CodeZilla 2025 – Website Development Competition Task

**Scenario:**

You are a developer team for a startup named **“EcoMind”**, which promotes **mental wellness through nature therapy**. They want a creative, AI-powered website that shares nature inspiration, engages users interactively, and collects wellness feedback from visitors.

**💡 Task Overview**

The task has **three parts**, all of which can be done **parallelly** by different group members.

**🔹 Part 1: Frontend Website Design – “Nature Therapy Portal”**

Create a **4-page website** using **HTML, CSS, JavaScript, and Bootstrap**:

**📄 Page 1: Home Page**

* Title: *EcoMind – Nature Heals*
* Navigation Bar linking all pages
* Hero Banner with intro to nature therapy
* Grid of calming quotes or nature therapy themes (use cards)
* Responsive design must work on desktop and mobile

**📄 Page 2: Explore Page**

* Activity cards like “Forest Bathing”, “Ocean Sounds”, “Stargazing”
* Each card shows image + a Bootstrap modal with more description

**📄 Page 3: AI Companion Page**

* Input field for user mood or name
* “Get Therapy Quote” button
* On click, connect to **free AI API** to fetch quote or calming suggestion
* Display result dynamically on the same page

**📄 Page 4: Wellness Feedback Page**

* Feedback Form:
  + Name (text)
  + Email (email)
  + Mood (dropdown: Happy, Sad, Calm, Stressed, Energetic)
  + Comments (textarea)
* On submission:
  + Convert form data to **JSON format** using JavaScript
  + Save it to **localStorage** (simulate data saving)
  + Show a confirmation message
  + Optionally display submitted data below the form

**🔹 Part 2: AI Integration – “AI Nature Companion”**

Use any **free Gen AI API** (e.g. ZenQuotes, HuggingFace) to:

* Generate **custom calming advice, quote, or image** based on user input
* Fetch and display data using JavaScript fetch()

✅ Bonus: Add loading animation or error handling  
✅ Bonus: Let users regenerate a new quote on button click

**🔹 Part 3: JSON Form Handling**

This is now embedded in **Page 4** of the website (Wellness Feedback Page):

* Use localStorage.setItem() or similar approach to store the form data in browser
* Structure the data as clean JSON
* Example:

json

CopyEdit

{

"name": "Ali",

"email": "ali@gmail.com",

"mood": "Calm",

"comments": "Loved the forest activity!"

}

🔒 **Note**: Simulated backend using frontend only — no server interaction required

Competition Task: "SkillSprint – AI-Powered Career Portal

**Team Size: 2 to 3 students**

**⏱️ Duration: 2 hours**

**💻 Technology Choice (any one):**

* PHP + MySQL
* ASP.NET + SQL Server
* MERN Stack (MongoDB, Express, React, Node.js)

**🔖 Scenario**

A non-profit organization wants to launch a mini web platform called **SkillSprint** to help fresh graduates discover career paths and prepare for interviews with the help of AI. Your team is hired to develop a basic version of this platform.

**📌 Project Parts**

**🔹 Part 1: Core Website Functionality (Full-Stack)**

Develop the following features:

1. **Home Page**
   * Welcome banner
   * Navigation menu to all other pages
2. **Career Paths Page**
   * A dynamic page that shows different career paths (e.g., Web Developer, Data Analyst, Designer).
   * Data should come from the database.
   * Each card includes a job title, short description, and salary range.
3. **Admin Panel**
   * Admin can log in.
   * Add/edit/delete career path entries using forms.
4. **Contact / Feedback Page**
   * A simple form (Name, Email, Message)
   * On submission, the data should be saved in the database (MySQL / SQL Server / MongoDB).

**🔹 Part 2: AI API Integration (Gen AI Feature)**

Create a separate page or a modal that allows users to:

**"Ask Career Coach" – Gen AI Integration**

* Input field: "Ask me anything about this career..."
* Send user query to any **free Gen AI API** (e.g., OpenAI free model, HuggingFace inference API, Cohere, etc.).
* Display AI response nicely (e.g., cards, chat bubbles).

🧠 **Examples** of user prompts:

* “What skills do I need to become a frontend developer?”
* “Suggest a study plan to become a data analyst.”

**Deliverables:**

* Complete working web app with all pages.
* Basic authentication for admin panel (hardcoded or dynamic).
* Proper integration of AI API in one page.
* Team must explain what each member contributed.

**Summary ER Diagram (Conceptual)**

text

CopyEdit

[admin\_users] [career\_paths] [feedback] [ai\_queries]

| | | |

|---------------- |------------------ |------------------ |----------------

id (PK) id (PK) id (PK) id (PK)

username title name user\_question

password description email ai\_response

email salary\_range message asked\_at

created\_at created\_at submitted\_at

**Competition Task Title: "FitBuddy – Your Smart Fitness Companion"**

**Scenario:**

In a fast-paced world, many people struggle to maintain a healthy lifestyle. Your team is tasked with building a **fitness companion mobile app** that helps users stay on track with their daily fitness goals, track workouts, and get personalized motivational content.

**🎯 Task Objective:**

Develop a basic version of the **FitBuddy mobile app** using Flutter in 2 hours. The app should have multiple screens and must store basic user inputs locally (using Provider or shared\_preferences for state/data persistence is allowed).

**🔧 Core Features to Implement:**

**🔹 1. Home Screen (Dashboard)**

* Display greeting ("Hello [User] 👋")
* Show current date and time
* Display motivational quote (fetched from a free **AI quote/motivation API**) like:
  + https://zenquotes.io/api/today
  + https://type.fit/api/quotes

**🔹 2. Workout Tracker Page**

* Allow user to:
  + Select workout type (Running, Yoga, Cycling, etc.)
  + Enter duration (in minutes)
  + Log workout
* Show list of today’s logged workouts with duration

**🔹 3. Daily Goal Progress Page**

* Set a daily workout goal (in minutes)
* Show progress bar indicating how much of the goal is completed
* Optional: Save goal using shared\_preferences

**🔹 4. Profile Page**

* Basic profile form: name, age, weight, gender
* Save form data locally (in JSON or via Flutter local storage)
* Display stored data when revisiting the screen