**Login View**

A simple web application to distribute the task to the Created Agents (using **MERN** stack)

**Features**

* Upload **CSV, XLSX, or XLS** files.
* Validate files for required fields (**FirstName, Phone, Notes**).
* Display uploaded data in a structured table.
* Assign records to agents.
* Clear all records with a single click.
* Show appropriate error messages for invalid files or missing fields.

**Tech Stack**

* **Frontend:** React.js
* **Backend:** Node.js, Express.js
* **Database:** MongoDB
* **File Parsing:** PapaParse (CSV) and XLSX (Excel)
* **HTTP Client:** Axios

**⚙️ Setup and Installation**

**Step 1: Clone the Repository**

git clone https://github.com/your-repo/file-upload-management.git

cd file-upload-management

**Step 2: Install Dependencies**

**Backend**

cd backend

npm install

**Frontend**

cd ../frontend

npm install

**Environment Variables**

Create a .env file in the backend folder and add the following:

PORT=5000

MONGO\_URI=mongodb://localhost:27017/projectdb

JWT\_SECRET = (the token which is generated)

**Running the Application**

**Start Backend**

cd backend

nodemon server.js

**Start Frontend**

cd frontend/react-app

npm run dev

* Frontend URL: http://localhost:5173
* Backend URL: http://localhost:5000

**File Upload Requirements**

* Allowed File Types: .csv, .xlsx, .xls
* Required Columns:
  + FirstName → Text
  + Phone → Number
  + Notes → Text

**Invalid files or missing fields will trigger an error alert.**

**Project Walkthrough**

**1. Login using the admin credentials**

* Enter the Email and Password for the admin
* This will direct to the Admin dashboard

**2. Admin dashboard**

* The dashboard of the admin consists of Admin details
* There are 2 functionalities **Agent management and Upload CSV** and a logout button

**3. Agent management**

* The agents can be created and managed using the input fields
* The CRUD operations can be performed in this page

**4. View Records**

* Records will be displayed in a table format with columns:
  + **FirstName**
  + **Phone**
  + **Notes**
  + **Assigned Agent**

**5. Clear Records**

* Click **Clear Records** to delete all records from the database.
* A confirmation popup will appear before deletion.

**Error Handling**

* **Invalid File Type:** Alerts if the uploaded file is not CSV, XLSX, or XLS.
* **Missing Fields:** Displays an error if required fields are missing.
* **Upload Failure:** Provides an alert if the file fails to upload.
* **Clear Records Error:** Alerts if the backend fails to delete records.

**Testing**

You can use the sample files provided in the /samples directory for testing:

* **Valid CSV File** (valid\_sample.csv)
* **Valid XLSX File** (valid\_sample.xlsx)
* **Valid XLS File** (valid\_sample.xls)
* **Invalid File** (invalid\_sample.csv) → Missing columns

**Contact**

* **Developer:** Ratheesh Kumar
* **GitHub:** [ratheesh-kmr](https://github.com/ratheesh-kmr)
* **LinkedIn:** [Ratheesh Kumar](https://www.linkedin.com/in/ratheesh-kumar-99b89b259/)
* **Email:** [ratheeshkmr2024@gmail.com](mailto:ratheeshkmr2024@gmail.com)