

Job Task

Dear Candidate, We are pleased to inform you that you have successfully passed the first round of the selection process! Your application and skills have impressed us, and we are excited to move forward with you in the next stages. This project is designed to assess your skills, creativity, and problem-solving abilities. It will help us understand how you approach challenges and your ability to deliver high-quality solutions.

Project Overview

We are seeking a proficient Car Rental System Developer to lead the creation of a feature-rich, user-centric car rental platform. Your role involves designing and implementing responsive frontend and robust backend systems. The platform will enable seamless car bookings, user authentication, and efficient management of car inventory. Proficiency in frontend technologies like HTML, CSS, JavaScript, React.js, and backend frameworks like Node.js, Express.js, and MongoDB is required. Familiarity with Firebase Authentication, JWT implementation, and advanced filtering techniques will be beneficial. Join us to innovate car rentals through technology and user experience!

Updates: 0 [Any updates will be mentioned here]

Let's Understand the Website Theme

You have to create a Car Rental System. On this platform:

- Users can **Add, Update, and Delete Cars** listed for rental.
 - Users can **Search and Book Cars** based on their availability.
 - Users can **Review Cars** they have rented.
 - The system provides real-time updates on car availability and booking statuses.
-

Ensure the Following for Full Marks

- Include **at least 15 meaningful commits** on the client side and **8 meaningful commits** on the server side with descriptive messages.
 - Include a **README.md** file with the project name, purpose, live URL, key features, and npm packages used.
 - Make the website **fully responsive** on mobile, tablet, and desktop.
 - Secure Firebase configuration keys and MongoDB credentials using **environment variables**.
 - Design with an eye-pleasing color contrast and proper alignment. Avoid cluttered designs.
 - Deploy properly; no CORS/404/504 errors should appear.
 - Ensure private routes do not redirect logged-in users to the login page on reload.
-

Deployment Guidelines

- Ensure the server works perfectly on production without errors.
 - Your live link should function properly, with no errors on landing or navigation.
-

Layout and Page Structure

Navigation Bar

- A beautiful header with the website name + logo.
- Conditional rendering for navigation menus:
 - Non-logged-in users see: Home | Available Cars | Log-in
 - Logged-in users see: Home | Available Cars | Add Car | My Cars | My Bookings | Logout

Footer

- Include a website logo, name, copyright information, and social media links.

Error Page

- A 404 page with an image or gif and a "Back to Home" button that redirects the user to the homepage.

Pages & Features

Home Page Structure

1. Banner Section

- **Content:**
 - **Motivational Heading:** Display an inspiring heading such as *"Drive Your Dreams Today!"* or *"Your Next Car Awaits You."*
 - **Call-to-Action (Button):** A prominently styled button, e.g., *"View Available Cars"*, redirects users to the "Available Cars" page.
 - **Design:**
 - Full-width background image or video showcasing premium or exotic cars.
 - Layer the content (heading and button) over the background with a slight shadow for readability.
-

2. Why Choose Us?

- **Purpose:** Highlight your platform's unique selling points to build trust and draw user interest.
 - **Content:**

Use icons or visuals to represent 3-4 points such as:

 1. **Wide Variety of Cars:** *From budget-friendly options to luxury vehicles.*
 2. **Affordable Prices:** *Competitive daily rates you can count on.*
 3. **Easy Booking Process:** *Seamlessly book your ride in just a few clicks.*
 4. **Customer Support:** *24/7 assistance for all your queries.*
 - **Design:**

A grid layout with an icon and a short description for each point.
-

3. Recent Listings

- **Content:** Showcase 6-8 of the latest cars added to your platform using a **card layout**. Each card should include:
 - **Car Image:** A high-quality thumbnail or photo of the car.
 - **Model:** The car model, e.g., *Toyota Camry 2023*.
 - **Daily Price:** The rental price per day, e.g., *\$45/day*.
 - **Availability:** A badge or text indicating if the car is available for immediate booking.
 - **Date Posted:** *e.g., Added 2 days ago*.
 - **Design:**
 - Cards arranged in a **grid layout (3 per row)**.
 - Add a hover effect: Zoom in slightly or display a shadow around the card.
-

4. Extra Sections (With Animations):

A. User Testimonials

- **Purpose:** Showcase customer reviews to build credibility.
- **Content:**
 - Add 3-5 user reviews displayed in a carousel/slider format.
 - Each review can include:
 - User Name
 - Profile Image
 - Rating (Stars)
 - A short review text (e.g., *"Amazing service and great cars. Highly recommend!"*).
- **Animations:**
 - Fade-in or slide-in effect for each testimonial as the carousel changes.

B. Special Offers

- **Purpose:** Highlight discounts or deals to attract users.
- **Content:**
 - Create banners/cards featuring promotions, e.g.:
 - *"Get 15% off for weekend rentals!"*
 - *"Luxury cars at \$99/day this holiday season!"*
 - Include a "Learn More" or "Book Now" button.
- **Animations:**
 - Offer cards slide in from the side or bounce slightly on hover.

Add Car Page (Private)

- A private route where authenticated users can add cars with fields:
 - Car Model
 - Daily Rental Price
 - Availability
 - Vehicle Registration Number
 - Features (e.g., GPS, AC, etc.)
 - Description
 - bookingCount(default 0)
 - Images (use `react-dropzone` for file uploads)
 - Location
- Save user details, date, and default booking status into the database.

My Cars Page (Private)

- A private route where authenticated users can manage their added cars.
- Display all cars added by the user in a **tabular format**, with columns for:
 - Car Image
 - Car Model
 - Daily Rental Price
 - Availability
 - Date Added
- Include the following actions in each table row:
 - **Update Button:**
 - Triggers a modal with the following editable fields:
 - Car Model
 - Daily Rental Price
 - Availability
 - Vehicle Registration Number
 - Features (e.g., GPS, AC, etc.)
 - Description
 - Images (option to upload new images using `react-dropzone`)
 - Location
 - On form submission, validate the input and save changes to the database.
 - Notify the user of successful updates via a modal or toast notification.

- **Delete Button:**
 - Prompts a confirmation modal before proceeding.
 - Upon confirmation, remove the car from the database.
 - Updates the table in real-time and notifies the user of the deletion.
- Add **Sorting** options(from backend or frontend):
 - **Sorting:**
 - Enable users to sort cars by:
 - Date Added (Newest First / Oldest First)
 - Price (Lowest First / Highest First)
- If no cars are added, display a message prompting the user to add a car with a link to the "Add Car" page.

Available Cars Page

- Display all available cars in a grid or card format.
- Add search functionality based on the car model, brand, or location.
- Add a toggle button for switching between grid and list views.
- Cards should show essential details with a "Book Now" button leading to the car details page.
- Include **Sorting** options as mentioned above for user convenience.

Car Details Page

- Detailed car information:
 - Model
 - Price Per Day
 - Availability
 - Features
 - Images
 - Description
 - Reviews
- "Book Now" button opens a confirmation modal summarizing booking details.

Login and Registration System

- **Login Page:**
 - Allow users to log in with Email/Password or via Google authentication using Firebase.
 - Redirect users to the homepage upon successful login.
 - Display relevant error messages when authentication fails.
 - Provide a link to redirect users to the Registration page.
- **Registration Page:**
 - Allow users to register with fields:
 - Name
 - Email
 - Password
 - Photo URL
 - Upon successful registration, redirect users to the login page.
 - Validate inputs and display appropriate error messages.
- **Social Login:**
 - Enable Google Sign-In functionality for easy access.
- **Private Route Access:**
 - Protect pages like "Add Car," "My Cars," "My Bookings," and "Manage Cars" by ensuring only authenticated users can access them.

My Bookings (Private)

This section will provide users with a detailed view of all the bookings they have made, offering functionality to **view, modify, or cancel bookings** in a user-friendly format.

Core Features

1. **Tabular Layout:**
 - A responsive and scrollable table displaying all bookings in a clean format.
 - **Columns Included:**
 1. **Car Image:** A small thumbnail of the booked car for easy identification.
 2. **Car Model:** The car's make and model, e.g., *Toyota Corolla 2023*.
 3. **Booking Date:** The date and time the car was reserved (formatted as *DD-MM-YYYY HH:MM*).
 4. **Total Price:** The cost of the entire booking period, including taxes/fees.

5. **Booking Status:** Clearly display whether the booking is *confirmed*, *pending*, or *canceled*.
 6. **Actions:** Buttons to modify the booking date or cancel the booking.
-

Design and Interactivity

- **Table Design:**
 - Styled rows with alternating background colors for better readability.
 - Column headings (Car Image, Car Model, etc.) should use bold text and a subtle background color.
 - Rows to highlight (e.g., with a light shadow or color) on hover for interactivity.
 - **Buttons:**
 - **Cancel Booking:**
 - A red button with a trash can icon and text *"Cancel"*.
 - On click, display a **confirmation modal** asking:
 - *"Are you sure you want to cancel this booking?"*
 - Include **Yes/No** options, with *Yes* executing the cancellation and *No* closing the modal.
 - Update the row to reflect the new status (e.g., *Canceled*).
 - **Modify Booking Date:**
 - A blue button with a calendar icon and text *"Modify Date"*.
 - On click, open a **modal with a date picker** allowing the user to select a new start and end date.
 - Include options to **Confirm** or **Cancel** the action.
 - Once confirmed, update the table with the modified booking details and save the changes.
-

Challenges

- **Search Functionality:** Add search capabilities based on car model or location.
- **JWT Authentication Requirement:**
- Implement JWT-based authentication to secure private routes and verify users. Store the generated token in an HTTP-only cookie after successful login and remove it during logout. Include middleware to validate tokens for access to protected APIs, ensuring session integrity.

- **Incremental Booking Updates:** Use MongoDB operators like `$inc` to track booking counts.
 - **Layout Toggles:** Add buttons for toggling the layout style (grid/list) on the "Available Cars" page.
 - **Data Visualization:** Use `chart.js` or `recharts` to visualize booking trends or revenue data.
-

Submission Checklist

- Live Site URL
- Client-side GitHub Repository
- Server-side GitHub Repository

Optional Requirement

1. On the "My Cars" page, implement server-side pagination to display five cars per page. Incorporate a search input field that dynamically queries the backend, allowing users to search across all car data fields. The search functionality should match the entered text with any car-related information for accurate and comprehensive results.