**Project Report**

**Car Rental Website**

Batoul Ballout, Rasha Harb, Razan Doughman

1. **Introduction**
   1. **Project Overview**

The website developed allows users to rent cars online. It provides features for browsing available cars, making reservations, paying, and managing user accounts. Key functionalities include user authentication, car reservation management, and administrative tools for data management.

* 1. **Project Goals and Objectives**

The main goal was to develop a fully functional car rental website with a user-friendly interface and robust backend system.

Objectives include:

- Implementing a user-friendly car reservation system.

- Providing secure user authentication.

- Enabling administrators to manage cars, users, and reservations efficiently.

- Displaying car information and specifications effectively.

- Generating invoices for reservations.

- Allowing users to provide feedback and chat with agents.

These objectives directly address the requirements outlined in the project specsifications description.

* 1. **Scope of the Project**

The project includes the development of:

- **Frontend:** User interface for browsing cars, making reservations, user authentication, and displaying information.

- **Backend:** Server-side logic for handling user requests, managing data, and processing transactions.

- **Database:** Storage for user information, car details, reservation data, and other relevant information.

1. **System Analysis and Design**
   1. **Functional Requirements**

**- Frontend:**

* User Authentication: Users can sign up, log in, and log out.
* Car Reservation: Users can search for cars, select pickup and drop-off locations from our 3 branches, specify reservation dates and times.
* Car Browsing and Filtering: Users can browse cars by group and filter cars based on specifications (e.g., number of seats, fuel type).
* Additional Services: Users can select additional services such as chauffeur, baby seat, insurance options, and fuel options.
* Reservation Management: Users can view, save, and finalize reservations.
* Payment Processing: Users can choose to pay for reservations online.
* Coupon Management: Users can apply discount coupons.
* Review System: Users can view and submit reviews.
* Chat with Agent: Users can communicate with a customer service agent.
* About Us and Contact Us: The website includes pages with company information and a contact form.
* Homepage Statistics: The homepage displays the most popular car and the average daily rental fee.
* User Points and Discounts: Users earn points for rentals, which can be used for discounts.

**- Backend:**

* User Management: Admins can create, read, update, and delete user accounts.
* Car and Car Group Management: Admins can manage car details and car groups.
* Reservation Order Management: Admins can view and update reservation orders.
* Invoice Generation: The system generates invoices for reservations.
* Statistical Information Processing: The backend calculates and provides statistical information for the frontend.
* API Endpoints: The backend provides API endpoints for the frontend to access data and functionality.
  1. **Non-Functional Requirements**
* Responsiveness: The website is designed to be responsive and accessible on various devices (desktops, tablets, and mobile phones).
* Security: User authentication is implemented to protect sensitive data and restrict access to authorized users.
* Performance: The website is optimized for fast loading times and efficient handling of user requests.
* Usability: The user interface is designed to be intuitive and easy to navigate, providing a positive user experience.
  1. **System Architecture**

The system follows a three-tier architecture:

- Frontend (Presentation Tier): Handles user interaction and displays information.

- Backend (Application Tier): Processes user requests, manages data, and enforces business logic.

- Database (Data Tier): Stores and retrieves data.

Technologies Used:

- **Frontend**: HTML, CSS, JavaScript

- **Backend:** NodeJS, ExpressJS

- **Database:** MongoDB

1. **Key Features Implementation for Backend:**

* Authentication: User authentication is implemented using JWT. When a user registers or logs in, the server generates a JWT that is sent to the client. The client includes this token in subsequent requests to authenticate the user. The `authMiddleware.js` verifies the token.
* API Endpoints:

- `/users`: Handles user-related operations

- `/cars`: Handles car-related operations

- `/reservations`: Handles reservation-related operations

- `/invoices`: Handles invoice generation and retrieval.

* Database Interaction: Mongoose is used as an ODM (Object Data Modeling) library to interact with the MongoDB database. Models define the structure of the data, and Mongoose provides methods for querying, creating, updating, and deleting documents.
* File Uploads: The `upload.js` middleware uses Multer to handle file uploads.
* Middleware:

- `authMiddleware.js`: Protects routes that require authentication.

- `errorHandler.js`: Handles errors and sends appropriate error responses to the client.

1. **Key Features Implementation for Frontend:**

* UI Design: The website has a clean and modern design with a focus on usability. CSS is used for styling and layout, and responsiveness is achieved using media queries.
* Responsiveness: CSS media queries are used to ensure the website adapts to different screen sizes. The layout adjusts dynamically to provide an optimal viewing experience on desktops, tablets, and mobile phones.
* User Interactions: Users interact with the website through form submissions (e.g., login, registration, reservation forms), button clicks (e.g. proceeding to checkout), and navigation links. JavaScript handles these interactions and updates the UI accordingly.
* Data Display Data from the backend is fetched using API requests and displayed on the page.
* Google Maps Integration: embedded Google Maps for branches.

1. **Team Contributions:**

* **Batoul Ballout:** worked on the backend, connections and helped in frontend details.
* **Rasha Harb:** worked on the frontend and UI design, and helped in backend details.
* **Razan Doughman:** worked on the database, backend, and helped in frontend details.

**Demo of the project found in the link below:**<https://drive.google.com/file/d/1H5uRC4y9PGp3G4RPvofFXg2JmxH4Nml8/view?usp=sharing>