Assignment12_category_016

📽️ Requirement Explaintion Video 📽️

Objective

Our company is looking for a highly skilled and dynamic Full Stack Developer to scratch a **Parcel Management System with MERN stack.** As a team member, you must write clean, reusable, and scalable react components.

You must develop a **Parcel Management App (an online platform where people can Book Parcel To Deliver, Admin Can Assign Delivery Person, and Delivery Person Delivers the Parcel)** using the MERN stack.

Main Tasks

Key Rules:

- Include a minimum of 20 notable GitHub commits on the client side
- Include a minimum of 12 notable GitHub commits on the server side
- Add a meaningful readme.md file with the name of your website, Admin username, password and live site URL. Include a minimum of 10 bullet points to feature your website.
- Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views. Make the dashboard responsive as well.
- After reloading the page of a private route, the user should not be redirected to the login page.

- Use the Environment variable to hide the Firebase config keys and Mongodb credentials.
- Don't use any Lorem ipsum text in your website.
- Show sweet alert/toast/notification for all the crud operations, successful authentication login, and sign-up. Don't use the default browser alert.
- Implement tanstack query in all the data fetching functionality (For GET method only)

HomePage:

1. **Navbar** has a **logo + website name, Home, Notification icon,** and **Login (when not logged in)** button. If the user is logged in, his/her profile picture should appear on the navbar.

   If the user clicks on the **profile picture,** a drop-down will appear with the following items: **User name (not clickable), Dashboard,** and **Logout** button.

   Use Cool animations throughout the application. Make a stunning footer for the app.

2. **The Banner** section will have a background image, a search bar, and heading text.
   Note: Do the necessary beautification of this section.

3. **Our Features Section** will show 3 features (in cards) of your Delivery system to attract users to use your app. I.e. Parcel Safety, Super Fast Delivery, etc.

   Each Feature Card Will Have

   ★ An Icon
   ★ Feature Title
   ★ A short description

Below The cards, you will show statistics of your app usage in 3 cards. The Numbers will come from the database.

a. Total Number of Parcels Booked
b. Total Number of Parcels Delivered
c. Total Number of People Using Your App (Number of registered users)

Note: Use the react-countup package to animate the numbers.

4. **The Top Delivery Man** Section will show the top 3 delivery men in 3 cards. You will sort them by the number of parcels they delivered and an average of their ratings. Each Card will have

   ★ Delivery Man's Name
   ★ Image
   ★ Number of parcels Delivered (will come from database)
   ★ Average Ratings (calculated from database review)

5. Make a **Login and Registration** system for the users to create accounts and login to private routes in your application. Implement at least one social login system. After successful registration, save the user info in the MongoDB collection.

   There will be 3 user types: User, DeliveryMen, Admin

   No One can Register as Admin, You have to manually insert one admin user in the database. Register With one email and from the MongoDB collection, edit the type to "Admin"

   You must take **user type input** by a select field when someone registers in the application. Also for email/password-based login take input for name, and profile image. Admin can make a user "DeliveryMen" or "Admin" from the admin dashboard.

   NOTE: While social login the user type will always be **'User'.**

6. **The Dashboard**: Clicking the Dashboard Menu in the navbar user will be redirected to the dashboard page. All The pages and Routes in the dashboard are **Private Routes.** You will show pages and menus in the dashboard Conditionally, according to the user type.

   There will be a sidebar on the dashboard page that will show the following menus according to the user type.

   - **Users Will See** - Book a Parcel, My Parcels, My Profile menu in the sidebar.

   - **Delivery Men Will See** - My Delivery List, and My Reviews menu in the sidebar.

   - **Admin Will See** - All Parcels, All Users, All Delivery Men, Statistics, menu in the sidebar.

7. **The User Dashboard:** If the logged-in user is a normal user he/she can navigate to the dashboard page and see the menus in the sidebar.

- **Book a Parcel:** This page will let the user book a parcel by filling out the following form.

  - Name(Auto filled from the logged-in user, read-only)
  - Email(Auto filled from the logged-in user, read-only)
  - Phone Number
  - Parcel Type(Text Field)
  - Parcel Weight (Number Field)
  - Receiver's Name
  - Receiver's Phone Number
  - Parcel Delivery Address
  - Requested Delivery Date(Date Input)
  - Delivery Address Latitude (i.e 21.121365496)
  - Delivery Address longitude (i.e 21.121365496)

- Price(Auto Calculated from the Parcel Weight Input. For 1 kg Price is 50Tk, for 2 kg 100Tk, more than 2kg price will be 150Tk)
- Book Button(submit)

Note: Make sure to validate the form to get the appropriate data type. Clicking on the **Book Button** the data will be stored in MongoDB collection. When you save the booking, by default it will have "**pending**" status.

❖ **My Parcels:** This page will show all the parcels booked by the logged-in user in a tabular form. Each row will have

- Parcel Type,
- Requested Delivery Date,
- Approximate Delivery Date,
- Booking Date(Auto Generated by **new Date()**)
- Delivery Men ID(this property will be updated when Admin Assigns a Delivery Men),
- Booking Status(There will be pending(default), on the way, delivered, returned, canceled statuses),
- Update and Cancel Button.
- **Review Button**- If the parcel status is delivered, show a review button. (See Bonus Point 2 For The Additional Guideline)
- **Pay Button**(See Bonus Point 3 For The Additional Guideline)

Update- Clicking on this button will redirect the user to the update booking page and let him update the booking. Make a note that the user can update the booking only if the booking status is 'pending'. Otherwise, you'll disable the update button.

Cancel- You will update the status to cancel if the user clicks on the cancel button, and shows an alert before you change the status. Users can cancel the booking only if the status is 'pending'

Make a Filter System on this page so the user can filter booking by status.

❖ **My Profile:** You'll Show Logged user Information on this page, Make the necessary Beautification for this page. There will be an upload profile picture button and an update button on this page. The user should be able to upload an image file and update his profile picture. (You can update it from both firebase and database if needed)

8. **The Admin Dashboard:** If the logged-in user is an Admin you'll show him the admin routes in the sidebar. By default, the dashboard will show the **statistics page.** On this page, you'll show 2 charts demonstrating the app usage data. Use the **React Apex Chart** for this section.

    ❖ A bar chart that shows bookings by date.
    ❖ A line chart that shows the comparison between the number of booked parcels and delivered parcels(Y Axis) on the basis of booking date(X axis) . **(optional)**

➔ **All Parcels:** On this page, you will show all the booked parcels by users in tabular form. Each Row will have the following info,

- User's Name Who booked the parcel,
- User's Phone Who booked the parcel,
- Booking Date
- Requested Delivery Date
- Cost
- Status
- **Manage Button**

Manage Button- Clicking on the manage button you'll show a modal. In this modal, You can select a Deliverymen From a select field and an Approximate delivery date From a Date input field. There will be an assign button. Clicking on this button you'll change the following thing in your database.

- Change The Booking status to **'On The Way'**
- Add the delivery men _id to the **'Delivery Men ID'** property in the booking.

HINTS - get all the delivery men from the database and map that array inside the select field and return option giving delivery men's id in the option value attribute.

Searching System: Make a Search System with Date Range on this page. There will be 2 date fields, admin can select a date from and date to for searching the booking that has **requested delivery date** in that selected date range.

HINTS - use MongoDB **$gte** and **$lte** operators to compare data.

➔ **All Delivery Men:** You will show all the delivery men in this page in a tabular form. Each Row will have,

- Delivery Man's Name
- Phone Number
- Number of parcels delivered
- Average review

➔ **All Users:** You will show all the registered users on this page in a tabular form. Each Row will have,

- User's Name
- Phone Number (you can take it as input when you register in website or can take from book parcel data in **Book a Parcel Page)**
- Number of parcels Booked
- Total Spent Amount(sum of the costs of all parcels booked)(**optional**)
- Make Delivery Men Button - Clicking on this button you'll change the user type to **Delivery Men from User**
- Make Admin Button - Clicking on this button you'll change the user type to **Admin from User.**

Make a pagination for showing users in the table, show 5 users per page.

9. **The Delivery Men's Dashboard:** If the logged-in user is a Delivery Man, you'll show routes for Delivery Man.

    ➔ **My Delivery List:** This page will show all the parcels assigned to the logged-in delivery man in a tabular form. He'll not be able to see parcels assigned to other Delivery Men. Each row will have

    - Booked User's Name
    - Receivers Name
    - Booked User's Phone
    - Requested Delivery Date
    - Approximate Delivery Date
    - Receivers phone number
    - Receivers Address
    - View Location Button(**See Bonus Point 1 For The Additional Guideline**)
    - **Cancel Button** - Clicking on this button status of the booking will be changed to Cancelled. Show an alert before sending a backend request.
    - **Deliver Button** - Clicking on this button status of the booking will be changed to Delivered. Show an alert before sending a backend request.

    ➔ **My Reviews:** On this page, you'll show all the reviews from the Users of this logged-in delivery man in card form. Each Card will have the Review giver's name, image, Review Giving Date, Rating out of 5, and Feedback Text.

Challenge Section

1. Use Shadcn as a Component Library for the whole website.

2. **See Location** - clicking on the see location button a modal will open. In this modal, you'll show a map and a pin on the map using the delivery

latitude and longitude given while booking a parcel. You can use MapBox and React MapGL or React Leaflet for free for this feature.

3. **Give Review** - Clicking on the Review Button a Modal will open. In this modal, there will be a form field and a submit button. The form will have the following fields,
    - User's Name(Auto Filled)
    - User's Image(Auto Filled)
    - Rating out of 5
    - Feedback Text input field
    - Delivery Men's ID (Auto Filled)

    Clicking on the submit button you'll save the info in a Mongodb collection and show it by getting them by delivery men's id in the 'My Reviews' page in the Delivery Man's Dashboard.

4. **Pay Button** - Clicking On this Button user will be redirected to the checkout page. Here the user can pay the parcel cost by entering card info. Implement **Stripe Payment System** For this feature. When The payment is successful Redirect The user To a Payment Success Page where you will show React Confetti Explosion.

5. Use JWT for any 2-3 routes (must use local storage not cookies)

Optional tasks

1. Use React Hook Forms
2. Use MongoDB aggregation for showing statistics data in the admin dashboard.
3. Make a light and Dark theme Toggling System.

What to Submit

1. **Admin email:**
2. **Admin password:**
3. **Delivery Men's email:**
4. **Delivery Men's password:**
5. **Front-end Live Site Link:**
6. **Client Side Github Repository Link:**
7. **Server Side Github Repository Link:**