Nile University
جامعة النيل

Car Reviewer

CSCI452

Team Members:
Seifeldin Khaled Ibrahim 211001832
Omar Mohammed Tawfik 2110003001
Alaa Hassan Hassan Ibrahim 19105762

Professor: Ahmed F. Elnokrashy

Fall-2024

Project Description and Requirements

1.a. Project Description and Objective

Project Name: Car Reviewer

Description:
The **Car Reviewer** project is an interactive 3D web application designed to enhance the online car shopping experience. Users can explore car models virtually, reducing the need for in-person showroom visits. This is achieved using modern web technologies, including **HTML**, **CSS**, **JavaScript**, and the **Three.js** library, which enables 3D rendering directly in a browser.

- The application features:

- A user-friendly homepage with navigation, a welcoming introduction, and a contact section.
- A 3D viewer where users can rotate, zoom, and explore car models in detail.

Objective:
The primary goal is to provide users with a visually rich and interactive car-shopping tool that:

- Offers detailed 3D visualizations of car models.
- Improves the user experience of online car browsing.
- Demonstrates the capabilities of web technologies like Three.js for creating dynamic 3D environments.

Target Audience:

- Car enthusiasts exploring new models.
- Potential buyers seeking a virtual way to assess vehicles.
- Web developers looking for a case study on implementing 3D web experiences.

1.b. Sequence of Steps to Accomplish the Project

1. **Define Project Requirements**:
    - Identify the primary features needed for the application (e.g., 3D model viewing, user interface).
    - Gather resources, including 3D car models in .gltf format.
2. **Design the Homepage**:
    - Develop the homepage structure using **HTML** and **CSS**.
    - Include navigation, a welcome message, and a contact section.
3. **Integrate the 3D Viewer**:
    - Set up the environment using the **Three.js** library.
    - Configure the scene, camera, renderer, and lighting to simulate a showroom-like atmosphere.
    - Add controls for rotating, zooming, and panning the 3D model.
4. **Test and Debug**:
    - Test the application across multiple devices and browsers to ensure compatibility.
    - Debug any issues with rendering, user interactions, or layout responsiveness.
5. **Optimize for Performance**:
    - Compress the 3D models and minimize CSS and JavaScript files.
    - Implement lazy loading for resources to improve initial load times.
6. **Deploy the Application**:
    - Host the project files on a local or online server (e.g., GitHub Pages, Netlify).
    - Verify deployment by testing functionality and layout on the live site.
7. **Prepare Documentation**:
    - Write detailed instructions for setting up and running the project.
    - Include file descriptions and any required dependencies.

Implementation

2.a. Code Structure and Organization

The project code is structured for clarity and maintainability, as follows:

- **HTML Files**:
  - index.html: Contains the homepage structure and links to required styles and scripts.
  - 3d-viewer.html: Implements the 3D car viewer functionality.
- **CSS Files**:
  - style.css: Custom styles for the homepage.
  - bootstrap.min.css: For responsive and standardized design.
- **JavaScript**:
  - Inline script in 3d-viewer.html to handle Three.js setup and interactivity.
- **Assets**:
  - scene.gltf: 3D model file for rendering.
- **Documentation**:
  - README.md: Comprehensive setup and usage guide.

This modular approach makes it easy to maintain and extend the project.

2.b. Creativity and Design

The project demonstrates creativity through:

- **User Interface Design**:
  The homepage features a modern and clean design, with well-organized sections for navigation, welcome text, and a contact form. Interactive buttons guide users to explore car models, enhancing the user experience.
- **3D Viewer Aesthetics**:
  The virtual showroom effect is created by combining ambient lighting, directional lighting, and point lights. These simulate a professional studio environment for showcasing cars.
- **Interactive Elements**:
  Users can seamlessly rotate, zoom, and pan the 3D car model, making the experience immersive and intuitive.

2.c. Comprehensive Usage of Three.js

The project uses Three.js extensively to implement advanced features, such as:

1. **Scene Setup**:
    - The scene is configured with a neutral background color (`0xdddddd`) to mimic a clean showroom environment.
2. **Camera Configuration**:
    - A perspective camera is placed strategically to provide a natural view of the car.
3. **Lighting**:
    - **Ambient Light**: Provides overall brightness to the scene.
    - **Directional Light**: Highlights the model, simulating sunlight or spotlights.
    - **Point Lights**: Add depth and realism by casting soft glows from multiple directions.
4. **Object Loading**:
    - The **GLTFLoader** module is used to import and render the car model (`scene.gltf`).
    - The model is scaled and positioned appropriately for optimal viewing.
5. **Interactive Controls**:
    - **OrbitControls** enable users to rotate, zoom, and pan the model, providing a fully interactive experience.
6. **Responsive Design**:
    - The renderer dynamically adjusts its size when the window is resized, ensuring the viewer works seamlessly across devices.