

"Marketplace Technical Foundation – [Car-Rental Website]"

Frontend Requirements :

Pages Overview :

Login:

The **Login** page will allow users to log in using their email/password or other available methods (Google, Facebook, etc.).

Once a user is logged in, they will remain logged in until they sign out.

SignUp :

The **SignUp** page will allow users to create an account by providing their email and password.

Secure Routes: Only authenticated users will be allowed to access protected routes (e.g., Cart, Checkout, Profile).

Navbar:

- *. Website logo.
- *. Search bar .
- *. Login (button).
- *. SignIn (button).
- *. User-friendly navigation menu.

Home Page:

- *. Best rental Cars.
- *. rental Car(button).

- *. pickup & drop-off section.

Popular Cars Page:

- *. Detailed car information (model, fuel type, seating capacity)
- *. Images.
- *. Pricing (current, original).
- *. Favourite (button).

Categories Page:

- *. Car (name, type, Price, discount)
- *. Detail (fuel, manual, seats)
- *. Button (Rent now)
- *. Responsive grid design for displaying cars.

Cart Page:

- *. List of selected cars with rental dates.
- *. Summary of rental charges.
- *. Option to add/remove cars.

Checkout Page:

- *. Rental info (location, date, time).
- *. Payment options (credit card, PayPal, bitcoin).

- *. Billing details.
- *. confirmation (privacy policy).
- *. Rent Now (button).

Tracking Page:

- *. Booking status ("Confirmed").
- *. Real-time updates on car availability and pickup location.
- *. Integration with a map to show car location or rental office.

Footer :

Include critical business information such as:

- *. About Us
- *. How It Works
- *. Featured Products
- *. Partnership
- *. Privacy Policy
- *. Terms & Conditions
- *. Contact Information

Social Media Links:

- *. Discord
- *. Instagram
- *. Facebook
- *. Twitter

Sanity Schema Example:

Product (Car) Schemas :

```
export default {
  name: 'car',
  title: 'Car',
  type: 'document',
  fields: [
    {
      name: 'type',
      title: 'Car Type',
      type: 'string',
    },
    {
      name: 'model',
      title: 'Car Model',
      type: 'string',
    },
    {
      name: 'year',
      title: 'Year',
      type: 'number',
    },
    {
      name: 'fuelType',
      title: 'Fuel Type',
      type: 'string',
    }
  ]
}
```

```
options: {  
  list: ['Petrol', 'Diesel', 'Electric', 'Hybrid'],  
},  
{  
  name: 'seatingCapacity',  
  title: 'Seating Capacity',  
  type: 'number',  
},  
{  
  name: 'currentPrice',  
  title: 'current Price',  
  type: 'number',  
},  
{  
  name: 'originalPrice',  
  title: 'original Price',  
  type: 'number',  
},  
{  
  name: 'carImages',  
  title: 'Car Images',  
  type: 'array',  
  of: [{ type: 'image' }],  
},  
{  
  name: 'manual',
```

```
    title: 'Manual',  
    type: 'string',  
  },  
],  
};
```

Customer Schema :

```
export default {  
  name: 'customer',  
  title: 'Customer',  
  type: 'document',  
  fields: [  
    {  
      name: 'firstName',  
      title: 'First Name',  
      type: 'string',  
    },  
    {  
      name: 'lastName',  
      title: 'Last Name',  
      type: 'string',  
    },  
    {  
      name: 'email',  
      title: 'Email',  
      type: 'string',  
    },  
  ],  
};
```

```

{
  name: 'phone',
  title: 'Phone Number',
  type: 'string',
},

{
  name: 'bookedCars',
  title: 'Booked Cars',
  type: 'array',
  of: [{ type: 'reference', to: [{ type: 'car' }] }],
},
],
};

```

Order Schema:

```

export default {
  name: 'order',
  title: 'Order',
  type: 'document',
  fields: [
    {
      name: 'orderDate',
      title: 'Order Date',
      type: 'datetime',
    },
    {

```

```
name: 'pickupDate',
title: 'Pickup Date',
type: 'datetime',
},
{
name: 'dropoffDate',
title: 'Drop-off Date',
type: 'datetime',
},
{
name: 'totalPrice',
title: 'Total Price',
type: 'number',
},
{
name: 'paymentStatus',
title: 'Payment Status',
type: 'string',
options: {
list: ['Pending', 'Paid', 'Cancelled'],
},
},
{
name: 'customer',
title: 'Customer',
type: 'reference',
to: [{ type: 'customer' }],
```



```
},  
{  
  name: 'car',  
  title: 'Car',  
  type: 'reference',  
  to: [{ type: 'car' }],  
},  
],  
};
```

API Requirements

Endpoint	Method	
/api/products	GET	Fetch product data from Sanity CMS.
/api/shippinglabel	POST	Generate a shipping label using ShipEngine.
/api/trackingorder	GET	Retrieve order status using ShipEngine label ID.
/api/checkout	POST	Integrate Stripe for payment processing.

Tools & Libraries

- Clerk: Authentication.
- Sanity CMS: Content management.
- ShipEngine API: Shipping and tracking.
- React Context API: Cart functionality.
- Delivery/Tracking