ADBMS

# Aim:

To implement CRUD (Create, Read, Update, Delete) operations on a Product database using Mongoose in a Node.js application.

# Tools Used:

* Hardware: Computer/Laptop with internet
* Software: Node.js, Express.js, MongoDB (local or Atlas), Mongoose (ODM library), Postman (for API testing), VS Code (IDE)

# Procedure:

1. Setup Environment: - Install Node.js and MongoDB. - Create a new project folder and initialize it using npm init -y. - Install dependencies: npm install express mongoose body-parser
2. Create Product Schema & Model: - Define schema for Product with fields: name, price, category, stock. - Use Mongoose to create a model.
3. Setup Express Server & MongoDB Connection: - Connect to MongoDB and use Express to create API routes.
4. Implement CRUD Operations: - Create (POST): Add new product. - Read (GET): Fetch all products / single product. - Update (PUT): Update product details by ID. - Delete (DELETE): Remove product by ID.
5. Run and Test Application: - Start server using: node server.js - Test APIs in Postman.

# Learning Outcomes:

* + Understand how to connect Node.js applications with MongoDB using Mongoose.
  + Define schemas and models for structured data.
  + Implement CRUD operations using RESTful API routes.
  + Test APIs using Postman.
  + Apply the concept of schema validation and error handling.

# Output:

Example API Calls & Responses:

## Create Product (POST /products)

Input:

{ "name": "Laptop", "price": 55000, "category": "Electronics", "stock": 10 }

Output:

{ "\_id": "66fcf88a12345abc12345678", "name": "Laptop", "price": 55000, "category": "Electronics",

"stock": 10 }

## Read All Products (GET /products)

[ { "\_id": "66fcf88a12345abc12345678", "name": "Laptop", "price": 55000, "category": "Electronics", "stock": 10 } ]

## Update Product (PUT /products/:id)

Input: { "price": 60000, "stock": 15 }

Output: { "\_id": "66fcf88a12345abc12345678", "name": "Laptop", "price": 60000, "category": "Electronics", "stock": 15 }

## Delete Product (DELETE /products/:id)

Output: { "message": "Product deleted successfully" }