ClassWay
[Eng.Hadeel Hakami]
[Mars-April 2025]

Table of Contents

Abstract

The **ClassWay** project aims to provide an innovative solution for campus navigation. Through an easy-to-use interface, the application allows users to quickly access various locations within the campus, such as offices, classrooms, cafeterias, libraries, and laboratories. It also features an interactive map that enables users to easily navigate between different locations using Google Maps technology.

Technologies Used:

- **HTML & CSS**: For building the layout and visual design of the pages.
- **Bootstrap**: To structure the layout and ensure responsiveness across devices.
- **JavaScript**: For interactive elements and user actions, such as page navigation and map integration.
- **Google Maps API**: To embed and interact with the campus map

ClassWay Downloads:

INSTALL DEPENDENCIES

The project relies on several tools and libraries to function correctly. Follow the steps below based on the tools you are using.

FOR HTML, CSS, AND BOOTSTRAP:

- BOOTSTRAP: The project uses the Bootstrap framework for responsive design. No additional installation is required for Bootstrap because we are linking to it via a CDN in the HTML file.

FOR JAVASCRIPT (FOR MAP AND PAGE NAVIGATION):

- GOOGLE MAPS API: The project integrates Google Maps to show the campus map. You need an API key to use Google Maps. Here's how to obtain one:
  1. Go to the Google Cloud Console.
  2. Create a new project.
  3. Enable the GOOGLE MAPS JAVASCRIPT API.
  4. Create an API key and replace `YOUR_API_KEY` in the script tag with your actual key

```
html
<script
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
async defer></script>
```

FOR FIREBASE (OPTIONAL FOR DATABASE INTEGRATION):

3. OPEN THE PROJECT IN YOUR CODE EDITOR

Once you've cloned the repository and installed any required dependencies (like Firebase or Google Maps API), open the project in your code editor (e.g., VS Code, Sublime Text).

4. RUNNING THE PROJECT LOCALLY

To run the project on your local machine:

- Open the project folder and locate the `index.html` file.
- You can open the HTML file directly in your browser, or if you prefer, use a local server (like using the LIVE SERVER extension in VS Code).

Functional

1. USER INTERFACE (UI)

- RESPONSIVE LAYOUT: The website is fully responsive, ensuring it works across various screen sizes, from desktop to mobile devices.
- NAVIGATION BAR: At the top, there is a navigation bar with a logo and a button labeled "Desk" which redirects users to the desk page for more specific services or information.
- CONTENT DISPLAY: The homepage displays a campus map and an image header. The map is interactive, allowing users to zoom in, zoom out, and explore different campus areas.

2. INTERACTIVE CAMPUS MAP

- GOOGLE MAPS INTEGRATION: The map uses the Google Maps API to display the campus layout. It helps users locate important facilities on campus.
- MAP CUSTOMIZATION: The map allows for basic customization (like showing markers) to indicate specific points of interest (e.g., classrooms, offices).
- ZOOM & PAN FEATURES: Users can zoom into different areas of the campus and pan the map to explore.
- MAP STYLE: The map can be styled to fit the design and aesthetic of the website.

3. FACILITY CARDS (ON DESK PAGE)

- CARD LAYOUT: The desk page displays a set of interactive cards. Each card represents a campus facility, including:
  - CLASSROOMS
  - OFFICES
  - CAFETERIAS
  - LIBRARIES
  - LABS
- CONTENT ON EACH CARD:
  - ICON: An appropriate icon is displayed on each card, such as a person icon for offices or a book icon for libraries.
  - FACILITY INFORMATION: Each card displays key details about the facility:
    - NAME: The name of the facility.
    - ROOM NUMBER: If applicable, the room or office number is displayed.
    - WORKING HOURS: The operational hours for the facility are shown.
  - STYLING: The cards have rounded corners, shadows, and hover effects, making the interface visually appealing and easy to navigate.

4. NAVIGATION BETWEEN PAGES

- HOME TO DESK PAGE NAVIGATION: The "Desk" button at the top of the page allows users to navigate to a new page where they can view more details about the various campus facilities.

- BACK NAVIGATION (OPTIONAL): On the desk page, a back button or other UI components can be added (not explicitly mentioned, but useful for user experience) to take users back to the homepage.

5. BOTTOM BAR

- FOOTER: A bottom bar appears at the bottom of the page, providing copyright information and a nice touch of design with a gradient background.
- CONSISTENCY: The bottom bar is styled consistently across different pages for a uniform look.

6. BUTTON WITH GRADIENTS

- The "Desk" button uses a gradient color scheme, making it stand out visually. The button has rounded corners and smooth hover transitions, ensuring an interactive and modern design.

7. FIREBASE INTEGRATION (OPTIONAL)

- Firebase could be used to store and retrieve user data (for example, user preferences, location settings, or feedback), though this feature is optional based on your project requirements.
- Firebase allows for seamless real-time updates and backend services if you choose to incorporate them.

8. CROSS-DEVICE COMPATIBILITY

- The design of the website ensures that it works well on both mobile and desktop platforms. This is crucial to provide an optimal experience for all users, no matter the device they're using.

9. CUSTOMIZABLE MAP MARKERS

- The Google Maps integration allows developers to customize markers, routes, and other features on the map. This can be extended to indicate specific locations on campus such as classrooms, cafeterias, or labs.

10. LOCALIZATION

- The project includes Arabic localization for button labels and text, providing accessibility for Arabic-speaking users. This makes the application more inclusive and accessible to a wider audience.

Future Plans

1. ADD TEACHER OFFICE TRACKING

- OBJECTIVE: Allow users to track the location and availability of teacher offices in real time.
- DETAILS:
  - Teachers' offices will be represented on the map, with the ability to display their availability for meetings or office hours.
  - This feature will enable students and staff to find available teachers, check office locations, and schedule appointments based on real-time data.
  - INTEGRATION WITH CALENDAR: The system can be integrated with the teachers' personal schedules to show whether they are available or busy at a given time.

2. ADD TRACKING FOR CAMPUS FACILITIES AND CLASSROOMS

- OBJECTIVE: Provide real-time tracking and availability of various campus facilities, such as classrooms, cafeterias, libraries, and labs.
- DETAILS:
  - Users can check the status of specific classrooms or facilities to see if they are occupied or free.
  - This feature can help students and faculty quickly identify available rooms or areas for meetings, study sessions, or group activities.
  - REAL-TIME UPDATES: The status of rooms can be updated in real time based on the current schedule and usage.

3. IMPLEMENT A DATABASE FOR TEACHER OFFICE INFORMATION

- OBJECTIVE: Create a comprehensive database that stores and manages all the relevant information about teachers' offices on campus.
- DETAILS:
  - The database will include data such as the office number, teacher name, department, contact information, office hours, and any other important details related to the office.
  - SEARCHABLE DATABASE: Users will be able to search for teacher offices by name, department, or office number.
  - DYNAMIC UPDATES: The database will be connected to a backend system (e.g., Firebase) that allows administrators to easily update office information, ensuring it remains accurate and current.

4. FUTURE ENHANCEMENTS FOR USER INTERACTION

- USER PROFILES: Allow users (students, teachers, and staff) to create profiles and save their preferred office or facility locations.
- NOTIFICATIONS: Users will receive notifications when their desired office or facility becomes available.

Sources

1. **User Interface**
   https://www.figma.com/design/JxqROO1aTWLWUBu43ST2jO/ClassWay?node-id=0-1&t=TVTKjufBQ8LyKBw9-1

2. **Source Codee**

https://github.com/Haxi2002/ClassWay-Version-1-.git

Additional Recommendation

ClassWay Project

To ensure smooth development, effective collaboration, and future scalability of the ClassWay project, consider the following suggestions:

1. Create a Dedicated Organization for the Project with a Custom Email for Cloud Services

Objective: Set up a dedicated organization and email account specifically for the project to streamline cloud service management and communication.

Cloud Services: Utilize a custom email associated with the project to access and manage cloud services like Firebase, Google Maps, or other external APIs. This ensures that all credentials and services are maintained in a professional and secure manner.

Better Security: A dedicated email ensures that only authorized individuals have access to sensitive project resources, reducing the risk of unauthorized access.

Professionalism: Using a dedicated email creates a more professional image for the project when working with cloud service providers, external partners, or other stakeholders.

2. Use GitHub for Version Control and Collaboration with Team Members

Objective: Implement GitHub or another version control system to facilitate team collaboration and track project changes.

Version Control: GitHub allows you to track every change made to the project, which helps maintain a clear history and enables you to revert to previous versions if needed.

Team Collaboration: Multiple developers can work on the project simultaneously without overwriting each other's changes. It also enables easy collaboration on different branches and ensures that everyone is working on the latest version.

Code Reviews: Team members can use GitHub’s pull request feature to review each other's code before it’s merged into the main branch, ensuring high-quality code and reducing errors.

Continuous Integration (CI): Integrate CI tools like GitHub Actions to automatically test and deploy your project whenever changes are made.

3. This is a Prototype Version – Strive to Create the Best Final Version

Objective: Keep improving the project and focus on making the final version polished and professional.

Prototype Focus: While the current version serves as a prototype to demonstrate core functionality, always aim to refine the user interface, optimize performance, and enhance security in future iterations.

User Feedback: Gather feedback from potential users, whether they are students, teachers, or staff, and implement their suggestions to improve usability and functionality.

Polish and Refine: Invest time and effort into improving the design, user experience (UX), and performance of the app. Pay attention to details such as responsive design, speed optimization, and ensuring that the project runs smoothly on all devices.

Testing: Conduct thorough testing (unit testing, user acceptance testing, etc.) to identify and resolve any bugs or issues. This will ensure the app's stability and smooth operation in the real world.