# Introduction

This To-Do List application is built using React and Material UI. The app allows users to create, edit, and manage to-do lists, as well as toggle between dark and light themes. Users can log in or register, manage their settings, and their data persists using browser localStorage. The design follows modern principles with an emphasis on user experience, accessibility, and responsive design.

# Colour Palette

The colour palette for the application is defined based on the Material UI theme with light and dark mode options.

* Light Mode:
  + Background: #F5F5F5
  + Primary: #1976D2 (Material Blue)
  + Secondary: #FF5722 (Material Orange)
  + Text: #212121 (Dark Gray)
* Dark Mode:
  + Background: #212121
  + Primary: #BB86FC (Purple)
  + Secondary: #03DAC6 (Teal)
  + Text: #FFFFFF (White)

The dark mode aims to reduce eye strain for users in low-light environments. The colours are adjusted accordingly in components using Material UI's theme capabilities.

# Libraries Used

* **React**: Core framework for building the user interface.
* **Material UI**: For implementing responsive and accessible UI components with pre-defined theming.
* **React Router**: For handling routing between various pages (login, register, dashboard, etc.).
* **Formik + Yup**: For form management and validation.
* **React Toastify**: For displaying toast notifications for success, errors, or other feedback.
* **LocalStorage**: Used to persist user data (to-do lists, user settings).

# Design Decisions

## Component-based Architecture

The application is split into reusable components such as:

* **Button**: A reusable button component with dynamic theming and styles.
* **ToggleThemeButton**: A toggle for switching between light and dark modes.
* **ToDoCard**: A card component used to display individual to-do items.
* **Modals**: Used for creating and editing tasks.

Each component is independent, making them easy to maintain, test, and update without affecting other parts of the application.

## Single Page Application (SPA)

The application follows a SPA architecture, where React Router handles navigation. This improves the user experience by reducing page load times.

## Responsive Design

All components are responsive, ensuring the application works across different screen sizes (from mobile to desktop). Material UI’s grid system and breakpoints are leveraged to achieve this.

# Routing

The application uses React Router for client-side routing. The following routes are defined:

* /login: Login page for user authentication.
* /register: Registration page for new users.
* /dashboard: Main dashboard displaying all the to-do lists.
* /todo/

: Displays details of a specific to-do.

* /settings: User settings page where the user can update their profile information.

# Forms and Validations

* Formik is used for handling forms, including the user registration, login, and settings forms.
* Yup is used for validating forms:
  + Name validation: Must be at least 2 characters long.
  + Email validation: Checks for valid email format.
  + Password validation: Ensures strong password requirements when used in authentication forms.
* When invalid input is detected, the forms will not submit, and the error messages will be displayed dynamically using Formik.

# User Authentication

* Login/Registration: The app uses a simple login and registration system where user credentials (email, name, password) are stored in localStorage. Upon successful login, the user is redirected to the dashboard. The user’s authentication state is managed globally using a custom hook (useAuth).

# To-Do Management

* Create, Edit, Delete: Users can create to-do lists with tasks, edit them, and delete them. The tasks inside a to-do list can also be individually managed (marked as complete, deleted, etc.).
* Unique Task Validation: Duplicate task names are not allowed. A user-friendly error is displayed if the user attempts to add a task with the same name.
* Date and Time Handling: Each to-do entry includes a createdAt and lastEdited timestamp. This allows users to view when each to-do list was created and last modified.

# Theming (Dark/Light Mode)

The app includes a **dark and light mode toggle**:

* Material UI’s theming system dynamically applies different color palettes depending on the selected mode.
* The toggle is implemented using a sun and moon icon to visually indicate the current mode. The theme toggle state persisted using localStorage.

# Notifications

* React Toastify is used for handling notifications. This includes success messages (e.g., "To-do list created successfully"), error messages (e.g., "Invalid input"), and other user feedback.
* Notifications are styled to match the application's overall theming (dark/light mode).

# LocalStorage Integration

The app stores data such as:

* User Information: (name, email, password) for authentication.
* To-Do Lists: Each user's to-do lists are stored in localStorage with a unique identifier.
* Theming Preferences: The user’s theme preference (dark or light mode) is persisted across sessions.

# Error Handling

* Validation errors: Handled via Formik and Yup (e.g., ensuring required fields are filled).
* Task Validation: Ensures that duplicate tasks cannot be added.
* Auth Error: If the user is not found or credentials are invalid, error messages are shown on the login page.

# Responsive Design

* Material UI Grid: Ensures that the layout adjusts to different screen sizes. Cards, buttons, and modals resize according to the viewport size.
* The app is designed to be fully functional on both desktop and mobile devices. Flexbox and grid layouts are used to ensure responsiveness.