Web-based Intelligent Career Advisor System (Rule-Based AI)

What is it?

A web app that helps students or job seekers get personalized career advice based on their interests, skills, education, and preferences — using **rule-based AI logic** (if-then rules) to suggest suitable career paths, courses, or skills to develop.

Why is it innovative and useful?

- Solves a **real-world problem**: Many students are confused about which career to pursue; this system helps guide them using logical, explainable AI rules.
- Addresses **education and local industry needs** by helping users make informed career choices that match market demand.
- Uses AI but **simple rule-based system** instead of complex ML, making it easier to implement and explain.
- Integrates **database, programming, OOP, and system/network** concepts.

How does it connect to your main courses?

Course	How it applies
Introduction to IT	Understanding basic IT concepts and system design
Programming Fundamentals	Writing JavaScript for frontend logic and rule implementation
OOP	Designing the system with objects like User, Career, Rule
Database Systems	Storing user data, career options, rules in a database
System and Network Administration	Hosting the app on a server, managing security and access
Database Administration	Creating, querying, and maintaining career and user data
Computer Networks	Implementing secure data transmission, authentication

---

Core Features:

1. **User Profile Creation:**
   Collect data about the user's interests, education, skills, and preferences through a questionnaire.
2. **Rule-Based AI Engine:**

    - Uses **if-then rules** like:
      If user likes math and problem-solving, then suggest careers in engineering or data science.
      If user prefers creative tasks, suggest careers in design or marketing.
    - Rules are stored in the database and executed in the backend.
3. **Career Suggestions:**
   Provides ranked career paths with explanations about why they fit the user.
4. **Skill & Course Recommendations:**
   Suggests skills or online courses to develop for chosen careers.
5. **Admin Panel:**
   Admins can add/update career options, rules, and skills.
6. **Responsive UI:**
   Built with HTML, CSS, Bootstrap, and JavaScript for easy navigation on all devices.
7. **Security:**
   Authentication for users and admins; secure data handling.

How to implement:

- **Frontend:**
    - Use HTML/CSS/Bootstrap for responsive design.
    - JavaScript for dynamic forms and displaying suggestions.
- **Backend:**
    - Use Node.js (or PHP if preferred) for handling logic and database queries.
    - Implement the rule engine in JavaScript as a set of functions that evaluate user input against rules.
- **Database:**
    - Store users, careers, rules, and skills in MySQL or SQLite.
    - Design tables for: Users, Careers, Skills, Rules (with conditions and outcomes).
- **Hosting & Network:**
    - Host on local server or free cloud services (like Heroku).
    - Use HTTPS for secure communication.

Why this project fits your guidelines:

- **Real-world problem:** Helps students & job seekers choose careers.
- **Applicable to local education & industry:** Can be customized for your region's job market.
- **Integration of courses:** Combines database, programming, networking, AI concepts.
- **Not difficult or too narrow:** Rule-based AI is simple yet powerful.
- **Novel:** Personalized career advice with explainable AI rules, not just static info.
- **Resource feasible:** Uses available languages and tools you know.

Extra ideas to enhance:

- Add **feedback loop** where users rate advice quality to improve rules.
- Include **local job market data** integration via API to suggest trending careers.
- Use **charts** or visualizations for career paths with JavaScript libraries.
