Q1: Write a schema for poetry with entities of Poet and Data.

Q2: Write a controller for schema mentioned in Q1 and make following routes

1. Get All Poetries

2. Create a Poetry

Q3. Create a react app with two pages

1. Page one must contain one input box and one text area

2. Page two must render all poetries.

Q4. Create a state management for pages mentioned in Q2 and integrate API created in Q1 and Q2.

Q5. Define the following terms:

**1. MongoDB**

MongoDB is a widely used open-source NoSQL database management system that falls into the category of document-oriented databases. It is designed to provide a flexible, scalable, and high-performance solution for storing, retrieving, and managing data. MongoDB uses a document model, storing data in a format similar to JSON called BSON (Binary JSON). Each record in a MongoDB database is a document, and collections of documents form databases

**2. Tailwind CSS**

Tailwind CSS is a popular utility-first CSS framework that provides a set of pre-designed, low-level utility classes. It is designed to be highly customizable and aims to help developers build modern, responsive web interfaces quickly and efficiently.

**3. State of react application**

In the context of a React application, the "state" refers to an object that represents the current condition or data within a component. The state allows React components to manage and handle dynamic data, and when the state of a component changes, React automatically re-renders the component to reflect those changes in the user interface

**4. useEffect hook**

In React, the useEffect hook is part of the Hooks API introduced in React 16.8. It allows functional components to perform side effects in response to component lifecycle events, such as mounting, updating, and unmounting. Side effects can include things like data fetching, subscriptions, manual DOM manipulations, or any operation that needs to be performed after the component is rendered.

**5. useState Hook**

In React, the useState hook is part of the Hooks API introduced in React 16.8. It allows functional components to manage state, which enables the component to keep track of data that can change over time. Prior to the introduction of hooks, state management was primarily associated with class components, but hooks make it possible to use state in functional components as well.