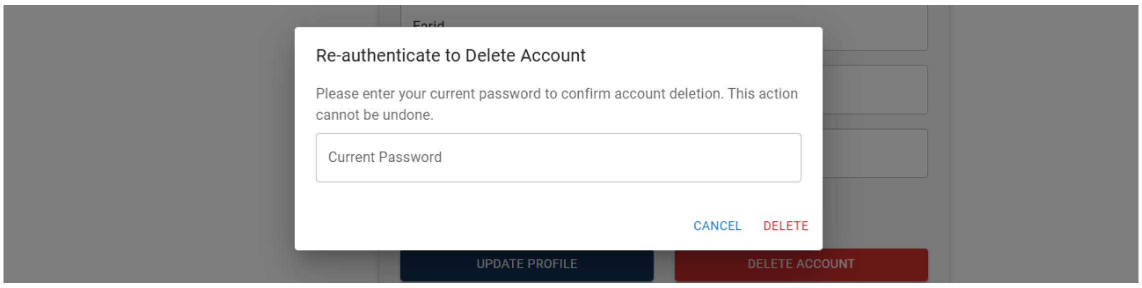
Update Password

Current Password *

New Password *

Confirm Password *

Figure 3.43: Updating password in settings page


Re-authenticate to Delete Account

Please enter your current password to confirm account deletion. This action cannot be undone.

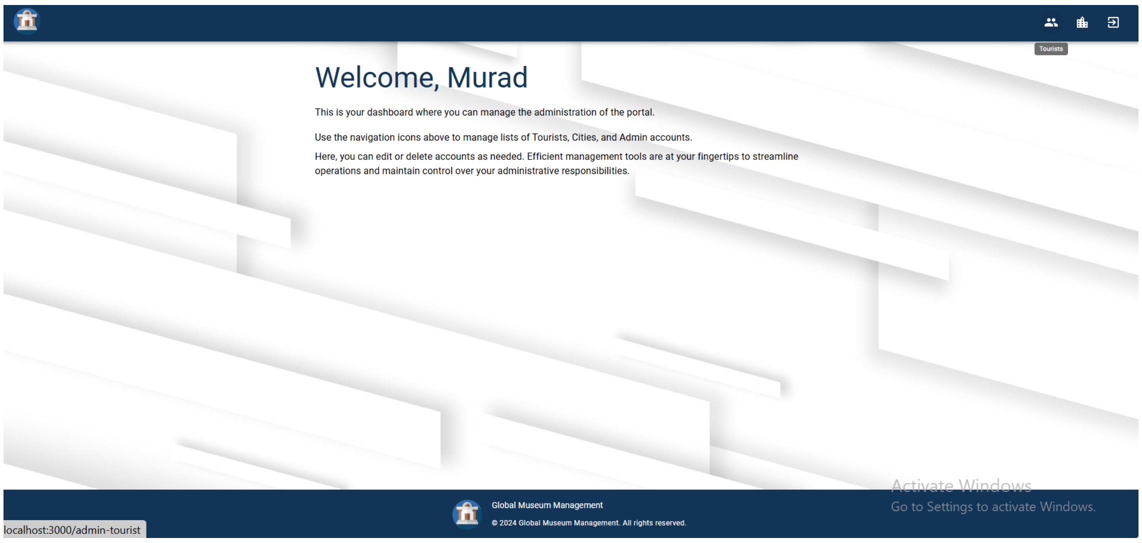
Current Password

Figure 3.44: Reauthentication before account deletion

3.7 Admin Account

3.7.1 Admin Dashboard

If a successful login is made with credentials that belong to an administrator role, the administrator dashboard will be shown. This page serves a welcoming page, providing some general textual information. Administrators can click on tourist or city icons on the navigation bar to redirect to the main pages where they can manage these roles separately.



Welcome, Murad

This is your dashboard where you can manage the administration of the portal.

Use the navigation icons above to manage lists of Tourists, Cities, and Admin accounts.

Here, you can edit or delete accounts as needed. Efficient management tools are at your fingertips to streamline operations and maintain control over your administrative responsibilities.

Global Museum Management

© 2024 Global Museum Management. All rights reserved.

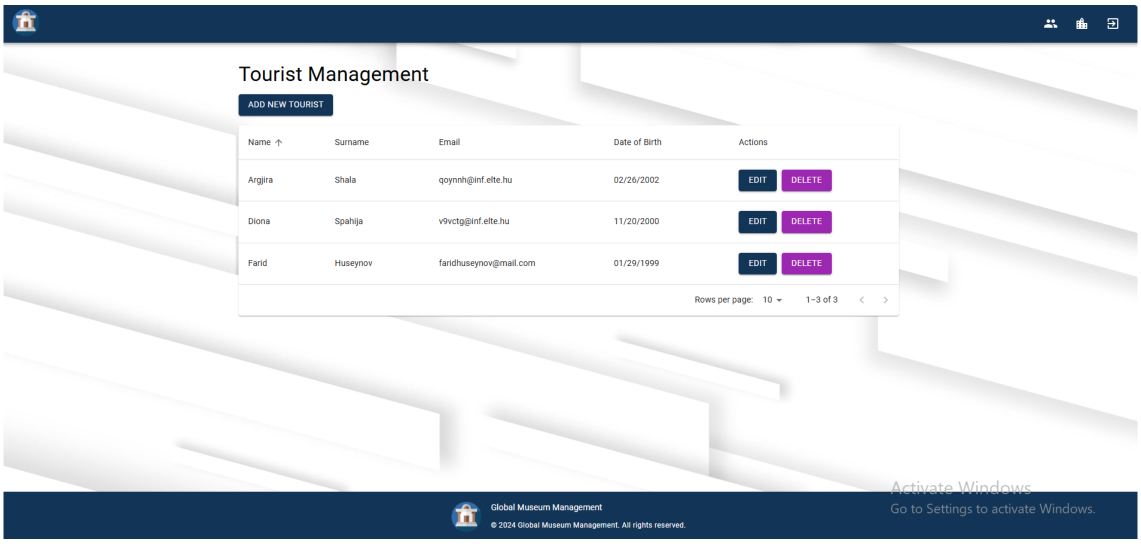
Activate Windows
Go to Settings to activate Windows.

localhost:3000/admin-tourist

Figure 3.45: Admin dashboard

3.7.2 Tourist Management

Administrators can view the list of tourists in tourist management page, and as a default, the list is sorted by name. However, it is optional to make the sorting by surname, email address, and date of birth as well. Each row in the table contains two action buttons which are for editing and deleting tourists. If "delete" button is clicked, a confirmation will be required to complete the process. Also, "Add new tourist" button can be used for registering a new tourist, however administrators cannot assign a password for tourists for security purposes and tourists have to use "forgot password" page from the login screen to set a new password.



Tourist Management

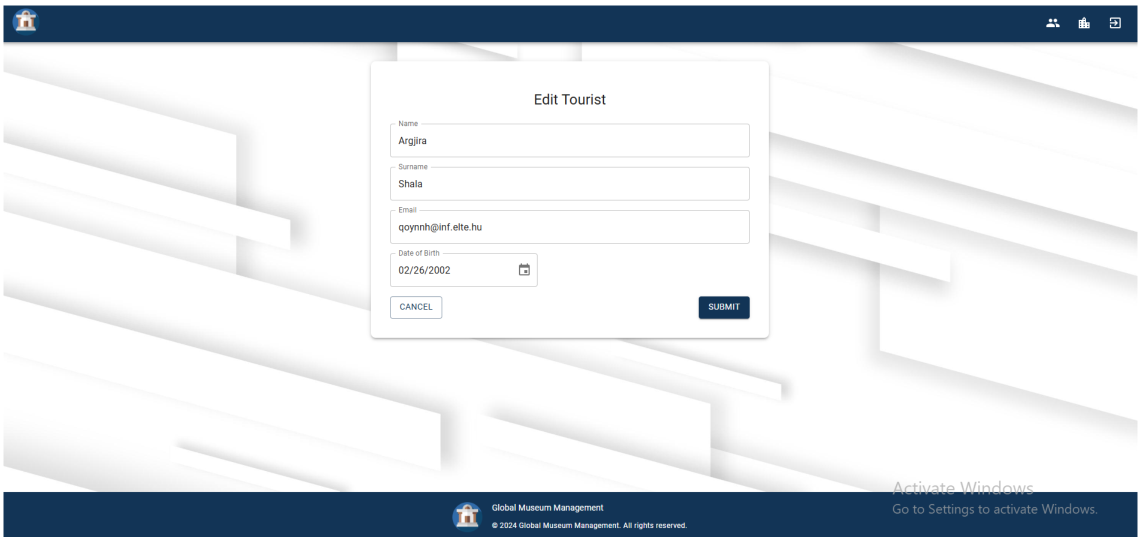
ADD NEW TOURIST

Name	Surname	Email	Date of Birth	Actions
Argjira	Shala	qoynnh@inf.elte.hu	02/26/2002	EDIT DELETE
Diona	Spahija	v9vctg@inf.elte.hu	11/20/2000	EDIT DELETE
Farid	Huseynov	faridhuseynov@mail.com	01/29/1999	EDIT DELETE

Rows per page: 10 1–3 of 3

Global Museum Management
© 2024 Global Museum Management. All rights reserved.

Figure 3.46: Tourist management by administrators



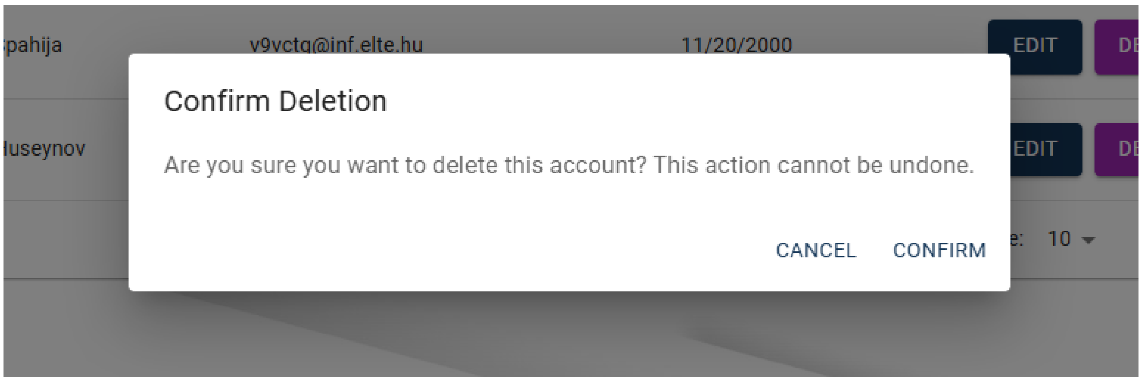
Edit Tourist

Name	Argjira
Surname	Shala
Email	qoynnh@inf.elte.hu
Date of Birth	02/26/2002

CANCEL SUBMIT

Global Museum Management
© 2024 Global Museum Management. All rights reserved.

Figure 3.47: Editing a tourist as an administrator



Confirm Deletion

Are you sure you want to delete this account? This action cannot be undone.

Figure 3.48: Delete confirmation for administrators

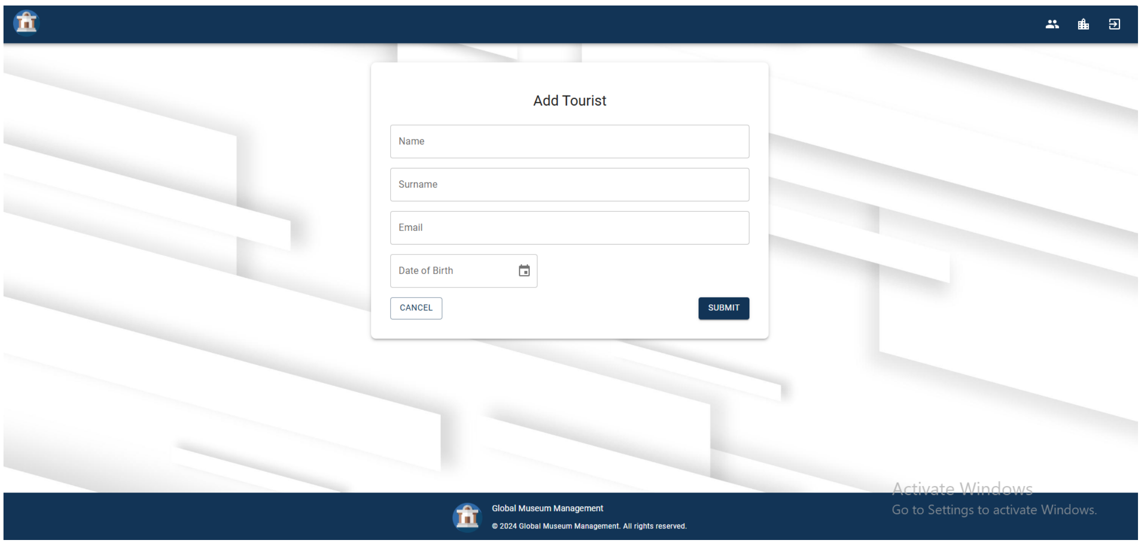
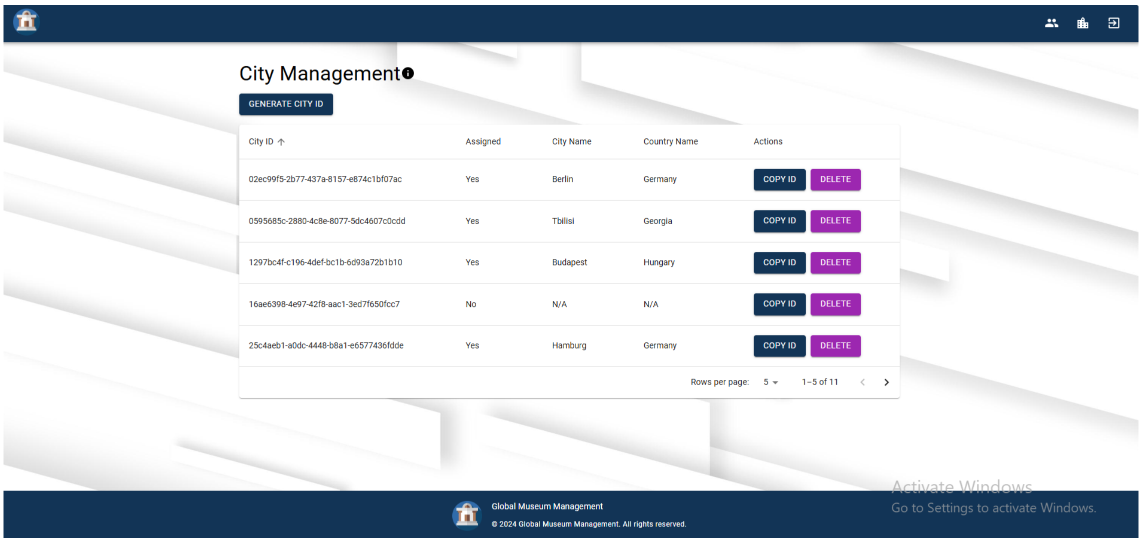


Figure 3.49: Registering a new tourist as an administrator

3.7.3 City Management

Similar to the tourist management page, administrators can view the list city registrars on the city management page. Ordinarily, the table is sorted by city IDs, however it can be done by using city and country names, and by the value of "assigned" column which can be true or false. If "generate city ID" button is clicked, a new row will be added on the table. Administrators can click on a row to edit an already registered city, or complete the form and register a new city. Using the buttons on the "actions" column, administrator can copy city IDs and/or delete a city from the database. The same confirmation for deleting action will be required on city side as well.



City Management

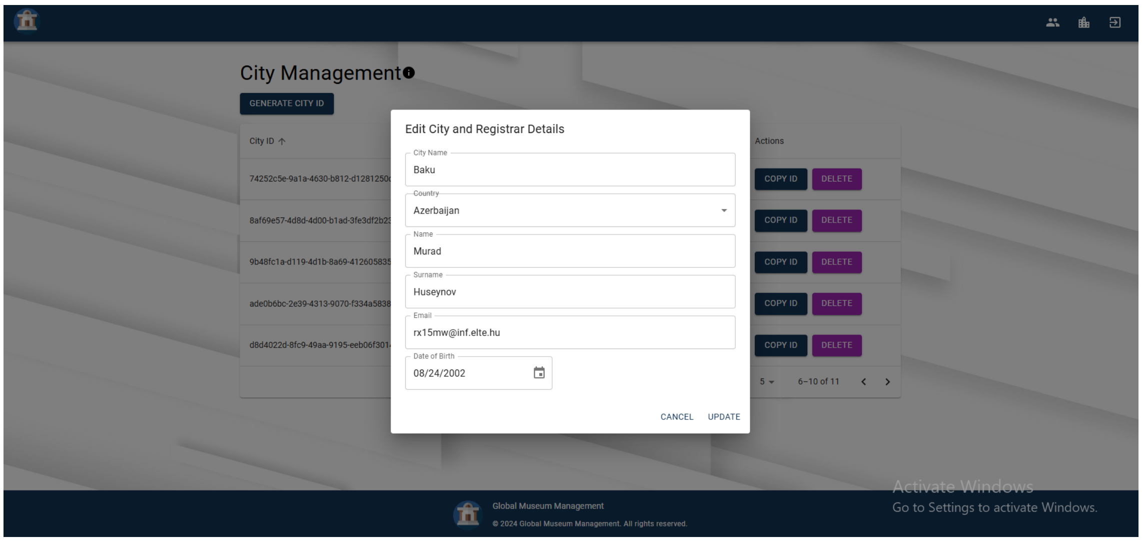
GENERATE CITY ID

City ID	Assigned	City Name	Country Name	Actions
02ec99f5-2b77-437a-8157-e874c1bf07ac	Yes	Berlin	Germany	COPY ID DELETE
0595685c-2880-4c8e-8077-5dc4607c0cdd	Yes	Tbilisi	Georgia	COPY ID DELETE
1297bc4f-c196-4def-bc1b-6d93a72b1b10	Yes	Budapest	Hungary	COPY ID DELETE
16ae6398-4e97-42f8-aac1-3ed7f650fcc7	No	N/A	N/A	COPY ID DELETE
25c4aeb1-a0dc-4448-b8a1-e6577436fdde	Yes	Hamburg	Germany	COPY ID DELETE

Rows per page: 5 1–5 of 11

Global Museum Management
© 2024 Global Museum Management. All rights reserved.

Figure 3.50: City management by administrators



City Management

GENERATE CITY ID

City ID	Actions
74252c5e-9a1a-4630-b812-d1281250	COPY ID DELETE
8af69e57-4d8d-4d00-b1ad-3fe3df2b23	COPY ID DELETE
9b48fc1a-d119-4d1b-8a69-412605835	COPY ID DELETE
ade0b6bc-2e39-4313-9070-f334a5838	COPY ID DELETE
d8d4022d-8fc9-49aa-9195-eeb06f301	COPY ID DELETE

5 6–10 of 11

Edit City and Registrar Details

City Name: Baku
Country: Azerbaijan
Name: Murad
Surname: Huseynov
Email: rx15mw@inf.elte.hu
Date of Birth: 08/24/2002

CANCEL UPDATE

Global Museum Management
© 2024 Global Museum Management. All rights reserved.

Figure 3.51: Registering a new tourist as an administrator

Chapter 4

Developer Documentation

4.1 User Stories

Using a structure of *"As a [user type/ role], I want/can [perform an action] so that [gain a benefit]"*, user stories are a natural description of an application that summarize the features of an application from the perspective of an end-user. They play a communication role between the end-users and developers, so the usage of user stories are crucial to make sure user needs are met. As there are three different roles in the Global Museum Management application, and each role has many unique functions, user stories will help to identify all the necessary functionalities and group them.

4.1.1 List of User Stories

User Story 1: Login and Registration

- As a tourist, I want to use my email and password to log in to my dashboard so that I can look up for museums.
- As a city registrar, I want to use my email and password to log in to my dashboard so that I can manage museum in my city.
- As an administrator, I want to use my email and password to log in to my dashboard so that I can manage tourist and city accounts.

User Story 2: Forgot Password

- As a user, I want to reset my password in case I forget it so that I can access my account again.

User Story 3: City Account

- As a city registrar, I want to add, edit, and delete museum so that I can provide the latest museum information to tourists.
- As a city registrar, I want to include images in museum details, in addition to textual information, so that tourists can see visuals about the museum.
- As a city registrar, I want to update my profile information and modify my password so that I can keep my data up to date and secure.

User Story 4: Tourist Account

- As a tourist, I want to list the countries and cities with museums so that I can organize a visit.
- As a tourist, I want to see comprehensive information about museums, so that I can choose which museum to visit and when.
- As a tourist, I want to book tickets for a desired museums, and receive the booking ticket so that I save time while visiting a museum.
- As a tourist, I want to check out weather reports for cities so that I can plan my visit better.
- As a tourist, I want to add reviews and rate museums so that I can share my experiences with other tourists.
- As a tourist, I want to see all review and ratings of a museum so that I can learn about people's experiences.
- As a tourist, I want to list all my previous bookings and their details so that I can keep the track of my bookings.
- As a tourist, I want to update my profile data and password so that I can assure the safety of my account.

User Story 3: Administrator Account

- As an administrator, I want to manage tourist accounts by adding, editing, and deleting users so that the system remains organized and updated.
- As an administrator, I want to manage city accounts by generating city IDs and controlling registrations so that only authorized cities can register.

4.1.2 Use Case Diagram

The following use case diagram provides a visual summary of the application by illustrating the key functionalities and how they are connected:

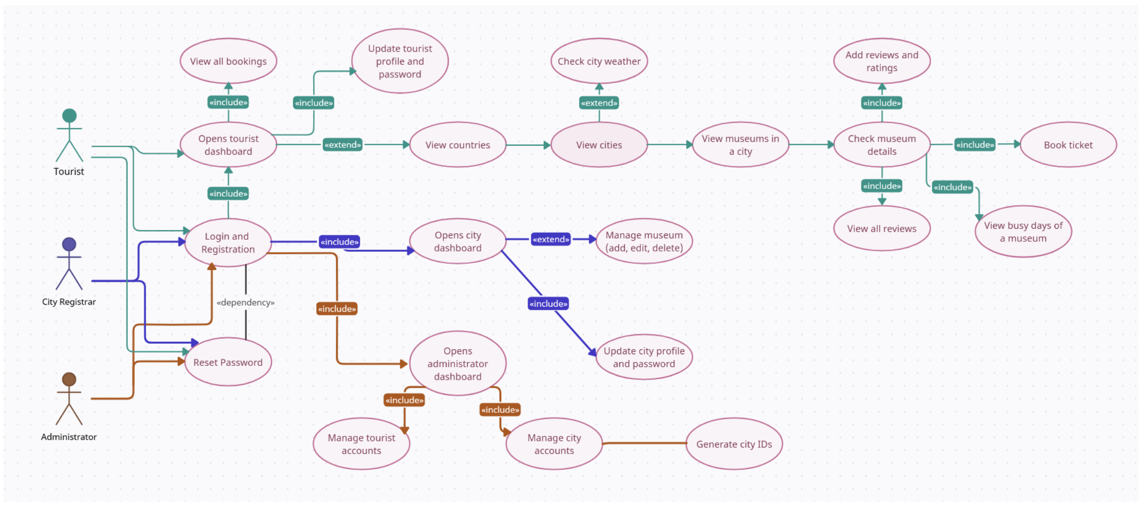


Figure 4.1: Use Case Diagram

4.2 Used Methodologies and Technologies

4.2.1 React.js

Developed by Facebook, React [9] is a multi-functional and popular JavaScript library that has been used as the front-end framework for the development of Global Museum Management application. Because of its component-based architecture, which enables developers to create reusable UI components, React helps to manage big and complex web applications smoothly.

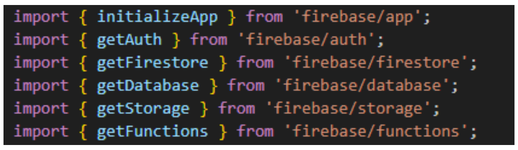
Features

Here are the essential features of React.js that has been used in the development of Global Museum Management application:

- Component-based Architecture: The framework allows web applications to be divided into reusable components, which eliminates code redundancy and promotes flexibility. This feature not only provides independence to many of the utility components, but also boosts the overall performance of the applications.
- State Management: In order to handle user interactions efficiently, React has a built-in state management which allows components and their state changes to be handled internally and across other components. Also, the framework has a context API which can be used for passing global variables anywhere within in the project.
- JSX syntax: React allows the usage of HTML-like markup code within JavaScript using JSX syntax, and this functionality can be used for displaying any errors and warnings to debug any potential issues.
- React Router: React supports centralized navigation between the pages and components through router components. Moreover, routers can be updated via additional functions that can provide secure navigation and disable external access to user-specific pages.

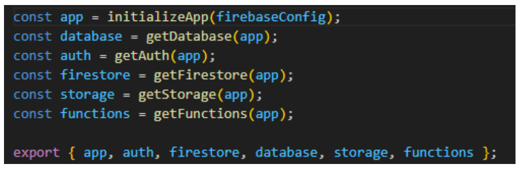
4.2.2 Firebase

Firebase [1], developed by Google, is the core back-end solution for the development of Global Museum Management application. The product's all-in-one support for cloud storage, database, authentication and functions, combined with its easy configuration and user-friendly interface, made Firebase an ideal choice for the back-end of the application.



```
import { initializeApp } from 'firebase/app';
import { getAuth } from 'firebase/auth';
import { getFirestore } from 'firebase/firestore';
import { getDatabase } from 'firebase/database';
import { getStorage } from 'firebase/storage';
import { getFunctions } from 'firebase/functions';
```

Figure 4.2: Importing necessary Firebase services



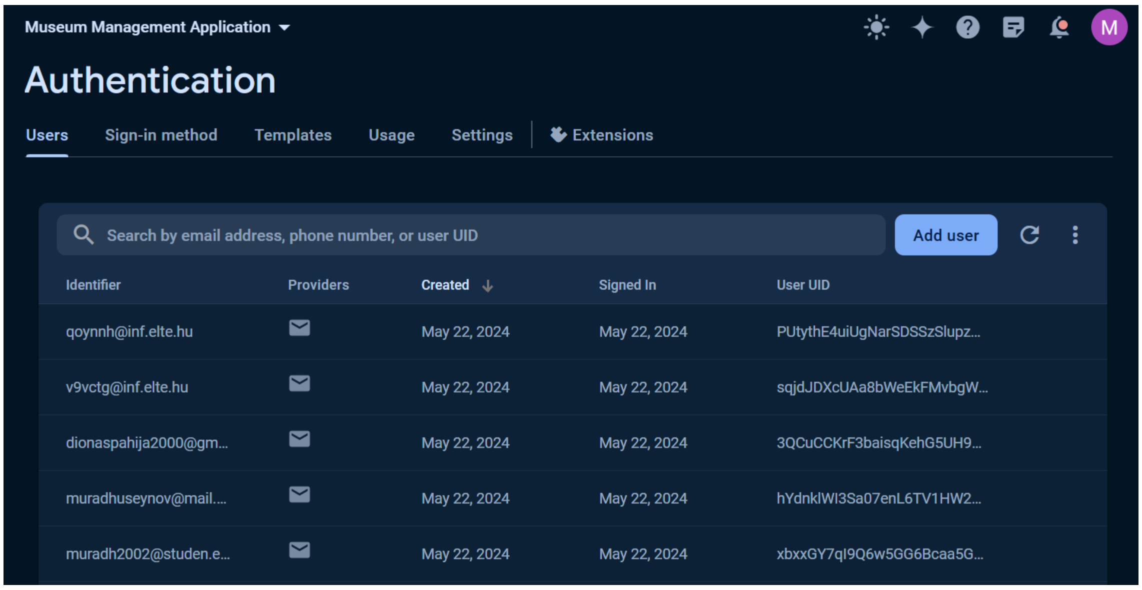
```
const app = initializeApp(firebaseConfig);
const database = getDatabase(app);
const auth = getAuth(app);
const firestore = getFirestore(app);
const storage = getStorage(app);
const functions = getFunctions(app);

export { app, auth, firestore, database, storage, functions };
```

Figure 4.3: Creating instances for Firebase imports

In the configuration file, the desired Firebase services have been imported, initially. Afterwards, instances for each service have been created and exported to make them available within the application.

Firebase Authentication

Firebase Authentication provides a secure mechanism for the registration, login and password management processes within the application. Moreover, the service streamlines the process of creating user authentication, while holding common security protocols.



Museum Management Application
Authentication
Users
Sign-in method
Templates
Usage
Settings
Extensions
M
Search by email address, phone number, or user UID
Add user

Identifier	Providers	Created	Signed In	User UID
qoynnh@inf.elte.hu	✉	May 22, 2024	May 22, 2024	PUtythE4uiUgNarSDSSzSlupz...
v9vctg@inf.elte.hu	✉	May 22, 2024	May 22, 2024	sqjdJDXcUAa8bWeEkFMvbgW...
dionaspahija2000@gm...	✉	May 22, 2024	May 22, 2024	3QCuCCKrF3baisqKehG5UH9...
muradhuseynov@mail....	✉	May 22, 2024	May 22, 2024	hYdnklWI3Sa07enL6TV1HW2...
muradh2002@studen.e...	✉	May 22, 2024	May 22, 2024	xbxxGY7qI9Q6w5GG6Bcaa5G...

Figure 4.4: Firebase Authentication

Firestore Database

Firestore Database is a NoSQL cloud database which helps to save and share data between different roles within the application. Thanks to its support for flexible data structures and real-time data synchronization, all the museum-related data is efficiently stored and updated immediately if a change is triggered.

In order to manipulate the data of the application productively, appropriate collections have been created for each role. Here is the basic summary of the database structure, which shows the fields of each collection and how collections are connected to each other:

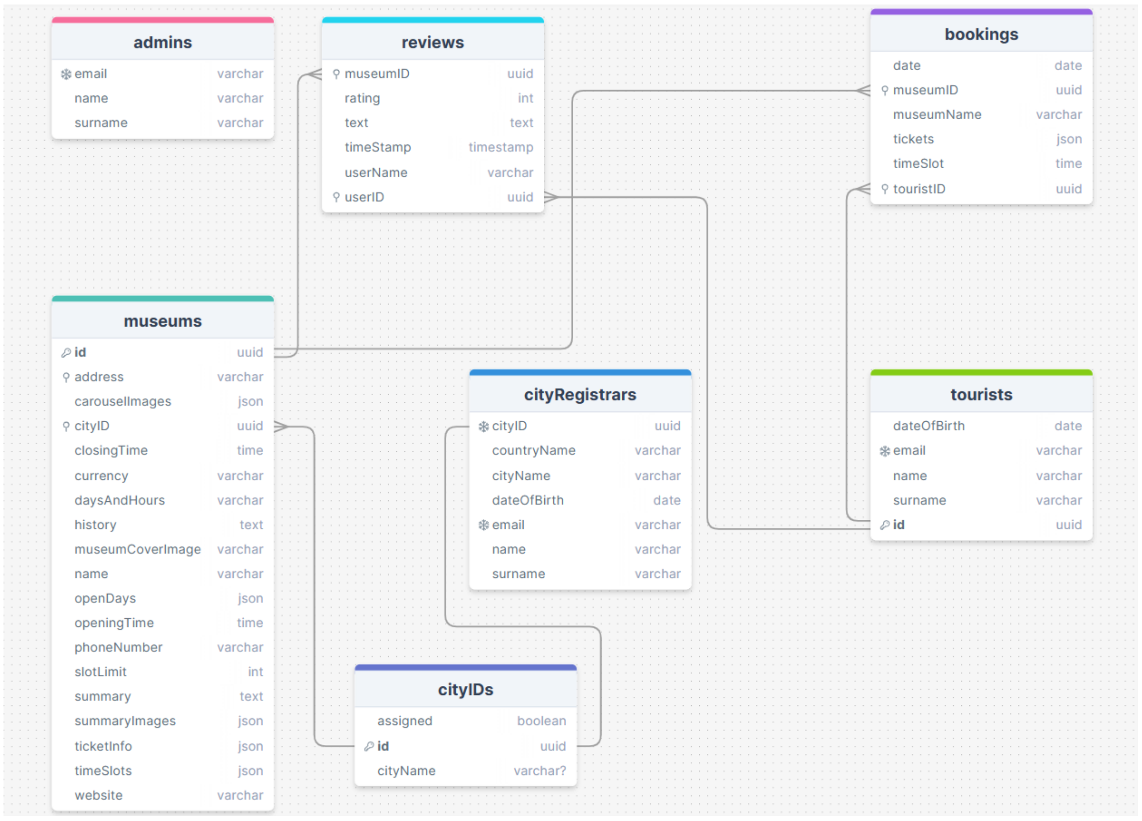


Figure 4.5: Database diagram

- **admins**: This collection stores administrators, and members of this role can only be registered and modified within Firestore. Each document in this collection has an email field which acts as the unique key. It means that there cannot be two accounts with the same role.

- **tourists**: This collection is dedicated to storing tourist accounts. In addition to email field which is a unique key as mentioned before, there is also an *id* field which is the primary key. This key is required to manipulate other tourist-related collections, which are *reviews* and *bookings*.

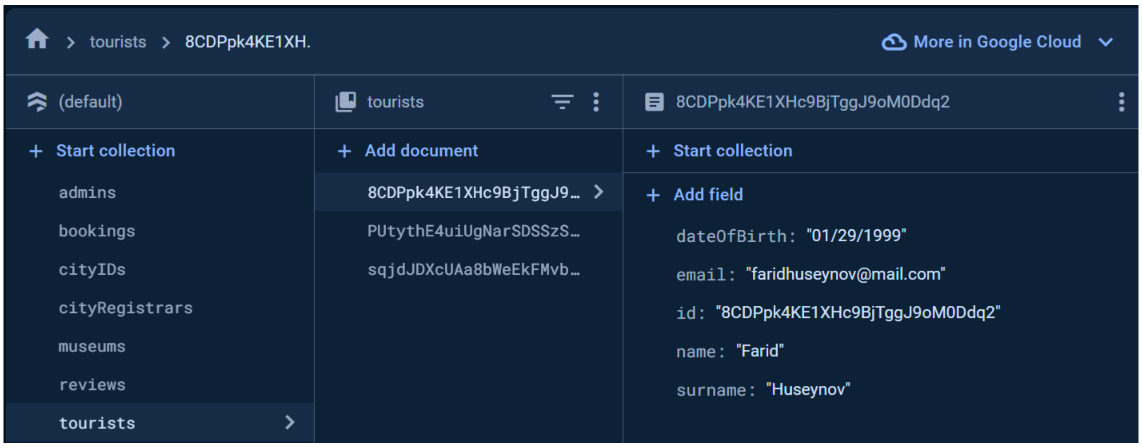
- **cityIDs**: When an administrator clicks on the button to generate a city ID, it is being stored in this collection. Before being assigned to any city, a member of this collection has only 2 fields, which are the boolean *assigned* field with the value false, and the unique city ID which is a string. After being used for a city registration, the boolean value changes to true, and a new field called *cityName* is being added to the document which contains the city name used for the registration.

- **cityRegistrars**: After a city registration is successfully completed, a document is being created in this collection that stores both registrar-related fields, and city/country names, together with the city ID which is also stored in the *cityIDs* collection.

- **museums**: If a new museum is added from the city-side, a document is created in this collection. The document contains its unique museum ID and the city ID to identify which city it belongs to. Since the images are stored in Firebase Storage, the document contains references which are links to those visuals.

- **reviews**: When a tourist leaves comments for a museums, they are saved in this collection. To identify the reviewer and the museum that the review belongs to, both museum ID and tourist ID are stored within the document.

- **bookings**: When a tourist books a ticket for a particular museum, the relevant document is created in this collection. Similar to the logic in *reviews* collection, museum and tourist IDs are being stored here as well, for easy management of bookings.


tourists > 8CDPpk4KE1XH.

(default)
- admins
- bookings
- cityIDs
- cityRegistrars
- museums
- reviews
- tourists

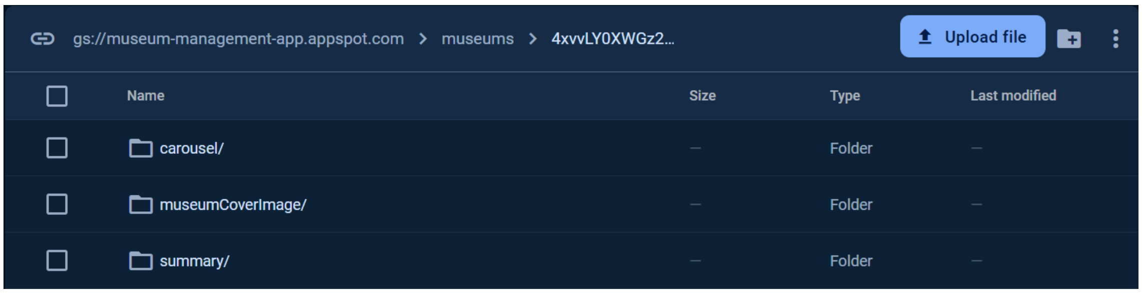
tourists
- 8CDPpk4KE1XHc9BjTggJ9…
- PUtythE4uiUgNarSDSSzS…
- sqjdJDXcUAa8bWeEkFMvb…

8CDPpk4KE1XHc9BjTggJ9oM0Ddq2
```
dateOfBirth: "01/29/1999"
email: "faridhuseynov@mail.com"
id: "8CDPpk4KE1XHc9BjTggJ9oM0Ddq2"
name: "Farid"
surname: "Huseynov"
```

Figure 4.6: Collection if Firebase Database

Firebase Cloud Storage

Cloud Storage allows users to upload user-generated media, such as images and videos, and use them securely. For Global Museum Management application, Storage is required for storing museum visuals. For convenience, images are collected based on the museum IDs, and sorted as history, summary and cover images.

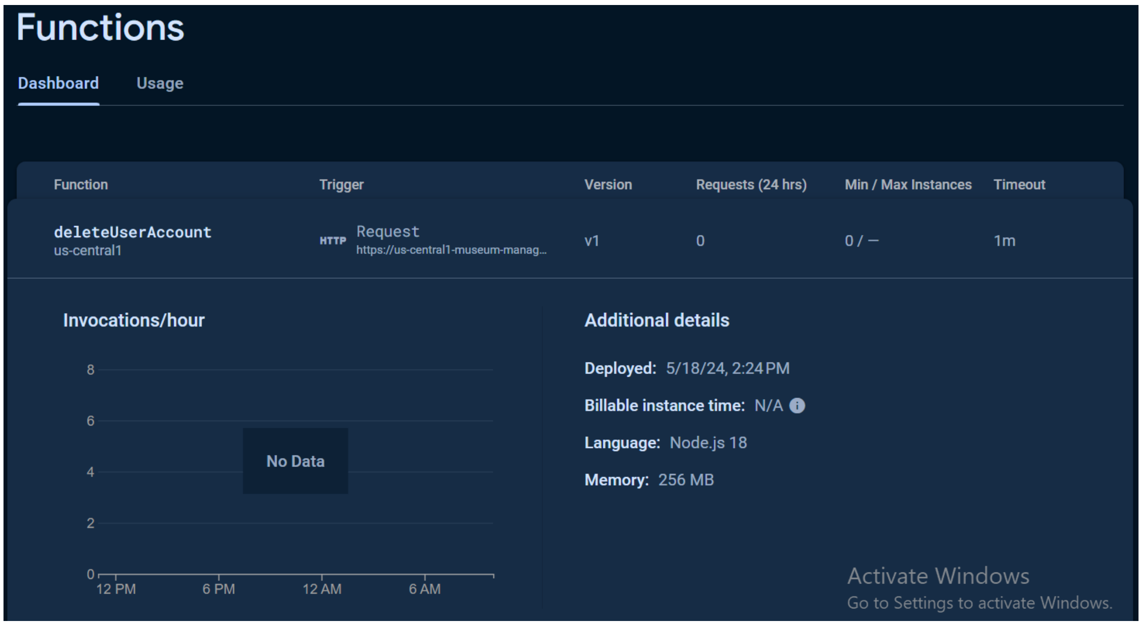

gs://museum-management-app.appspot.com > museums > 4xvvLY0XWGz2…

Name	Size	Type	Last modified
carousel/	–	Folder	–
museumCoverImage/	–	Folder	–
summary/	–	Folder	–

Figure 4.7: Firebase Cloud Storage

Firebase Cloud Functions

Cloud Functions is another tool provided by Firebase that allows users run back-end code in response to the events caused by Firebase or HTTP calls. In the application, the framework is responsible for successful deletions of tourist and city accounts by giving administrative access to the users registered under *admins* collection.


Functions

Dashboard Usage

Function	Trigger	Version	Requests (24 hrs)	Min / Max Instances	Timeout
deleteUserAccount us-central1	HTTP Request https://us-central1-museum-manag...	v1	0	0 / —	1m

Invocations/hour

Additional details

Deployed: 5/18/24, 2:24 PM

Billable instance time: N/A

Language: Node.js 18

Memory: 256 MB

Activate Windows
Go to Settings to activate Windows.

Figure 4.8: Firebase Cloud Functions

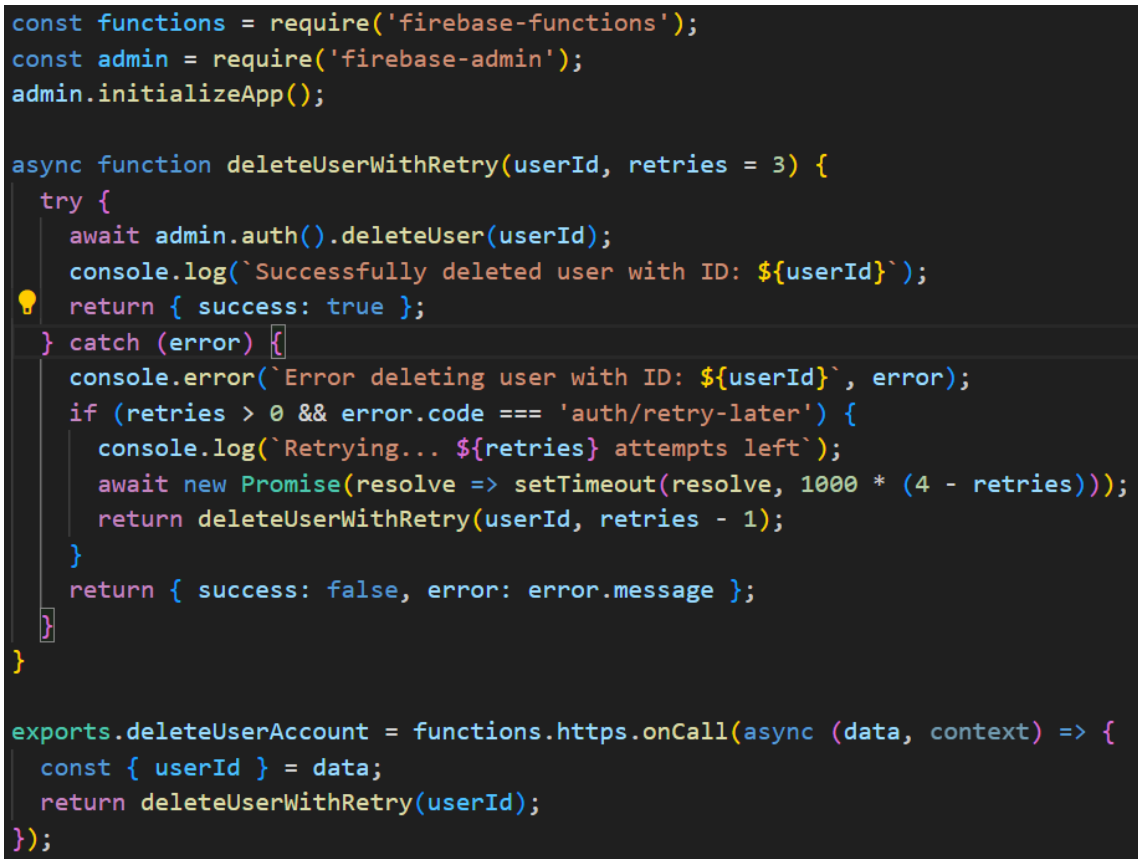

```
const functions = require('firebase-functions');
const admin = require('firebase-admin');
admin.initializeApp();

async function deleteUserWithRetry(userId, retries = 3) {
  try {
    await admin.auth().deleteUser(userId);
    console.log(`Successfully deleted user with ID: ${userId}`);
    return { success: true };
  } catch (error) {
    console.error(`Error deleting user with ID: ${userId}`, error);
    if (retries > 0 && error.code === 'auth/retry-later') {
      console.log(`Retrying... ${retries} attempts left`);
      await new Promise(resolve => setTimeout(resolve, 1000 * (4 - retries)));
      return deleteUserWithRetry(userId, retries - 1);
    }
    return { success: false, error: error.message };
  }
}

exports.deleteUserAccount = functions.https.onCall(async (data, context) => {
  const { userId } = data;
  return deleteUserWithRetry(userId);
});
```

Figure 4.9: Configuration of a Cloud Function for user deletion by administrator

4.2.3 Material UI

Material UI [5] is a popular open-source React library that follows Google's Material Design. Because of its rich and customizable set of reusable components, it is possible to create modern and responsive applications. The library has been chosen as the cornerstone of the UI/UX design of the Global Museum Management application for several reason listed below:

1. Rich Component Library: Advanced components, such as date pickers, sliders, and tables have been used extensively during the development of the application, along side with the basic elements such as buttons, dialogs, cards, forms and so on.

2. Creating of Custom Themes: Thanks to the various theming functionalities of Material UI, it is possible to create global and individual themes for components, using APIs such as *makestyles* and *styled*.

3. Responsive Design: To provide the best experience for users, it is important to make sure the application is adjustable to every screen size. Material UI provides a special hook called *useMediaQuery* that listens for matches to a CSS media query that ensures the screens are responsive.

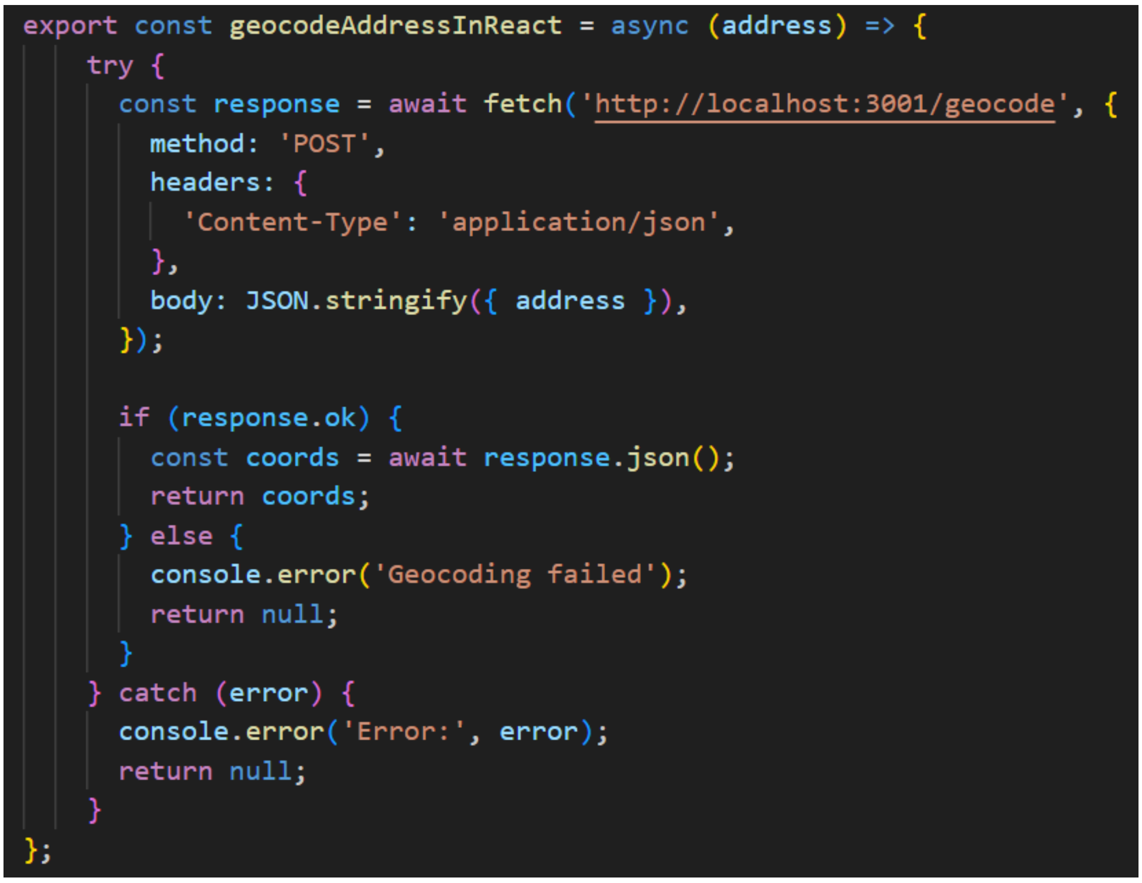
4.2.4 Node.js Server

Along side with Firebase, the Node.js [6] server in Global Museum Management application plays a crucial role in the management of server-side processes, such as sending PDF tickets during booking confirmations, geocoding, and fetching weather data of a city.

External APIs

In order to incorporate some of the functionalities, the application utilizes different external APIs to make sure users interact with real-time and continuous data.

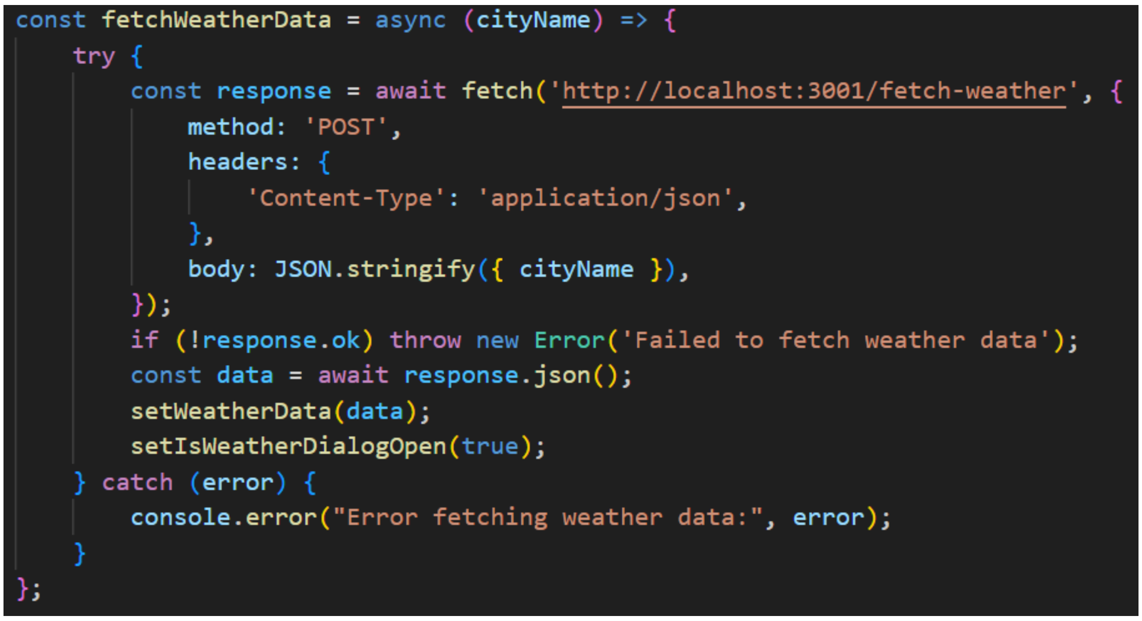
- Google Maps Geocoding API [2]: By using geographical coordinates like latitude and longitude, this API helps to pinpoint a location on the map. This functionality has successfully been implement in the museum detail pages, for both city and tourist sides.



```
export const geocodeAddressInReact = async (address) => {
    try {
      const response = await fetch('http://localhost:3001/geocode', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ address }),
      });

      if (response.ok) {
        const coords = await response.json();
        return coords;
      } else {
        console.error('Geocoding failed');
        return null;
      }
    } catch (error) {
      console.error('Error:', error);
      return null;
    }
};
```

Figure 4.10: Making an API call for Geocoding

- OpenWeatherMap API [8]: This API is necessary for displaying weather reports by using the name of a city.

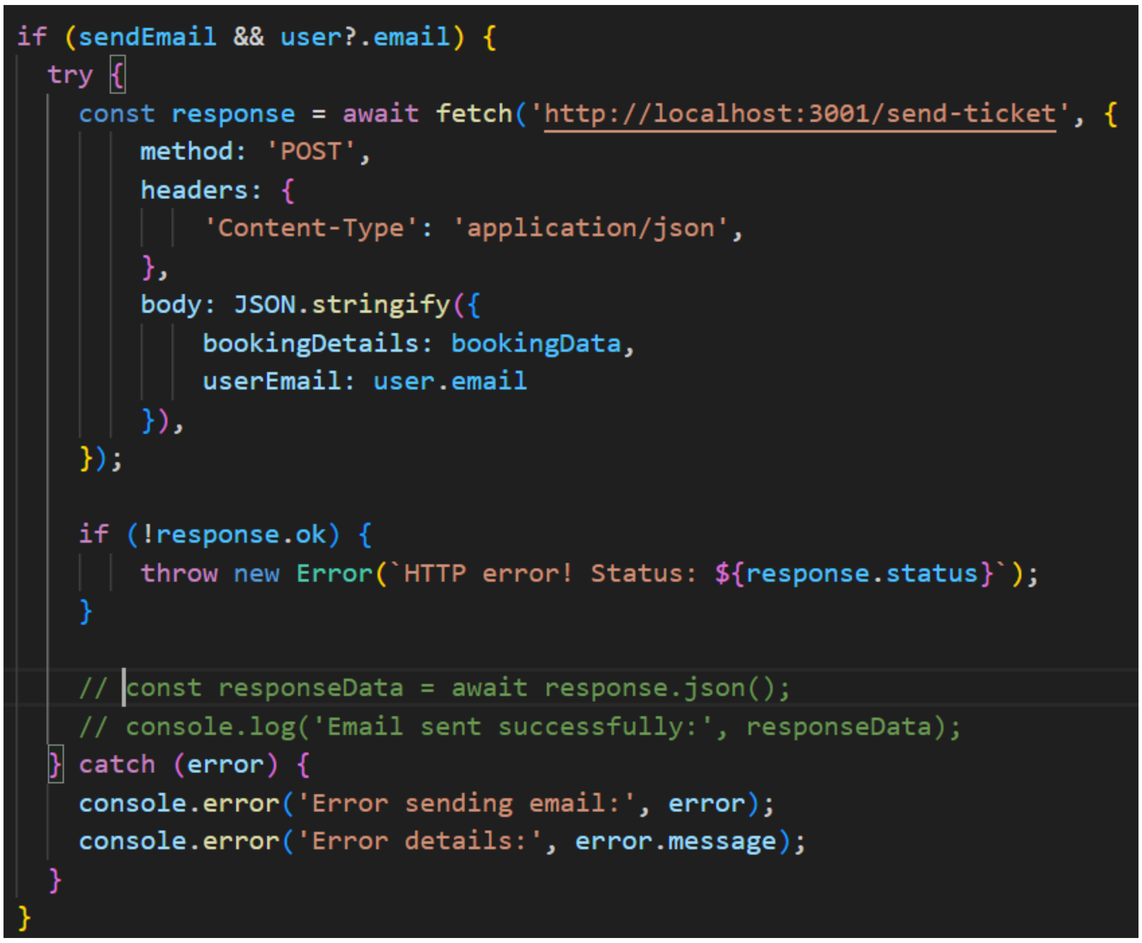


```javascript
const fetchWeatherData = async (cityName) => {
    try {
        const response = await fetch('http://localhost:3001/fetch-weather', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify({ cityName }),
        });
        if (!response.ok) throw new Error('Failed to fetch weather data');
        const data = await response.json();
        setWeatherData(data);
        setIsWeatherDialogOpen(true);
    } catch (error) {
        console.error("Error fetching weather data:", error);
    }
};
```

Figure 4.11: Making an API call for fetching weather data

Node Mailer

Node Mailer [7] is used for handling the email functionality of sending a PDF ticket in the application. This is important if tourists want to document their visits, and also to provide a complete service to the users.



```
if (sendEmail && user?.email) {
  try {
    const response = await fetch('http://localhost:3001/send-ticket', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({
          bookingDetails: bookingData,
          userEmail: user.email
        }),
    });

    if (!response.ok) {
        throw new Error(`HTTP error! Status: ${response.status}`);
    }

    // const responseData = await response.json();
    // console.log('Email sent successfully:', responseData);
  } catch (error) {
    console.error('Error sending email:', error);
    console.error('Error details:', error.message);
  }
}
```

Figure 4.12: Making an API call for sending a PDF ticket

4.3 Program Structure

The architecture of the Global Museum Management application promotes reusability and accessibility by following clean and organized code rules. The architecture successfully differentiates reusable components, data structures, and API connections. This approach not only supports better development process, but also eases the process of testing, as each functionality can be tested individually.

4.3.1 Logical Structure

As mentioned before, the Global Museum Management application uses React.js for its front-end development and while organizing the components and pages, keeping the hierarchical and flexible order was one of the main key points. Below is the main structure of the *src/* directory which contains the essential components:

```
src
│
├────────→__mocks__
├────────→assets
├────────→components
├────────→contexts
├────────→hooks
├────────→pages
├────────→router
└────────→styles
```

Each folder includes:

- **__mocks__**: Mocks which are necessary to for performing Firebase test during the testing phase.
- **assets**: The user-generated media elements that were including the application interface.
- **components**: The common reusable components.
- **hooks**: The hook elements what allow for React states.
- **pages**: Page-level components, and each of them represent a distinct route in the application.
- **router**: The router components that allows secure navigation between different pages.
- **styles**: Separated style elements for each component.

4.3.2 Physical Structure

Data Structures

The application synthesizes a number of data structures, each guaranteeing a smooth manipulation of data and enabling good performance during execution by efficiently handling memory. The application also encapsulates advanced data structures such as maps and date formats, aligning with the features of the application.

Databases

As mentioned earlier, the application uses Firebase as the main database. While Firestore Database efficiently manages any data related changes by providing real-time updates, Cloud Storage is saving all the media elements securely.

Third-Party APIs

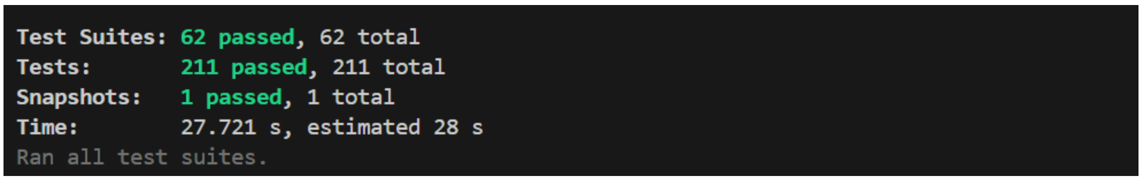
By integrating third-party APIs on the application, users can get weather-related data for any city, see the pinpointed location of a museum during their search, and receive optional PDF tickets for their bookings, enhancing user experience.

4.4 Testing

To test the Global Museum Management application, unit tests, integration tests, and manual testing have been used. React testing library and Jest [4] were chosen for React and Firebase testing, while Mocha is used for Node.js tests.

Every individual component has been tested, and each test file is located in the same folder as its corresponding component. In order to run the React and Firebase tests, users should follow the steps below:

1. Locate the project folder and open a command prompt or terminal inside.
2. Navigate to React project folder by using the command:
```
cd museum-app1
```
3. Run the tests using the following command:
```
npm test
```
NOTE: If testing does not start automatically after running the command, users need to press the letter ***"a"*** *on keyboard.*
4. After all the tests finish running, users will see data about testing:



```
Test Suites: 62 passed, 62 total
Tests:       211 passed, 211 total
Snapshots:   1 passed, 1 total
Time:        27.721 s, estimated 28 s
Ran all test suites.
```

Figure 4.13: Result of React and Firebase tests

In order to run the Node.js tests, users should follow the steps below:

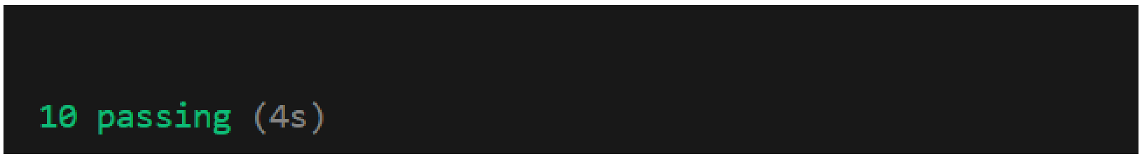
1. Locate the project folder and open a command prompt or terminal inside.
2. Navigate to React project folder by using the command:

```
cd node-server
```

3. Run the tests using the following command:

```
npm test
```

4. After all the tests finish running, users will see data about testing:



```
10 passing (4s)
```

Figure 4.14: Result of React and Firebase tests

Bibliography

[1] Firebase
https://firebase.google.com/docs.

[2] Geocoding API
https://developers.google.com/maps/documentation/geocoding/overview.

[3] Git
https://git-scm.com/downloads.

[4] Jest
https://jestjs.io/.

[5] Material UI
https://mui.com/.

[6] Node.js and npm
https://nodejs.org/en/download.

[7] Nodemailer
https://www.nodemailer.com/.

[8] OpenWeather API
https://openweathermap.org/api.

[9] React
https://react.dev/.

[10] Visual Studio Code
https://code.visualstudio.com/download.