MERN Full Stack Assignment- 1

1. User Registration System

Objective:
Build a full-stack user registration system that allows users to sign up and store their details in a database.

Requirements:

- **Frontend:** A registration form with fields for name, email, password, and confirm password.
- **Backend:** An API endpoint to handle registration requests and store user data in the database.
- **Database:** A `users` table to store user details (name, email, hashed password).
- **Validation:** Ensure email is unique and passwords match.

2. To-Do List Application

Objective:
Create a full-stack to-do list application where users can add, update, and delete tasks.

Requirements:

- **Frontend:** A task input form and a list to display tasks.
- **Backend:** API endpoints for adding, updating, and deleting tasks.
- **Database:** A `tasks` table to store task details (id, title, description, status).
- **Features:** Mark tasks as completed and filter tasks by status.

3. Blog Post System

Objective:
Develop a full-stack blog post system where users can create, read, update, and delete blog posts.

Requirements:

- **Frontend:** A form to create/edit blog posts and a page to display posts.

- **Backend:** API endpoints for CRUD operations on blog posts.
- **Database:** A `posts` table to store blog post details (id, title, content, author, date).
- **Features:** Add pagination and search functionality.

4. E-Commerce Product Listing

Objective:
Build a full-stack e-commerce product listing page with a shopping cart.

Requirements:

- **Frontend:** Display products with images, prices, and an "Add to Cart" button.
- **Backend:** API endpoints to fetch products and manage the shopping cart.
- **Database:** A `products` table to store product details (id, name, price, image URL).
- **Features:** Calculate the total price of items in the cart.

5. Weather App with API Integration

Objective:
Create a full-stack weather application that fetches weather data from a public API.

Requirements:

- **Frontend:** A form to input a city name and display weather details (temperature, humidity, etc.).
- **Backend:** An API endpoint to fetch weather data from a third-party API (e.g., OpenWeatherMap).
- **Database:** Store search history in a `search` table.
- **Features:** Display recent search history.

6. Online Quiz System

Objective:
Develop a full-stack online quiz system where users can take quizzes and view results.

Requirements:

- **Frontend:** Display quiz questions with multiple-choice options.
- **Backend:** API endpoints to fetch questions and submit answers.

- **Database:** A `questions` table to store quiz questions and answers.
- **Features:** Calculate and display the user's score.

7. Movie Review Platform

Objective:
Build a full-stack movie review platform where users can post and read reviews.

Requirements:

- **Frontend:** A form to submit reviews and a page to display reviews.
- **Backend:** API endpoints to submit and fetch reviews.
- **Database:** A `reviews` table to store review details (id, movie title, review, rating).
- **Features:** Allow users to upvote/downvote reviews.