Nearby ATM Locator
Integrating OpenStreetMap API for an Interactive Nearby ATM Locator

Catequista, Glenn Marco A.
Magtibay, John Christian Q.
Osia, Mark F.

Integrative Programming and Technologies 1 (IPT01)
Vilog, Jericho
ICCT Colleges
February 26, 2025

Integrating OpenStreetMap API for an Interactive Nearby ATM Locator

I. Application Design

- **System Components**
  1. Built using **HTML, CSS, and JavaScript**.
  2. Uses **Leaflet.js** for interactive mapping.
  3. Implements a clean and modern UI for better user experience.
- **Main UI Elements**
  1. **Search Bar:** Allows users to input a location or landmark to find nearby ATMs.
  2. **Map Interface:** Displays the OpenStreetMap map with markers for the user's location and nearby ATMs.
  3. **Geolocation Button:** Enables users to find ATMs near their current location.
  4. **Bank Filter Dropdown:** Allows users to filter ATMs based on a specific bank.
  5. **ATM Information Popups:** Displays details such as the ATM name, bank, and a "Get Directions" button.
  6. **Route Display:** Shows a blue navigation path with estimated distance and time when directions are requested.
  7. **Error Messages:** Provides real-time feedback when an error occurs, such as "No ATMs found" or "Geolocation not allowed.

II. API Integration

- **OpenStreetMap (OSM) & Overpass API:** Retrieves ATM locations within a specified radius.
- **Nominatim API:** Converts location queries into geographic coordinates.
- **OSRM Routing API:** Generates walking routes and calculates estimated travel time.

III. User Interaction

1. Accessing the Application
   - Users open the ATM Locator system in a web browser.
   - The system loads an interactive OpenStreetMap interface.

2. Enabling Geolocation *(Optional)*
    - Users can allow the browser to access their location.
    - If geolocation is enabled, the system retrieves the user's coordinates and centers the map on their location.
    - If denied, an error message ("**Geolocation not allowed.**") is displayed, and users can manually search for a location.
3. Searching for a Specific Location or Landmark
    - Users enter a city, landmark, or address in the search bar.
    - The system uses the **Nominatim API** to find the location.
    - If a valid location is found, the map updates and searches for nearby ATMs.
    - If no results are found, an error message ("**Location not found.**") is displayed.
4. Viewing Nearby ATMs
    - The system automatically fetches nearby ATMs within a 1km radius using the **Overpass API**.
    - ATM locations are marked on the map.
    - Users can click on an ATM marker to view its details, including:
        - **ATM Name** (if available)
        - **Bank Name**
        - **"Get Directions"** button for navigating
5. Filtering ATMs by Bank
    - Users can select a specific bank from the dropdown filter.
    - The system updates the ATM markers to show only ATMs from the selected bank.
    - If no ATMs from the selected bank are found, an error message ("**No ATMs found for [selected bank].**") is displayed.
6. Getting Directions to an ATM
    - Users click the **"Get Directions"** button on an ATM popup.
    - The system fetches a walking route using the **OSRM Routing API**.
    - A **blue path** appears on the map, showing the shortest route.
    - The popup displays estimated **distance (km) and travel time (minutes)**.
    - If no route is found, an error message (**"No route found."**) is displayed.
7. Closing Directions
    - Users can click **"Close Directions"** to remove the displayed route.

- The route disappears from the map, and the estimated distance and time information is cleared.

8. Handling Errors and Notifications
    - If geolocation is unsupported or denied, a message is displayed.
    - If no ATMs are found in the selected area, users receive a notification.
    - If the search API request fails, an error message is displayed.

IV. Challenges

The challenge we faced during development was making the marker disappear when the close button (x). We couldn't do that, so we made an alternative to that problem; we put a button to make the marker disappear.

When we want to make a placeholder for the map, the placeholder is working, but the distance marker to the ATM isn't working. We tried other techniques for this one, but they all lead to that problem, so we tried not to include JavaScript for the placeholder, only properties from HTML and CSS, and somehow we tried to fix it again and it works now.

Overall, it was a good experience for us to make the program, even though it was hard to make it step by step, and it made us more collaborative by making a group vote to make a decision for every problem we encountered. We will explore more soon to make this little website perfect.