நான்
முதல்வன்
உலகை வெல்லும் இளைய தமிழகம்

IBM

College Code : 9509

College Name : Holy Cross Engineering College

Department : Computer Science Engineering

Student NM id DAE074EB4A8886B8B224E01DC0C33A55

Roll No : 950923104044

Date : 15.09.2025

Project Name : IBM-FE-List-to do application

Completed the project named as

Sumbitted By,

Sanjay K

9342667280

Phase 1 — Problem Understanding & Requirements

Project: List-to-Do Application

1. Problem Statement

In today's fast-paced world, individuals often struggle with remembering and prioritizing their daily tasks, leading to missed deadlines, decreased productivity, and stress. Traditional methods such as paper notes or memory are unreliable and inefficient.

A digital "List-to-Do" application can help users efficiently organize, prioritize, and track their tasks. The solution will provide features like task creation, editing, deletion, categorization, and timely reminders. This ensures users stay organized, meet deadlines, and improve productivity.

2. Users & Stakeholders

Primary Users

- **Students** – manage personal assignments, deadlines, exams, and schedules.
- **Working professionals** – track meetings, project tasks, and reminders.
- **Homemakers** – maintain daily chores, shopping lists, and personal activities.

Secondary Users (Optional – for future extension)

- **Team members** – could use shared lists in collaborative features (not in MVP since we're front-end only).
- **Parents** – might track children's tasks (future scope, not in front-end MVP).

Stakeholders

- **End-Users** – people who will directly use the app's front-end interface.
- **Front-End Developer (You)** – responsible for design, layout, and user interaction.
- **Mentor/Faculty/Client** – evaluates the project and provides feedback.

3. User Stories

1. As a student, I want to add and edit tasks so that I don't miss my assignment deadlines.
2. As a professional, I want to prioritize my tasks (High/Medium/Low) so that I can focus on important work first.
3. As a homemaker, I want to create shopping lists so that I don't forget items while shopping.
4. As a user, I want to mark tasks as complete so that I can track my progress.
5. As a user, I want to receive reminders before deadlines so that I stay on schedule.
6. As a user, I want to delete tasks so that I can remove completed or irrelevant ones.
7. As a user, I want my data stored securely so that I can access it anytime without loss.

4. MVP Features

- **Task Management**: Create, edit, delete tasks
- **Prioritization**: Mark tasks with priority levels (High/Medium/Low)
- **Status Tracking**: Mark tasks as *Completed / Pending*
- **Reminder & Notifications**: Alerts before deadlines
- **Data Persistence**: Store tasks in database or local storage
- **User-Friendly Interface**: Clean, minimal design for ease of use

(Optional for later versions: Task sharing, recurring tasks, analytics, voice input)

5. Wireframes / API Endpoint List

Wireframes

- **Login/Signup Screen** – User authentication
- **Dashboard/Home** – Display list of tasks with filters (All, Completed, Pending)
- **Add Task Screen** – Fields: Title, Description, Due Date, Priority
- **Edit Task Screen** – Update existing task details
- **Notifications** – Pop-ups or reminders before deadlines

API Endpoint List

- POST /signup → Register new user
- POST /login → User authentication

- GET /tasks → Fetch all tasks of a user
- POST /tasks → Create a new task
- PUT /tasks/{id} → Update task details
- DELETE /tasks/{id} → Remove a task
- GET /tasks?status=completed → Fetch completed tasks
- GET /tasks?priority=high → Fetch high priority tasks

6. Acceptance Criteria

Task Creation

- User can create a task with title, description, due date, and priority.
- Task is saved in the database and visible immediately.

Task Editing

- User can update the title, description, due date, and priority.
- Changes reflect instantly in the task list.

Task Deletion

- User can delete any task permanently.
- Deleted tasks are removed from the task list and database.

Task Status Update

- User can mark tasks as *Completed* or *Pending*.
- Completed tasks should move to a separate section or get a strike-through style.

Reminders

- User receives a notification before the due date/time (e.g., 10 mins prior).
- Notification must display task title and due time.

Data Persistence

- All tasks remain stored after logout/login.
- User's data is only accessible to them.