You're right — let's **fully integrate your feature set** into the **AIspire web app roadmap**, adapted for the **Gemini API** and **MERN stack**. Here's the **complete, improvised version** including all the missing features:

**🚀 AIspire: Personalized Career Mentor**

**Powered by Google Gemini | MERN Stack Web App**

**🧠 Key Features Summary**

**1. User Profile & Onboarding**

* Name, email, age, education level
* Career interests, preferred industries
* Self-assessed skill proficiency
* Stored in MongoDB

**2. AI-Powered Career Guidance (via Gemini)**

* 🎯 Input: User profile & preferences
* ✅ Output:
  + Top 3 career paths
  + Required skills for each
  + Personalized course roadmap
  + Suggested job roles
  + Interview prep tips & insights

**3. Skill Gap Analysis**

* Compares user's skills to target roles
* Highlights missing skills
* Suggests ways to acquire them

**4. Course & Resource Recommendations**

* Integrate:
  + ✅ Static DB (for MVP)
  + 🔄 Later: APIs like Coursera, Udemy, edX, YouTube
* Filters: Topic, difficulty, cost, platform

**5. Job Path Explorer**

* Explore roles, qualifications, and salary range
* (Later) Connect to APIs like LinkedIn or Indeed
* Career path visual flow

**6. Progress Tracking**

* Track learned skills, completed courses, goals
* Users can bookmark/save recommendations

**7. Chat-Based AI Mentor (Gemini)**

* Conversational UI to:
  + Ask career/skill questions
  + Get real-time suggestions
  + Receive actionable plans

**8. Bookmarks & History**

* Save recommendations
* View past AI chats, completed actions

**9. Analytics Dashboard (Admin or Firebase Analytics)**

* Track usage
* Popular career paths
* Drop-off insights (onboarding → chat → recommendations)

**10. Security & Compliance**

* 🔐 JWT Auth + bcrypt
* 🔐 Encrypted user data (interests/goals)
* 🛡️ Gemini data usage disclaimer
* ✅ GDPR/CCPA-ready user controls

**📊 Database Schema Summary (MongoDB/Mongoose)**

**🧾 User.js**

{

email, passwordHash, name, age, educationLevel,

interests, careerGoals, currentSkills,

createdAt, updatedAt

}

**📂 CareerRecommendation.js**

{

userId, careerPath, requiredSkills,

reasoning, actionPlan, recommendedByAI, createdAt

}

**📂 SkillGap.js**

{

userId, targetCareer, userSkills,

missingSkills, courseSuggestions, createdAt

}

**📂 Course.js**

{

title, platform, url, tags, difficulty

}

**📂 Progress.js**

{

userId, type: 'course' | 'skill' | 'goal',

title, status: 'completed' | 'in-progress', completionDate

}

**🧩 System Architecture (Updated for Gemini)**

**📱 Frontend (React)**

* Onboarding → Dashboard → Chat → Tracker
* Axios for API calls
* Protected routes (JWT token)

**⚙️ Backend (Express + Gemini)**

* /api/auth – signup/login
* /api/chat/mentor – send prompt to Gemini
* /api/user/recommendations – fetch/save
* /api/user/progress – track saved/completed items

**🔌 Gemini Integration (via REST API)**

* Endpoint: gemini-pro:generateContent
* Prompt format:

You are an expert AI career mentor. Given:

- Age: 22

- Education: B.Tech in CS

- Skills: Python, SQL

- Goals: Data Scientist

Suggest 3 career paths, required skills, free/paid resources, and action plan.

**🧪 MVP Roadmap**

**✅ Phase 1: MVP (Launchable in Weeks)**

* Profile onboarding + login
* Gemini-powered chat-based mentor
* Static course + career recommendations
* Skill gap display
* Basic progress tracking

**🚀 Phase 2: Enhancements**

* Dynamic course API integrations
* Job board integration
* User analytics
* Gamification (XP, badges)
* Resume analyzer + mock interviews

**💻 Want to Start Building?**

I can scaffold the **React frontend**, **Express backend**, and **Gemini connection** right now.

Would you like:

1. 🌐 Frontend starter with auth + dashboard?
2. 🔧 Backend setup (Mongo + Express + Gemini)?
3. 📦 Course/job dummy data set for MVP?
4. 🎨 Figma layout for UI?

Let me know where you want to begin, and I’ll generate the first full code module.