💻 CodeX — MERN Online Judge Platform

CodeX is a modern, feature-rich **DSA (Data Structures & Algorithms) practice platform** built with the **MERN stack**, Docker, Redux, and other cutting-edge technologies.
It allows users to solve coding problems, run and submit code, track progress, view leaderboards, and much more — all with an intuitive interface and robust backend architecture.

🚀 Features

🧑‍💻 For Users

- 🔍 **DSA Problem Bank**
  - Browse, search, sort, and filter problems by difficulty (`Easy`, `Medium`, `Hard`) and status (`Solved`, `Unsolved`).
- 📝 **Code Editor**
  - Write, run (with custom input), and submit code.
  - Code is cached — remains intact even after page reload.
- ⏱️ **Submission Metrics**
  - Get **time and space complexity** results upon submission.
- 📊 **Submissions Dashboard**
  - View submission history for each problem and see all submissions on your profile.
- 🏆 **Leaderboard**
  - Compete with others based on solved problems, accuracy, and a calculated rating.
- 👤 **Profile Management**
  - Update or delete your profile, and view problem-solving distribution by difficulty.

👨‍💼 For Admin

- 🧩 **Admin Dashboard**
  - Create, update, and delete DSA problems.
  - View and respond to `Contact Me` messages sent by users.

🧰 Tech Stack & Architecture

🌍 Frontend

- React + Redux Toolkit + Redux Persist
- React Router DOM
- Framer Motion, Recharts, SweetAlert2
- TailwindCSS + DaisyUI
- Vite for blazing fast builds
- Lazy Loading & Code Splitting
- Modular, maintainable components

🖥️ Backend 1 (CRUD Server)

- Node.js + Express
- MongoDB + Mongoose
- JWT-based Authentication (cookies)
- Problem CRUD APIs, Leaderboard, Profile management, Contact messages

🖥️ Backend 2 (Compiler Server)

- Node.js + Express
- Docker-based code execution
- Handles both `Run` and `Submit` requests
- Calculates time and space complexity safely

📦 Project Structure

CodeX/
├── backend/ # CRUD server
| └── package.json
| └── .env
├── compiler/ # Docker-based code runner
| └── package.json
| └── .env
├── frontend/ # React frontend
| └── package.json
| └── .env
└── README.md

Each service runs independently and communicates via REST APIs.

🧪 Installation & Setup

Prerequisites

✅ Node.js ≥ 18
✅ MongoDB (local or Atlas)
✅ Docker

📄 Environment Variables

Backend (CRUD) `.env`:

```
PORT=5005
ORIGIN_URL=http://localhost:5173
COMPILER_BASE_URL=http://localhost:5008
```

```
MONGO_URI=mongodb+srv://<username>:
<password>@<cluster>.mongodb.net/<dbname>?
retryWrites=true&w=majority&appName=<appName>

JWT_SECRET_KEY=<your_jwt_secret_here>
JWT_EXPIRES_IN=7d
NODE_ENV=development
```

Backend (COMPILER) `.env`:

```
PORT=5008
ORIGIN_URL=http://localhost:5173

MONGO_URI=mongodb+srv://<username>:
<password>@<cluster>.mongodb.net/<dbname>?
retryWrites=true&w=majority&appName=<appName>
```

Frontend `.env`:

```
VITE_API_URL=http://localhost:5005/api/v1
VITE_COMPILER_URL=http://localhost:5008/
```

Clone the repository

- git clone https://github.com/AyushGupta3900/SummerProject.git
- cd SummerProject

Setup Backend (CRUD)

- cd backend
- cp .env.example .env # and fill in your credentials
- npm install
- npm run dev

Setup Frontend

- cd ../frontend
- cp .env.example .env # and fill in your URLs
- npm install
- npm run dev

Setup Compiler (Docker-based)

- cd ../compiler

- cp .env.example .env # and fill in your credentials
- docker stop $(docker ps -q) || true
- docker rm $(docker ps -aq) || true
- docker image prune -a -f
- docker build --no-cache -t compiler-server .
- docker run -p 5008:5008 compiler-server