Project Code with Short Explanations

3. vite.config.js

This Vite configuration sets up a project with React and Tailwind CSS plugins.

4. backend/index.js

The code sets up an Express server with middleware for CORS, JSON parsing, and request logging, connects to a MongoDB database, and mounts authentication and plan routes. The server listens on port 5000.

1. eslint.config.js

This ESLint configuration sets up rules and plugins for JavaScript and React projects, including recommended settings, React hooks, and React refresh for Vite, while ignoring the `dist` directory and applying specific parsing and global options. It also enforces a rule to ignore unused variables that start with an uppercase letter or underscore.

2. index.html

This HTML code sets up a basic web page for a Vite + React application, including meta tags and a script that loads the main React component.

5. backend/config/db.js

The code defines an asynchronous function `connectDB` that connects to a MongoDB database using Mongoose and logs the result. It exports this function as the default export.

6. backend/models/User.js

The code defines a Mongoose schema for a User model with fields for name, email, and password, and exports the model.

8. backend/routes/plan.js

This code sets up an Express route that uses the Google Generative AI API to generate a personalized fitness and diet plan for an Indian user based on their details. The plan includes a workout routine, meal plan, and lifestyle tips, formatted in Markdown.

7. backend/routes/auth.js

This code sets up an Express router with routes for user registration and login, handling validation, user existence checks, and error responses.

9. src/App.css

The code styles a webpage with a centered layout, logos with hover effects and animation, and additional styling for cards and text.

10. src/App.jsx

The code sets up a React application with routing using `react-router-dom`, defining routes for an authentication form, a user details form, and a dashboard.

11. src/index.css

The code sets up global styles for a web page, including font, color, and layout, with specific styles for links, buttons, and headings, and includes dark and light mode support.

15. src/pages/Dashboard.jsx

The Dashboard component displays a personalized fitness plan, including workout routines, meal plans, and lifestyle tips, with a sidebar for navigation and dynamic content loading based on user interaction.

14. src/components/Sidebar.jsx

The code defines a `Sidebar` component for a web application, featuring navigation links and a responsive mobile design that includes an overlay and sidebar toggle functionality. It uses mock dependencies and icons for demonstration.

13. src/components/AuthForm.jsx

This component renders a signup/login form that toggles between signup and login modes, handles form input changes, and submits the form data to the appropriate API endpoint. Upon successful submission, it navigates to the '/form' page.

12. src/main.jsx

The code sets up a React application, rendering the `App` component inside a `StrictMode` wrapper and applying Tailwind CSS styles.

16. src/pages/UserDetailsForm.jsx

This React component renders a form for users to input their details and generate a personalized plan. It handles form submission, displays the generated plan, and allows users to finalize or edit the plan.