ZIDIO
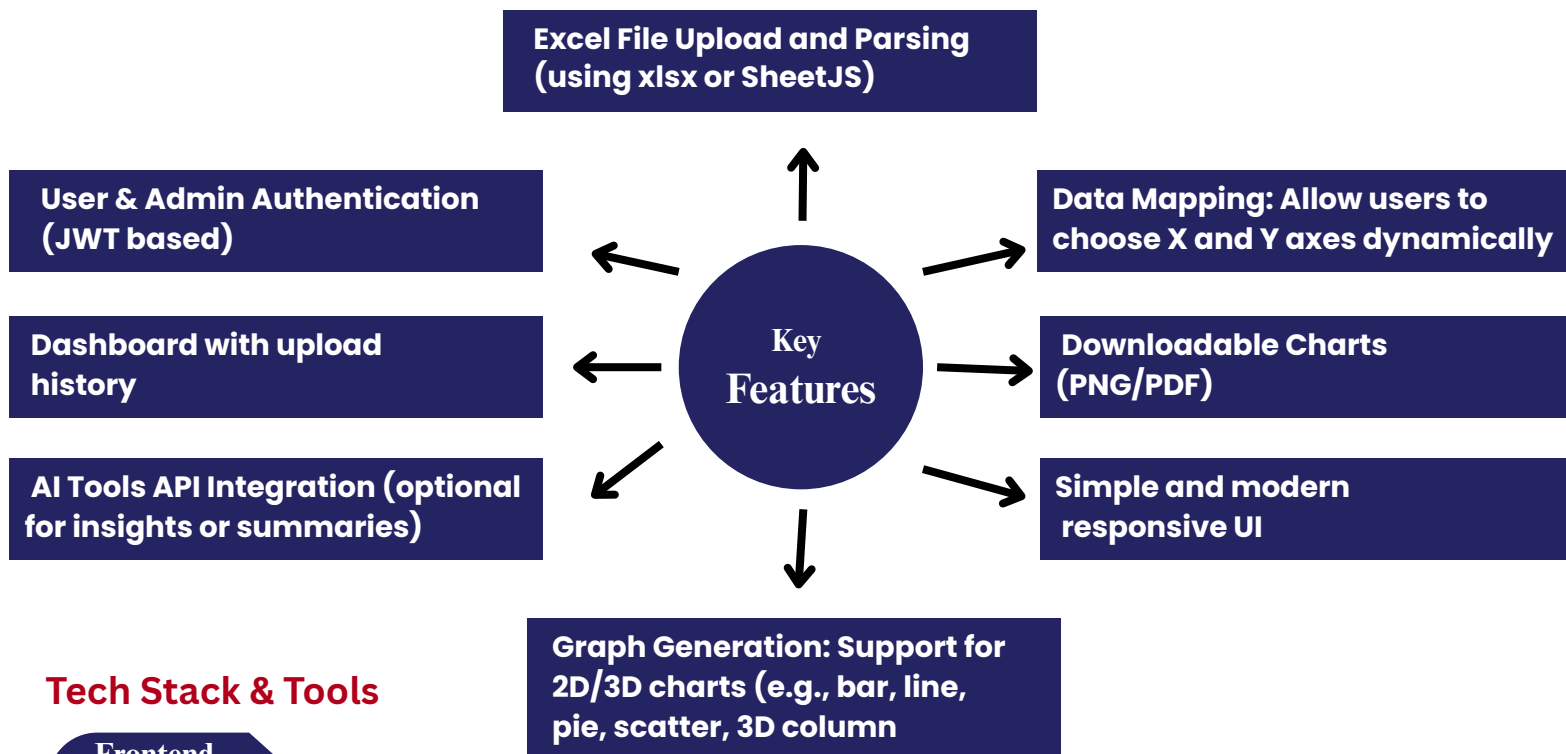## DEVELOPMENT

MERN
FULL STACK
PROJECT

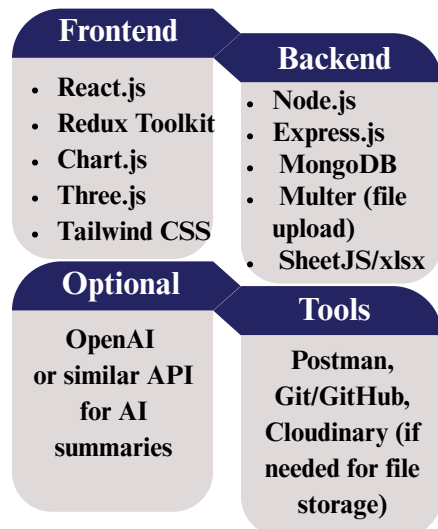## ASSIGNMENT

M E R N

Project 1: Excel Analytics Platform

These two MERN (MongoDB, Express.js, React.js, Node.js) stack-based full-stack projects are designed for students to complete over a 10-week period. Each project is divided into 5-week modules with structured goals, guidance, and references for development. The purpose is to give hands-on experience with complete application development using the MERN stack.

Description:

A powerful platform for uploading any Excel file (.xls or .xlsx), analyzing the data, and generating interactive 2D and 3D charts. Users can select X and Y axes from the column headers of the Excel file, choose chart types, and generate downloadable graphs. Each user's history of uploads and analysis is saved and visible on their dashboard. Admin can manage users and data usage. The platform will optionally integrate AI APIs to provide smart insights or summary reports from the uploaded data
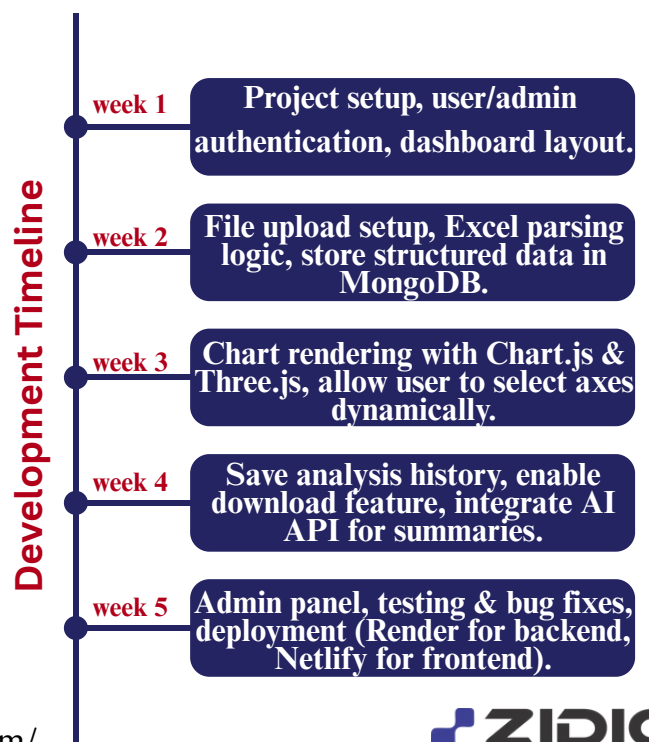
Excel File Upload and Parsing (using xlsx or SheetJS)
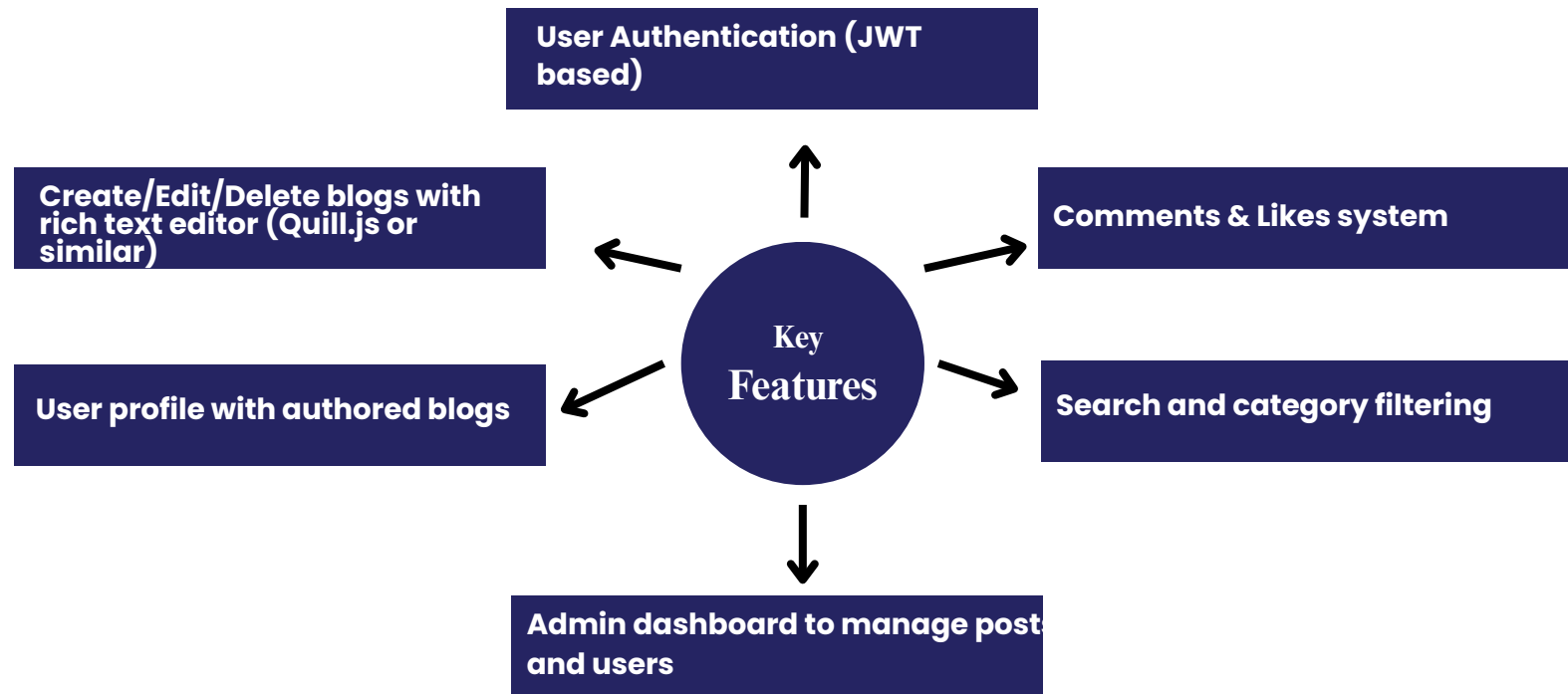
User & Admin Authentication (JWT based)

Data Mapping: Allow users to choose X and Y axes dynamically

Key Features

Dashboard with upload history

Downloadable Charts (PNG/PDF)

AI Tools API Integration (optional for insights or summaries)

Simple and modern responsive UI

Graph Generation: Support for 2D/3D charts (e.g., bar, line, pie, scatter, 3D column

Tech Stack & Tools

Frontend
- React.js
- Redux Toolkit
- Chart.js
- Three.js
- Tailwind CSS

Backend
- Node.js
- Express.js
- MongoDB
- Multer (file upload)
- SheetJS/xlsx

Optional
OpenAI or similar API for AI summaries

Tools
Postman, Git/GitHub, Cloudinary (if needed for file storage)

References
- Sheet JS (https://sheetjs.com/)
- Chart.js (https://www.chartjs.org/)
- Three.js (https://threejs.org/)
- YouTube: JavaScript Mastery, Codevolution, Fireship
- Sample inspiration app: https://chart-studio.plotly.com/

Development Timeline

week 1 — Project setup, user/admin authentication, dashboard layout.

week 2 — File upload setup, Excel parsing logic, store structured data in MongoDB.

week 3 — Chart rendering with Chart.js & Three.js, allow user to select axes dynamically.

week 4 — Save analysis history, enable download feature, integrate AI API for summaries.

week 5 — Admin panel, testing & bug fixes, deployment (Render for backend, Netlify for frontend).

ZIDIO DEVELOPMENT
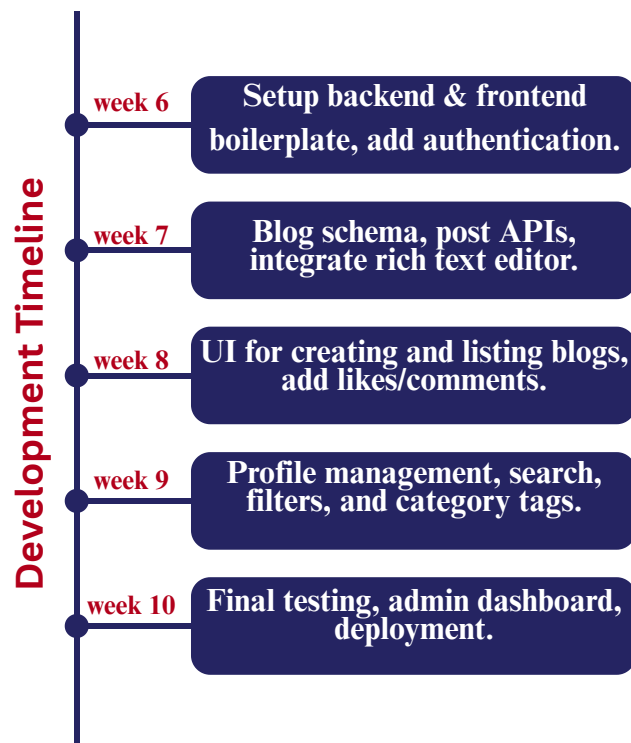
Project 2: Blogging Platform

Description:

A blogging application where users can create, edit, delete and view blogs. Includes rich text editor, likes, comments, and profile management.

User Authentication (JWT based)

Create/Edit/Delete blogs with rich text editor (Quill.js or similar)

Comments & Likes system

Key Features

User profile with authored blogs

Search and category filtering

Admin dashboard to manage posts and users

Tech Stack & Tools

Frontend:

React.js, Redux, Tailwind CSS, React Quill

Backend:

Node.js, Express.js, MongoDB

Tools:

GitHub, Cloudinary (for image upload), Postman

ZIDIO
DEVELOPMENT

Development Timeline

Week	Task
week 6	Setup backend & frontend boilerplate, add authentication.
week 7	Blog schema, post APIs, integrate rich text editor.
week 8	UI for creating and listing blogs, add likes/comments.
week 9	Profile management, search, filters, and category tags.
week 10	Final testing, admin dashboard, deployment.

References

YouTube: **Clever Programmer (https://www.youtube.com/c/CleverProgrammer)**

Codevolution (https://www.youtube.com/c/Codevolution)

Sample app: https://medium.com/

ZIDIO
DEVELOPMENT