Web Development

Team - 7

- CS21B022 HARSH MAWAR
- CS21B023 JATOTHU SRINIVAS NAYAK
- CS21B025 K E NANDA KISHORE
- CS21B038 NIRAJ KUMAR
- CS21B061 SHAFI UR RAHMAN KHAN

Problem Statement: Design and develop a web application that allows users to explore a 3D map with markers indicating monuments and artifacts, as well as the ability to view 3D models of these landmarks.

Primary Objectives:

1. Display a 3D map using Google Maps API, with the ability to zoom and pan.
2. Add 3D models to the map to represent various monuments and artifacts.
3. Implement a user-friendly interface to interact with the map and markers.
4. Make the 3D model react with the environment w.r.t. Tilt and pan.

Design Decisions and Technologies Used:

1. **MERN Stack:** Utilize the MERN stack for the web application development:
    - React: Develop the frontend user interface and interactive map components.
    - MongoDB, Express.js and Node.js is not being used currently since the main objective of ours during release 1 was to handle the smooth integration of 3D models and a 3D map to out website. They will be used later to fetch data about the various artifacts.

2. **Google Maps API Integration:** Utilize Google Maps JavaScript API for rendering the 3D map and adding markers.

3. **3D Modeling Libraries:** Use Three.js to display 3D models of monuments and artifacts. These libraries provide tools for rendering 3D graphics within a web browser.

4. **UI/UX Design:** Design a user-friendly and responsive UI using React and CSS frameworks like Bootstrap or Chakra UI. Focus on intuitive map navigation and marker interaction.

5. **Documentation and User Support:** Provide comprehensive documentation for developers and users, including installation instructions, API documentation, and user guides. Offer user support and feedback channels.

6. **Vector Maps** instead of Street View : We initially planned to use Street View for the project, but we later realised that street view is essentially just a panoramic 360 degree 2D photo and is not fit for a truly 3D experience. Vector map on the other hand is a completely 3D environment implemented with WebGL which we can directly interact with using the WebGLOverlay, and achieve proper depth occlusion and tilting support.

How to Setup:

To host our website locally, follow these steps:

1. Clone our GitHub repository to your local machine.
2. cd into the nex directory.
3. Run `npm install` to install all dependencies.
4. make a .env file and copy the contents of .env.example into it.
5. Replace the value of the REACT_APP_GOOGLE_MAPS_API_KEY variable with your own Google Maps API key.
   (Note: You will need a Google maps platform account with a billing account to get an API key, and then enable the Maps JavaScript API.
   For more info visit : https://developers.google.com/maps/documentation/javascript/cloud-setup)
6. Replace the REACT_APP_GOOGLE_MAPS_ID variable with your own Google Maps ID.
   (Note: Make sure to use a vector map with tilt and rotation enabled.
   For more info visit : https://developers.google.com/maps/documentation/javascript/webgl)
7. Run `npm start` to start the server.

How to Use:

Using our website is simple and intuitive:

1. Navigate to the website URL.
2. Use your mouse or touchpad to pan around the map.
3. Use the zoom in/out buttons to get a closer look or a wider view.
4. Hold the shift button and use mouse or arrow buttons to change the viewing angle.

GitHub Repository: **HTTPS:** https://github.com/niraj-kumar-r/NexArtifacts.git
Github CLI: gh repo clone niraj-kumar-r/NexArtifacts

Contributions:

- **CS21B022**: Searched for a 3D model,designed logo for website,designed a user-friendly UI and documented the process in a README file.
- **CS21B023**: Set up the website and created a UI layout.
- **CS21B025**: Implemented dynamic resizing of the 3D object based on map zoom.
- **CS21B038**: Integrated Google Maps API, rendered a 3D model with accurate depth occlusion and tilt.
- **CS21B061**: Created a Dockerfile to containerize the application & made project documentation.