

## **Machine Test for MERN Stack Developer**

### **Objective**

To implement a basic application using the MERN stack with the following features:

1. Admin User Login
2. Agent Creation & Management
3. Uploading and Distributing Lists

### **Tasks and Features**

#### **1. User Login**

- Create a login form with the following fields:
- Email
- Password
- Authenticate the user by matching the credentials with the registered user data in the MongoDB database.
- Use JSON Web Tokens (JWT) for user authentication
- On successful login, redirect the user to the dashboard.
- On failure, display an appropriate error message.

#### **2. Add Agents**

- Create a feature to add agents.
- Each agent should have the following details:
- Name
- Email
- Mobile Number with country code
- Password

#### **3. Upload CSV and Distribute Lists**

- Create a functionality to upload a CSV file containing a list of
- FirstName – Text
- Phone - Number
- Notes – Text

- Add validation for file upload where it will accept only csv,xlsx and axls
- Validate the uploaded CSV to ensure it is in the correct format.
- Distribute the tasks or items equally among the 5 agents.
- For example, if there are 25 items in the CSV, each agent should receive 5 items.
- If the total number of items is not divisible by 5, distribute the remaining items sequentially.
- Save the distributed lists in the MongoDB database.
- Display the distributed lists for each agent on the frontend.

### **Technical Requirements**

1. Use MongoDB for the database.
2. Use Express.js and Node.js for the backend.
3. Use React.js / Next.js for the frontend.
4. Ensure proper validation and error handling.
5. Maintain clean and readable code with comments.
6. Provide a .env file for configuration (e.g., database connection string, JWT secret).
7. Provide instructions on how to run the application.

### **Deliverables**

1. Source code for the application.
2. A README file with setup and execution instructions.
3. A working video demonstration of the application with a link of the video hosted on Google Drive.

### **Evaluation Criteria**

1. Functionality: Does the application meet the requirements?
2. Code Quality: Is the code clean, readable, and well-documented?
3. Validation and Error Handling: Are edge cases handled effectively?
4. User Interface: Is the interface user-friendly?
5. Execution: Is the application easy to set up and run?

**Good luck!**