

## **TECHNOKART CONSULTANCY SERVICES**

### **Assignment: Ledger and Transactions Management System Using MERN Stack**

#### **Objective:**

Develop a web application using the MERN (MongoDB, Express, React, Node.js) stack to manage ledger entries and their corresponding transactions. Users should be able to create ledger entries, add transactions to these ledgers, view transactions for a selected ledger between specific dates, and generate a PDF report of the transactions.

#### **Requirements:**

##### **Functional Requirements:**

###### **1. Ledger Management:**

- Implement functionality to create a ledger entry with a ledger name.

###### **2. Transactions Management:**

- Add transactions for a ledger with the following details:
  - Ledger name (linked to the created ledger)
  - Transaction amount
  - Date of transaction
  - Type of transaction (Given or Taken)

###### **3. Dashboard:**

- Display a list of all created ledgers.
- Select a ledger to view its transactions.
- Filter transactions for the selected ledger between specific dates.

###### **4. PDF Generation:**

- Generate a PDF report of the transactions for a selected ledger within the specified date range.

##### **Non-Functional Requirements:**

###### **1. Data Security:**

- Ensure that data stored in MongoDB is encrypted and cannot be viewed by unauthorized users.

#### **Instructions:**

###### **1. Setup and Configuration:**

- Set up a MongoDB database.

- Initialize a Node.js project and install necessary dependencies (Express, Mongoose, etc.).
- Set up a React project for the front end.
- Implement API endpoints for managing ledgers and transactions.

## **2. Backend Implementation:**

- Create API endpoints to:
  - Create a new ledger.
  - Add transactions to a ledger.
  - Retrieve transactions for a ledger within specific dates.
  - Generate a PDF report of transactions.

## **3. Frontend Implementation:**

- Develop forms for creating ledgers and adding transactions.
- Create a dashboard to display all ledgers.
- Implement functionality to select a ledger and filter transactions by date.
- Integrate PDF generation functionality.

## **4. Data Security:**

- Implement encryption for data stored in MongoDB.
- Ensure that sensitive data is not exposed in plain text.

## **5. Testing and Validation:**

- Test the application to ensure all functionalities work as expected.
- Validate that data encryption is implemented correctly and securely.

## **Deliverables:**

1. Source code for the backend (Node.js/Express).
2. Source code for the frontend (React).
3. MongoDB database setup with encrypted data.
4. A README file with instructions on how to set up and run the project.
5. Screenshots or a demo video showing the application in action.

## **Submission:**

Submit your project repository link (GitHub, GitLab, etc.) along with any necessary documentation for setting up and running the application.