

College Details :-

- **College Name** - **IEMS (Institute of Engineering & Management Studies) College**
- **Address** – **offic no,07, second floor, sarvody mall, kalyan sheel road, opp,apmc market, near patri pul, bhanunagar, bhoiwada, kalyan(west),mumbai,maharashtra(421301)**
- **Website** - <https://iemsedu.in/index.html>
- **Email** - info@iemsedu.in
- **Diploma Start to End** – Admission **Aug 2014** – Complete in **July 2017**
- **Degree Start to End** – Started in **Sep 2017** – Completed on **June 2020**

About Us College –

Welcome to **IEMS INSTITUTE OF ENGINEERING AND MANAGEMENT STUDIES**, is a leading **ISO-9001:2008**, certified B-School and registered with , govt. of Maharashtra under the SHREE EDUCATION FOUNDATION'S with the Registration No. **MAHA/2176/2004**.

For the Certification course of Management and Engineering has been started with intent to impart quality in the field of education and equip young people with knowledge and skills which would help them, make meaningful contribution to global competition. The institute offers academic programmes at both undergraduate and postgraduate levels.

Education INSTITUTE OF ENGINEERING AND MANAGEMENT STUDIES (IEMS) is established to provide value based learning to each aspiring professional and laid a path for leading a bright and prosperous career ahead. We provide distance learning and classroom study for Management and Engineering. As a participant, you get multiple courses to pick and choose. We make you create your future as per your time plan. Time is no more a constraint, when you are in IEMS.

Company Details :-

1. **Company Name:** **SIGMA SOFTWARE SOLUTIONS**
2. **Address:** Tech Centre, 5th floor, Plot No. 30, Phase 1, Rajiv Gandhi Infotech Park, MIDC, Hinjewadi, Pune, Maharashtra 411057.
3. **Website:** www.sigma-coding.com
4. **Organisation info:** Development, Consulting and Outsourcing
5. **No of Employees:** 110+
6. **Hr Details:**
 - hr@sigma-coding.com
 - **Name:** Naveen Human Resources Manager / Manager
 - **Mail id:** naveen@sigma-coding.com
 - **Phone Number:** +91 9603145386
7. **Reporting Manager:**
 - **Name:** Roja .R
 - Team lead / Supervisor / Manager / Project Manager / Reporting Manager
 - **Mail id:** roja.p@sigma-coding.com
 - **Phone Number:** +91 9603145386
8. **Mail Access:**
 - **URL:** sigma-coding.com/webmail
 - **Mail Id:** rushikesh.d@sigma-coding.com
 - **Password:** Utedge@1234\$

Introduction-About me & my Project

My name is **Rushikesh Dikule** I am from **Dharashiv, Maharashtra**. As a frontend developer with over all **4.3 years of experience** in **SIGMA SOFTWARE SOLUTIONS**, which is located in **Pune**. In this position I worked with prestigious customers like a **Specsavers, Rynair & Bollore Logistics**.

Currently, I am working on **Retail domain Project** for our **Client Specsavers**, Where we are responsible for developing the **Customer Management, Product Management, Order Management, Store Management, Online Sales, Supply chain & Logistics Management, Finance & Account Management**.

In this Project, I have Worked on **Product Management Module**, In Which includes key functionalities such as a **Product Catalog, Inventory Management, Pricing & Discounts, Product Recommendations & Vendor Management**.

In this Module, I have worked on **Product catalog management System**, In this Sub-module basically my focus on **Product Listing**, where I contributed to building **Single & Bulk Product Uploading and Managing & Updating Products** features.

In this module, **Redux-Toolkit** used For Complex **State Management**, **React-Hooks** for **Handling Side Effects & React-Router** for **Navigation**.

Also used Custom **CSS** for **Styling**, **Git & Github** for **Version Control** & **Jira** as a **Ticketing Platform**. Additionally, we using this project **chart.js** for **Displaying Data** and **Jest & React Testing Library** for **Testing purpose**.

I have strong experience working in Agile Scrum Framework, actively participating in **sprint planning**(2weeks), **daily stand-ups calls**, and **retrospective meetings**. I also use **Jira** for efficient **tracking & management** of **User Stories, Tasks and Bugs**. **In this my** career, I have gained extensive practical knowledge in **HTML, CSS, JavaScript, React, Bootstrap, Git and Jira**.

I have also worked with clients like **Ryanair** (from April 2022) and **Bollore Logistics** (from Dec 2020).

- At Client **Ryanair**, I contributed to building a **Flight Booking Management System**, which is streamlined the flight search, booking, and payment processes for millions of users. This system improved user experience and operational efficiency for the airline.
- At **Bollore Logistics**, I worked on a **Air cargo handling management**, which provided real-time tracking and management of shipments, ensuring transparency and efficiency in global supply chain operations.

Roles & Responsibilities in Company:

- Develop and maintain responsive user interfaces using React.js, HTML, CSS, and JavaScript.
- Create reusable, scalable, and efficient UI components.
- Implement state management using Redux, Context API, or other solutions.
- Integrate frontend applications with RESTful APIs and GraphQL.
- Optimize performance through lazy loading, code splitting, and caching.
- Ensure cross-browser compatibility and mobile responsiveness.
- Write and maintain unit and integration tests using Jest, React Testing Library, or Cypress.
- Use Git for version control and collaborate with teams via GitHub or GitLab.
- Stay updated with the latest frontend technologies and industry trends.
- Work closely with backend developers, UI/UX designers, and stakeholders.

JavaScript Experience:

- Developed and optimized web applications using modern JavaScript (ES6+).
- Implemented asynchronous programming with Promises, Async/Await, and API calls.
- Utilized JavaScript best practices for clean, maintainable, and efficient code.
- Worked with DOM manipulation, event handling, and performance optimization.

React.js & Redux/RTK Experience:

- Built scalable and reusable React components with functional and class-based approaches.
- Managed application state efficiently using Redux and Redux Toolkit (RTK).
- Implemented API interactions using RTK Query for efficient data fetching.
- Applied hooks like useState, useEffect, useMemo, and custom hooks for performance optimization.
- Integrated React Router for seamless navigation and dynamic routing.
- Ensured component reusability, modularity, and maintainability following best practices.

Experience in Git & GitHub

- Proficient in Git for version control, including branching, merging, and rebasing.
- Worked with GitHub for collaborative development, code reviews, and issue tracking.
- Used Git workflows like Feature Branching, Git Flow, and Pull Requests for efficient teamwork.
- Resolved merge conflicts and maintained a clean commit history with rebasing and squashing.
- Managed repositories, access controls, and CI/CD integrations in GitHub.
- Utilized GitHub Actions for automated testing and deployment processes.

Testing Experience (Jest & React Testing Library):

- I Wrote a unit tests and integration tests using Jest and React Testing Library.
- I will Tested React components, hooks, and Redux logic to ensure functionality and reliability.
- Implemented mock API responses and simulated user interactions.
- Ensured code quality and bug-free development by maintaining high test coverage.
- Used snapshots, assertions, and mock functions for testing UI components.

Specsaver → Product Management System → Product Catalog → Product Listing → 1.Single & Bulk Product Management, 2.Managing & Updating Products

1. Product Management System(**Module**)
 - a. Product Catalog Management(**sub-module**)

i. Product Listing

1. Single & Bulk Product Management :-

- a. Allows users to **add, edit, and manage products** individually or in bulk.
- b. Supports **bulk CSV/Excel uploads** for faster product listing.
- c. Implements **validation rules** to prevent incorrect product entries.
- d. Includes **image and media uploads** for better product representation.
- e. Enables **SEO-friendly product descriptions and metadata**.

2. Managing & Updating Products :-

- a. Edit Product details(Name, Price, Stock, etc).
- b. Bulk Updates using the dashboard or CSV import.
- c. Activate/deactivate products based on avilabilty.
- d. Search, filter, and sort products for easy management
- e. Soft delete (deactivation) or permanent deletion of products

Project Name :- **SPECSAVERS** (Aug 2023 – Currently Working still now){15+month}
Domain :- **RETAIL**
Model Name :- **PRODUCT MANAGEMENT SYSTEM**
Sub-Module :- Product Catalog{Product Listing(Single & Bulk Product)}
Website :- <https://www.specsavers.com/>
Client Description :- Specsavers Optical Group Limited is a British multinational optical retail chain, which operates mainly in the UK, Ireland, Australia, New Zealand, Canada, and the Nordic countries.

Description :-

- The **Product Management System** is a tool for retail businesses to manage their products more easily. It helps with organizing products, setting prices, tracking inventory, and handling promotions, all in one place.
- This system also connects smoothly with inventory and supply chain systems, making it easier for retailers to keep everything updated and make better decisions.
- As a frontend developer, I worked on creating a user-friendly interface using **React.js, JavaScript, HTML, CSS, and Tailwind CSS**. My main tasks included building responsive pages, ensuring the app could handle many products, and using tools like **Redux** for managing data, **React Router** for smooth page navigation, and **React Hooks** for writing cleaner code.
- I worked closely with the backend and design teams to make sure the app looked good and worked well across different devices.

Overview :-

- The **Product Management System module** in Specsavers' retail domain serves as a centralized platform for **managing product data, ensuring accuracy, consistency, and efficiency in product uploading**. It streamlines the addition of new products through **bulk uploads, validations, and categorization** while maintaining up-to-date details such as **pricing, images, and specifications**. The module enables seamless **inventory and pricing control, ensuring products** are correctly listed across **online and offline stores**. By integrating with supplier networks and compliance checks, it guarantees product quality and regulatory adherence. Ultimately, PMS enhances operational efficiency, reduces errors, and improves the overall customer experience.

Workflow of Product Catalog Module

The Product Catalog Management System helps businesses manage and organize their product database efficiently. It allows users to add, update, categorize, and manage products using a structured interface.

Technologies Used (Step-by-Step Frontend Breakdown)

A. Core Frontend Technologies

1. **HTML (Hyper Text Markup Language)**
 - Provides the basic structure of web pages.
 - Defines elements like **forms, tables, buttons, and modals**.
2. **CSS (Cascading Style Sheets) & Tailwind CSS**
 - Used for styling the user interface to ensure a **clean and responsive design**.
 - **Tailwind CSS** allows for rapid UI development with utility classes.
3. **JavaScript**
 - Adds interactivity and handles **dynamic UI updates**.
 - Fetches and processes product data from APIs.

B. React.js Framework (Component-Based Architecture)

4. **React.js**
 - Builds a **modular frontend** using reusable components.
 - Components like **ProductForm, ProductTable, CategoryList, and UploadManager**.
 - Ensures fast UI updates with **Virtual DOM**.
5. **Redux Toolkit (RTK) – State Management**
 - Manages product data, category structures, and user actions globally.
 - Ensures **smooth API communication** for fetching and updating products.
6. **Bootstrap & Tailwind CSS (UI Frameworks)**
 - **Bootstrap** provides ready-to-use UI elements (tables, buttons, modals).
 - **Tailwind CSS** helps with **custom styling and responsiveness**.
7. **Jest (Testing Framework)**
 - Ensures components function correctly before deployment.
 - Tests user interactions like **adding/editing/deleting products**.
8. **Git (Version Control)**
 - Tracks changes in code and ensures a **collaborative development process**.
 - Used with **GitHub or GitLab** for repository management.

Summary of the Workflow(Product Catalog Module)

1. After logging in, users will access the Product Catalog module.
2. Products can be uploaded one at a time or in bulk Product uploading with using an Excel or CSV upload.
3. Products are arranged using categories and subcategories.
4. The right categories are given to the products.
5. Product statuses can be edited, updated, deleted, or managed by users.
6. Product data is verified by the system before to publication.
7. Product search and filtering is simple for customers and store administrators.

Project Workflow & Development Process:

1. Requirement Analysis & UI/UX Design

- Worked closely with business analysts and designers to understand functional and UI requirements.
- Translated **Figma-based designs** into reusable and responsive **React components**.
- Ensured consistent styling and responsiveness using **Tailwind CSS**.

2. Frontend Development with React.js

- Developed intuitive **product listing, category structure management, and bulk upload features**.
- Built dynamic and reusable UI components to enhance modularity and scalability.
- Implemented **responsive layouts** for optimal performance on desktop and mobile devices.

3. State Management & API Integration

- Used **Redux Toolkit (RTK)** for efficient state management.
- Integrated REST APIs for **product data retrieval, creation, and updates**.
- Handled API errors gracefully with **custom error handling mechanisms**.

4. Performance Optimization & Testing

- Improved UI rendering with **React.memo** and **lazy loading**.
- Conducted **unit testing using Jest and React Testing Library**.
- Ensured compliance with accessibility standards and cross-browser compatibility.

5. Agile & Scrum Process with Jira

- Worked in **Scrum sprints**, participating in **daily stand-ups, sprint planning, and retrospectives**.
- Used **Jira** to track and manage tasks efficiently.
- Collaborated with backend developers and QA teams for seamless project execution.

Team Size :- 10 members are in this project

- Front-end Team :- 4
 - Front-end Lead :- 1
 - Oversees UI/UX implementation.
 - Reviews code quality and best practices.
 - Coordinates with Backend & QA teams.
 - Front-end Developer :- 3
 - Develop UI for **Single & Bulk Uploading, Product Management**.
 - Implement **validation, media uploads, and search/filtering**.
 - Integrate APIs with backend services.
- Back-end Team :- 3
 - Back-end Lead :- 1
 - Back-end Developer :- 2
- QA Engineer :- 1
- Automation Tester :- 1
- DevOps Engineer :- 1

Product Listing Flow (Step-by-Step Process)

1. User Authentication & Access

- User logs in to the system with credentials.
- Based on user roles (Admin, Vendor, Manager), access to **Product Listing** is granted.

2. Navigating to Product Listing Section

- From the dashboard, the user selects **Product Management System**.
- Clicks on **Product Catalog** → **Product Listing**.
- Two options are available:
 - i. **Single Product Upload**
 - ii. **Bulk Product Upload (CSV/Excel)**

3. Single Product Upload Flow

- User clicks "**Add New Product**".
- Fills in product details in a form:
 - i. **Basic Information:** Name, SKU, Brand, Description
 - ii. **Pricing:** Regular Price, Discount Price, Tax Information
 - iii. **Inventory Details:** Stock Quantity, Availability Status (Active/Inactive)
 - iv. **Category Assignment:** Select appropriate category/subcategory
 - v. **Product Images & Media Upload**
 - vi. **SEO Optimization:** Keywords, Meta Description, Slug
- Clicks **Save & Publish** → Product is added to the catalog.

4. Bulk Product Upload Flow (CSV/Excel)

- User downloads the **CSV/Excel Template** from the system.
- Fills in multiple product details in the file (Name, SKU, Price, Category, Stock, etc.).
- Uploads the file to the system.
- The system **validates the file**:
 - i. Checks for missing/incorrect values.
 - ii. Displays errors (if any) for correction.
 - iii. If errors exist, the user **fixes & re-uploads** the file.
- Once validated, all products are **added in bulk & published**.

5. Managing & Updating Products

- User can **search, filter, and edit products** from the product list.
- Available actions:
 - i. **Edit:** Modify product details.
 - ii. **Delete:** Remove a product.
 - iii. **Bulk Actions:** Update multiple products at once.
- Users can toggle **Active/Inactive** status.

6. Approval & Review (Optional)

- If an approval system is enabled, newly added products require admin review.
 - Once approved, products are **live on the store/catalog**.
-

Flowchart Representation (Pointwise)

1. **User Logs in** → Navigates to **Product Catalog** → Clicks **Product Listing**
 2. **Choose Upload Type:**
 - **Single Product Upload**
 - **Bulk Product Upload (CSV/Excel)**
 3. **Enter Product Details (Manual Entry or File Upload)**
 4. **Validation & Error Checking**
 5. **Save & Publish** (or Fix Errors & Retry)
 6. **Products Appear in Catalog**
 7. **Users Can Edit, Update, or Delete Products**
 8. **Approved Products Go Live on Store**
-

My Contribution in Project

- Built and developed the **frontend** using **React.js**, **JavaScript**, **HTML**, **CSS**, and **Tailwind CSS**.
- Implemented dynamic and **responsive web pages** that handle a large number of products efficiently.
- Used **Redux** for managing the application's state and ensuring real-time data flow across different components.
- Implemented **React Router** for smooth page navigation between different modules.
- Used **React Hooks** to improve code maintainability and reduce the complexity of using class components.
- Collaborated closely with **backend** and **UX/UI teams** to ensure seamless integration and design functionality.

Git Experience in this Project:

- **Version Control & Branching Strategy:** Followed Git best practices, including feature branching, rebasing, and merging, to maintain a clean commit history.
- **Collaborative Development:** Worked in a team using GitHub/GitLab, creating pull requests (PRs) and conducting thorough code reviews.
- **Conflict Resolution:** Resolved Git merge conflicts efficiently, ensuring smooth collaboration across multiple developers.
- **CI/CD Integration:** Contributed to Git-based CI/CD workflows, automating deployments and ensuring code quality.

Flow Of Retail Domain:-

1. Product Management System → Product Creation & Categorization

a. Product Upload (Creation & Setup)– Add, edit, and manage product details.

- **Product Data Collection** – Gather product details (name, SKU, price, specifications, images).
- **Data Formatting & Validation** – Ensure data accuracy, correct format, and mandatory fields.
- **Single & Bulk Product Uploading** – Add products manually or via CSV/Excel bulk upload.
- **Image & Media Upload** – Attach high-quality images and ensure correct format/resolution.
- **Product Categorization & Variants** – Assign categories, manage color/size/material options.
- **Inventory & Pricing Setup** – Define stock levels, warehouse locations, and pricing strategies.
- **Quality Check & Approval** – Validate product details and submit for approval before activation.
- **Product Activation & Storefront Sync** – Make products live and synchronize with sales channels.
- **Testing & Verification** – Check product listings, stock availability, and correct pricing.
- **Go Live & Monitoring** – Continuously track product performance, stock levels, and customer feedback.

b. Product Categorization – Organize products by type, brand, and attributes.

c. Product Information Management (PIM) – Maintain descriptions, specifications, and images.

d. Inventory Management – Track stock levels, warehouses, and replenishment.

e. Vendor Integration – Manage supplier relationships and purchase orders.

f. Pricing & Promotions – Set base prices, discounts, and special offers.

g. Order Management – Handle order processing, shipping, and returns.

h. Compliance & Quality Control – Ensure regulatory compliance and product safety.

i. Reporting & Analytics – Generate sales insights, demand forecasts, and stock reports.

2. Inventory & Stock Management → Stock Allocation & Tracking

3. Supplier & Vendor Management → Supplier Onboarding & Ordering

4. Pricing & Promotions → Price Setup, Discounts & Offers

5. Order Management → Order Placement & Processing

6. Fulfillment & Logistics → Shipping & Delivery Coordination

7. Storefront Integration → Syncing with Online & Offline Stores

8. Customer & Compliance Management → Quality Control & Regulations

9. Reporting & Analytics → Sales Performance & Forecasting

My Role for Single & Bulk Product :-

- **Single Product Uploading**

- Develop intuitive UI components for adding a single product, ensuring smooth navigation and user experience.
- Implement form validation to ensure mandatory fields (SKU, name, price, category, images) are correctly entered.
- Use **React.js (Redux/RTK)** to manage state efficiently and update product details dynamically.
- Handle API integration to send product data to the backend and display real-time status updates.
- Provide feedback messages (success, error, warnings) using toast notifications or modals.

- **Bulk Product Uploading**

- Design and develop a bulk upload feature that allows users to upload **CSV/Excel** files.
- Implement a file validation system to check data format, missing values, and duplicate entries.
- Parse and preview uploaded data before submission, allowing users to make corrections.
- Integrate with APIs to process bulk uploads efficiently and display progress updates.
- Handle error logging and provide meaningful feedback if any issues occur during the upload process.

My Role Managing & Updating Product:-

- **Managing Products**

- **Develop Product Listing Pages** – Build dynamic tables or grids to display product information, integrating features like sorting, filtering, and searching.
- **Implement Pagination & Lazy Loading** – Optimize performance when handling large datasets by loading products efficiently.
- **Enable Product Status Updates** – Allow users to activate, deactivate, or archive products with real-time UI updates using **Redux/RTK** state management.
- **Role-Based Access Control** – Ensure proper permissions for editing and updating products based on user roles.

- **Updating Products**

- **Edit Product Details** – Provide UI components to edit product attributes like name, price, inventory, and images.
- **Inline & Bulk Editing** – Allow users to update single or multiple products at once through batch update functionality.
- **File & Image Management** – Enable product image replacement or additional image uploads with preview options.
- **Validation & Error Handling** – Ensure correct data formats before submitting changes and display meaningful error messages.

Tech stack :-

| Category | Technologies/Tools |
|----------------------|--|
| Front-End Technology | React |
| State Management | Redux, Context API (where needed) |
| Data Fetching | Axios, RESTful APIs |
| Styling | CSS, SCSS (pre-processor), Bootstrap |
| Backend | <ul style="list-style-type: none">- Java with Spring Boot (for Microservices)- Node.js with Express (for specific microservices, if applicable) |
| Databases | <ul style="list-style-type: none">- PostgreSQL (for structured data like user information, cargo details, transactions)- MongoDB (or another NoSQL database for unstructured data like logs, analytics, real-time monitoring) |
| Testing | <ul style="list-style-type: none">- Jest, React Testing Library (for front-end) |
| Tools | <ul style="list-style-type: none">- Postman (API testing)- Jira (project management)- GitHub (version control) |
| CI/CD | Jenkins |
| Deployment | AWS |

Project Name :- Ryanair (April 2022 – Aug 2023){16 month (1.4years)}

Domain :- Airline

Model Name :- Flight Booking Management System

Sub-Model :- Booking & Reservation Management

Website :- <https://www.ryanair.com/ie/en>

Client Description :- Ryanair is an Irish ultra low-cost airline group headquartered in Swords, Dublin, Ireland. Established in 1984, it has grown to become one of Europe's largest and most recognized airlines

Project Description:

This project is a comprehensive web application designed to automate and streamline the entire process of booking flights. It offers an integrated platform where users can search for flights, book tickets, modify reservations, and track flight statuses in real time. For airline operators, the system provides tools to manage flight schedules, pricing, and passenger details, ultimately reducing manual effort and errors while enhancing customer satisfaction. The platform is built to support scalability and seamless integration with third-party APIs (e.g., payment gateways, airline data feeds, weather updates) to ensure that both users and administrators have access to accurate, real-time information.

Project Details :- The Ryanair Airline Project is a **flight booking and management system** designed to enhance user experience for **flight search, booking, seat selection, and check-in processes**. The platform ensures a smooth, efficient, and responsive UI, enabling customers to easily manage their travel plans.

As a **Junior React.js Developer**, I was responsible for designing and optimizing the frontend interface using **React.js, Redux, Redux Toolkit (RTK), HTML, CSS, JavaScript, Tailwind CSS, and Jest**. I also worked with **REST APIs, async programming, error handling, and callback functions** to ensure smooth data fetching and user interactions. The project followed an **Agile Scrum methodology**, with **Jira** for sprint planning and task tracking.

Project Overview

The **Flight Search & Availability Module** is a web application that allows users to search for available flights based on their travel preferences, including **one-way, round-trip, and multi-city** options. The system fetches real-time flight data and displays available options with relevant details like **airline, price, departure time, duration, and stopovers**.

Project Workflow & Development Process:

1. Requirement Analysis & UI/UX Design

- Collaborated with designers and stakeholders to understand functional requirements.
- Converted **Figma-based UI designs** into responsive and reusable **React components**.
- Used **Tailwind CSS** for efficient styling and consistent UI across devices.

2. Frontend Development with React.js & Redux

- Developed interactive **flight search, booking, and seat selection features**.
- Implemented smooth **date pickers, dropdowns, and real-time availability updates**.
- Used **Redux Toolkit (RTK)** for efficient state management.
- Managed complex UI states such as **user-selected seats, booking details, and API responses**.

3. State Management & API Integration

- Integrated **REST APIs** to fetch flight details, booking confirmations, and user preferences.
- Handled **async API requests** with proper loading indicators and error handling.
- Used **callbacks and Promises** for seamless data flow between components.
- Managed API response caching to **reduce redundant requests and improve performance**.

4. Error Handling & Performance Optimization

- Implemented **try-catch blocks, error boundaries, and centralized error handling mechanisms**.
- Used **React.memo, lazy loading, and code splitting** to optimize UI performance.
- Improved API response handling with **debouncing, throttling, and optimistic updates**.

5. Testing & Agile Process with Jira

- Conducted **unit and component testing using Jest and React Testing Library**.
- Ensured **high test coverage** to minimize regressions.
- Worked in **Scrum sprints**, participating in **daily stand-ups, sprint planning, and retrospectives**.
- Managed development tasks and issue tracking via **Jira**.

Key Objectives

- Provide an intuitive **search interface** for users to input flight details.
- Fetch real-time **flight availability data** using an external API.
- Display **search results dynamically** with sorting and filtering options.
- Manage flight search state globally using **Redux Toolkit (RTK)**.
- Ensure a **responsive UI** using **React & Tailwind CSS**.

Modules Airline:-

- **Booking & Reservation Management**

- **Flight Search & Availability (One-way, Round-trip, Multi-city)**

- **Search by:** Destination, Date, Time, Flight Class, Direct/Stopover
 - **One-way:** Select source & destination with a single date
 - **Round-trip:** Select departure & return dates
 - **Multi-city:** Search & book multiple destinations in one transaction
 - **Filters:** Airlines, Price Range, Departure Time, Duration, Stops
 - Airline filter (e.g., Ryanair, Emirates, Lufthansa)
 - Price range selection
 - Departure & arrival time filters
 - Flight duration & stopovers
 - **Real-time Availability Check**
 - Fetch available seats & pricing dynamically
 - Display seat classes (Economy, Business, First-Class)
 - Indicate sold-out flights
 - **Fare Comparison & Promotions Display**
 - Compare fares for different airlines & days
 - Display promo codes & discount options

- Passenger Management
 - Booking & Reservation Management :-
 - Pricing & Fare Management
 - Payment Processing (Credit/Debit Card, Wallets, Vouchers)
 - Booking Modification & Cancellation
 - Loyalty & Membership Program
 - Check-in & Boarding Pass Generation
 - Notifications & Alerts
 - Admin & Reporting Panel

- Flight Management
- Customer Management
- Ancillary Services
- Check-in & Boarding
- Payment & Billing
- Customer Support & Chatbot
- Operations & Crew Management
- Safety & Compliance
- Analytics & Reports

Workflow Of My Module :-

- **User Inputs Flight Details** → Origin, destination, travel dates, passengers.
- **Redux Dispatches API Call** → Flight search request is sent to an API.
- **API Returns Available Flights** → Data is stored in Redux state.
- **Flights Are Displayed Dynamically** → List updates based on search results.
- **User Selects a Flight** → Proceeds to booking or modifications.

Roles & Responsibility :-

- Front-End Developer-
 - Develop **user-friendly UI components** for flight search and results display.
 - Implement **responsive design** using **Tailwind CSS**.
 - Integrate **Redux Toolkit (RTK)** for state management.
 - Handle **API requests and responses** using **Axios & Redux Thunk**.
 - Implement **sorting, filtering, and pagination** for search results.
 - Optimize performance with **lazy loading and caching**.
- UI/UX Developer -
 - Design **intuitive flight search forms and result displays**.
 - Create **interactive seat selection & filters**.
 - Ensure **mobile-first design** for accessibility.
 - Improve **user flow and experience** based on testing feedback.

THE END PROJECT **AIRLINE(RYANAIR)**

Project Name :- **Bolloré Logistics** (Feb 2021 – March 2022){13 month (1.1years)}

Domain :- **Logistic**

Model Name :- **Logistic Tracking System**

Sub-Module :-

Website :- <https://www.bolloré.com/en/>

Client Description :- Bolloré Logistics is a leading international transport and logistics company headquartered in Puteaux, France. Established in 1986, it has developed a robust global presence, operating in 146 countries with a workforce of approximately 15,000 employees.

Client Details:-

Bolloré Logistic is a French multinational corporation founded in **1822**, ranking among the world's top 500 companies. Headquartered in **Puteaux, France**, the group is predominantly controlled by the Bolloré family, ensuring a stable shareholder base that supports long-term investment strategies.

Over its extensive history, Bolloré has diversified into several key sectors:

- **Transportation and Logistics:** Through Bolloré Logistics, the company offers international transport and logistics services, including multimodal transport, customs and regulatory compliance, global supply chain management, and industrial project support.

Project Description:

Bolloré Logistics is a comprehensive logistics management system designed to streamline and optimize supply chain operations. The platform provides end-to-end solutions for freight forwarding, warehousing, inventory management, and transportation. It integrates advanced tracking systems, real-time analytics, and automated workflows to enhance operational efficiency and ensure seamless logistics management.

Bolloré Logistics is a global supply chain management platform that streamlines logistics, transportation, and freight operations. The system is designed to provide efficient multimodal transport management, real-time shipment tracking, warehouse management, and automated documentation.

As a **Junior React.js Developer**, I worked extensively on the frontend of this platform, utilizing **HTML, CSS, JavaScript, and React.js** to create responsive, scalable, and user-friendly interfaces.

- **Freight Management:** Supports air, sea, and land freight with real-time tracking.
- **Warehouse & Inventory Management:** Optimizes storage, order fulfillment, and stock tracking.
- **Automated Documentation:** Manages invoices, shipping labels, and customs clearance.
- **Route Optimization:** Enhances delivery efficiency with AI-driven logistics planning.
- **Reporting & Analytics:** Provides insights into shipment status, costs, and performance metrics

Project Workflow & Development Process:

1. Requirement Analysis & UI/UX Design

- o Collaborated with business analysts and designers to understand functional requirements.
- o Implemented Figma-based UI designs into reusable React components using **HTML, CSS, and JavaScript**.
- o Ensured cross-browser compatibility and responsiveness.

2. Frontend Development with React.js

- o Developed interactive dashboards for shipment tracking and logistics management.
- o Built reusable and optimized React components for forms, modals, and tables.
- o Integrated **Bootstrap** and **CSS Flex/Grid** to enhance UI styling.

3. State Management & API Integration

- o Used **Redux Toolkit (RTK)** to manage application state efficiently.
- o Integrated RESTful APIs to fetch real-time shipment and inventory data.
- o Implemented loading indicators and error handling for a better user experience.

4. Performance Optimization & Testing

- o Optimized component rendering using **React.memo** and lazy loading.
- o Used **Jest and React Testing Library** for unit and component testing.
- o Conducted UI testing and fixed accessibility issues.

5. Deployment & Version Control

- o Used **Git for version control**, collaborating with backend and QA teams.
- o Deployed frontend updates in coordination with DevOps teams.
- o Debugged issues and optimized code for production.

Jira Workflow

1. Ticket Assignment & Acknowledgment:

- Once a ticket is assigned by the Team Lead, we acknowledge it and move it to "In Progress".

2. Initial Analysis:

- Perform technical and business analysis.
- If clarification is needed, communicate with the Business Analyst (BA).
- If further details are required, a meeting with the Team Lead, Manager, and client is scheduled.

3. Development:

- Begin coding in the development environment.
- Update the ticket status and details.

4. Testing (Unit & Integration):

- Perform unit and integration testing.
- If successful, move the ticket to "Code Review" or "Ready for QA".

5. Deployment to QA Environment:

- Deploy the code to the QA environment.
- Resolve any issues and redeploy to QA, if necessary.

6. User Acceptance Testing (UAT):

- Deploy to UAT after QA.
- Resolve any UAT issues and retest.

7. Production Deployment:

- Deploy to the Production environment after successful UAT.
- If no issues occur, mark the ticket as "Done" or "Closed".

Communication & Reporting

- All communication, daily updates, comments, and reports are tracked in Jira.
- Jira stores the entire history of task assignment, work logs, and resolution processes.

1. Day-to-Day Activities as a React Developer Morning Tasks (9:30 AM – 12:30 PM)

a. Daily Scrum Meeting (15–30 mins)

- i. Attend daily stand-up under Agile methodology.
- ii. Discuss completed tasks, current tasks, and blockers.
- iii. Align with backend developers, QA, and project managers.

b. Code Review & Merge Requests (30–45 mins)

- i. Review pull requests on GitHub.
- ii. Ensure code follows coding standards (Prettier, SonarLint).
- iii. Provide feedback or approve merge requests.

c. Work on Assigned Tasks (Feature Development / Bug Fixes)

- i. Implement new UI components for Product Catalog Module using React.js & React Hooks.
- ii. Develop responsive and scalable layouts with Material UI & CSS.
- iii. Optimize performance by reducing re-renders and improving load times.
- iv. Use Redux Toolkit for state management and handle async operations.
- v. Integrate RESTful APIs with Axios.

2. Afternoon Tasks (1:30 PM – 5:30 PM)

a. Testing & Debugging

- i. Perform unit testing with Jest.
- ii. Debug UI issues and fix rendering problems.
- iii. Verify API responses and handle errors efficiently.

b. Collaboration & Code Integration

- i. Work with backend developers to consume APIs and troubleshoot data inconsistencies.
- ii. Collaborate with QA team for bug fixes and enhancements.
- iii. Follow Agile workflows: update Jira with task progress and blockers.

c. Performance Optimization

- i. Identify and resolve performance bottlenecks.
- ii. Optimize API calls using caching techniques.
- iii. Improve component rendering efficiency to enhance user experience.

3. Evening Tasks (5:30 PM – 7:00 PM)

a. Version Control & Deployment

- i. Push code updates to GitHub and create pull requests.
- ii. Merge feature branches into the main branch after review.
- iii. Participate in CI/CD pipeline and deployment discussions.

b. Documentation & Knowledge Sharing

- i. Update technical documentation for newly developed features.
- ii. Share knowledge with the team on best practices and improvements.

Q. All gose good then what is the Project & Whats Roles & responsibility for this?

If Any Questions from your Side ?:-

1. What is Tech stack in your project?
2. What about project duration?
3. How many Team size?
4. What My roles & responsibilities
5. What is the culture of your organization?

