B.Tech. 3rd Semester (2025-26) Project: Car Pooling Web Application

Project Overview

Project Start Date: 7th July 2025
Project End Date: 27th September 2025

Project Description

You are required to build a **Car Pooling Web Application** with the following features:

- **Homepage**: Display an overview of the app with a navigation bar linking to other pages.
- **Ride Search Page**: Allow users to search for available rides by entering details like source, destination, and date.
- **Offer Ride Page**: Enable users to post a ride by providing details like car type, seats available, and time.
- **Profile Page**: Display user information (e.g., name, contact, past rides).
- **Contact Us Page**: Include a form for users to submit inquiries or feedback.
- **Responsive Design**: Ensure the application is usable on both desktop and mobile devices.
- **Backend (Node.js)**: Implement API endpoints to store and retrieve ride and user data using a database.

Week-wise TODO List

Week	Dates	Tasks	Deliverable	Detailed Specification
Week 1	7th July - 13th July 2025	- Research at least 5 carpooling websites available in the market. - Prepare a document detailing their functionalities. - Understand how these functionalities can guide your project.	Document analyzing 5 carpooling websites and their functionalities.	**Purpose**: Understand the features of existing carpooling platforms to inform your project's design. **Functionality**: Research five carpooling websites (e.g., Uber Pool, BlaBlaCar, Lyft Line, Waze Carpool, Scoop). Document key functionalities like user registration, ride matching, payment systems, route optimization, user ratings, and real-time tracking. For each website, include: (1) a brief overview, (2) key features (e.g., ride matching based on location, cost-sharing options), (3) user interface elements (e.g., search forms, map integration), and (4) how

				these can be simplified for your project (e.g., basic ride search without payment). Submit a markdown or PDF document summarizing findings and their relevance to your app.
Week 2	14th July - 27th July 2025	- Prepare wireframes for the Car Pooling Web Application.	Wireframe designs for all pages.	**Purpose**: Plan the layout and structure of the application's user interface. **Functionality**: Create wireframes for the homepage, Ride Search, Offer Ride, Profile, and Contact Us pages. Include key elements like navigation bars, forms, buttons, and placeholders for images (e.g., car images, maps). Ensure wireframes reflect the functionalities identified in Week 1 (e.g., ride search form, user profile details). Use tools like Figma, Sketch, or pen-and-paper sketches. Submit wireframes as images or a PDF, explaining the purpose of each page and its components.
Week 3	28th July - 3rd August 2025	- Design static HTML and CSS and Bootstrap pages based on the wireframes.	Static HTML/CSS/Bootstrap pages for the application.	**Purpose**: Build the visual structure of the application using static HTML,CSS and Bootstrap. **Functionality**: Create HTML files for the homepage, Ride Search, Offer Ride, Profile, and Contact Us pages, following the Week 2 wireframes. Use semantic HTML tags (e.g., <header>, <nav>, <section>) for structure. Apply basic CSS and Bootstrap for layout (e.g., navigation bar, form alignment), colors (e.g., blue for navigation), and fonts (e.g., Arial). Include placeholder images from the provided project images. Ensure pages

				are visually consistent with the wireframes.
Week 4	4th August - 10th August 2025	- Enhance web pages using HTML, CSS, and Bootstrap.	Design HTML/CSS/Bootstrap pages.	**Purpose**: Improve the static pages with responsive and polished designs using Bootstrap. **Functionality**: Complete the design for Profile and Contact Us pages using Bootstrap classes and custom CSS. Ensure all pages (homepage, Ride Search, Offer Ride, Profile, Contact Us) have consistent styling, responsive layouts, and polished visuals (e.g., Bootstrap cards for ride listings, forms with validation styles). Test responsiveness across devices and ensure images are properly integrated. Submit all HTML/CSS/Bootstrap files.
Week 5	18th August - 24th August 2025	- Set up a React project using CDN links for React and Babel. - Convert static HTML pages to React components.	Initial React project with page components.	**Purpose**: Transition the static pages to a dynamic React frontend. **Functionality**: Set up an HTML file with React and Babel CDN links. Convert the Week 5 HTML pages (homepage, Ride Search, Offer Ride, Profile, Contact Us) into React components (Home.js, RideSearch.js, OfferRide.js, Profile.js, ContactUs.js). Include navigation using react-router-dom. Ensure the layout matches the Week 5 Bootstrap designs. Test navigation in a browser.
Week 6	25th August - 31st August 2025	- Implement forms in React components for Ride Search and Offer Ride.	React project with functional forms.	**Purpose**: Add interactive forms to the frontend. **Functionality**: In RideSearch.js, create a form with inputs for source, destination, and date. In OfferRide.js, create a form with inputs for source, destination, date, seats, and carType. Use React state to

				handle form inputs and display entered data temporarily (e.g., in a preview section). Apply Bootstrap classes (via className) for styling. Test form functionality in a browser.
Week 7	1st September - 7th September 2025	- Integrate React components with the provided API (Part 1).	React project with initial API integration.	**Purpose**: Connect the frontend to the provided API for ride and user data. **Functionality**: In RideSearch.js, fetch ride data from the provided API's GET /rides endpoint and display rides in a list (e.g., using Bootstrap cards). In Profile.js, fetch user data from GET /users/:id (use a static user ID). Handle API errors by displaying messages (e.g., "No rides found"). Ensure Bootstrap styling is maintained. Test API calls using browser developer tools.
Week 8	8th September - 14th September 2025	- Integrate React components with the provided API (Part 2).	React project with full API integration.	**Purpose**: Complete API integration for all features. **Functionality**: In OfferRide.js, send form data to the API's POST /rides endpoint to create rides. Add form validation to ensure required fields (e.g., source, destination) are filled, displaying error messages if invalid. In RideSearch.js, add a feature to view ride details using GET /rides/:id. Ensure submitted rides appear in the Ride Search list. Test all API interactions for accuracy.
Week 9	15th September - 21st September 2025	- Add advanced interactivity to React components. - Refine React components and styling.	Polished React frontend with advanced interactivity.	**Purpose**: Enhance and finalize the frontend's functionality and appearance. **Functionality**: In RideSearch.js, add a feature to delete rides using the API's DELETE /rides/:id endpoint. In OfferRide.js, allow editing rides using PUT /rides/:id. In

				ContactUs.js, implement a form to send feedback to a POST /feedback endpoint (if available). Add form validation feedback (e.g., highlight invalid fields). Refine Bootstrap and custom CSS styling (e.g., consistent fonts, colors) and add CSS animations (e.g., fade-in for ride listings). Ensure responsiveness (320px to 1200px).