

Career Guidance ChatBot – A Career Mapping Tool for Girls in Rural Areas

1. Title

Career Guidance ChatBot: Creating a Career Mapping Tool for Girls in Rural Areas

2. Problem Statement

In rural areas, girls often face significant barriers in exploring and pursuing career opportunities. Some common challenges include:

- Lack of access to career counseling or guidance teachers.
- Limited awareness of modern career opportunities.
- Social and economic constraints that restrict decision-making.
- Absence of resources and motivational support.



Due to these factors, many girls are unable to align their **skills, interests, and education** with suitable career paths. There is a clear need for a **digital, AI-powered solution** that can help bridge this gap and empower rural girls to make informed career choices.

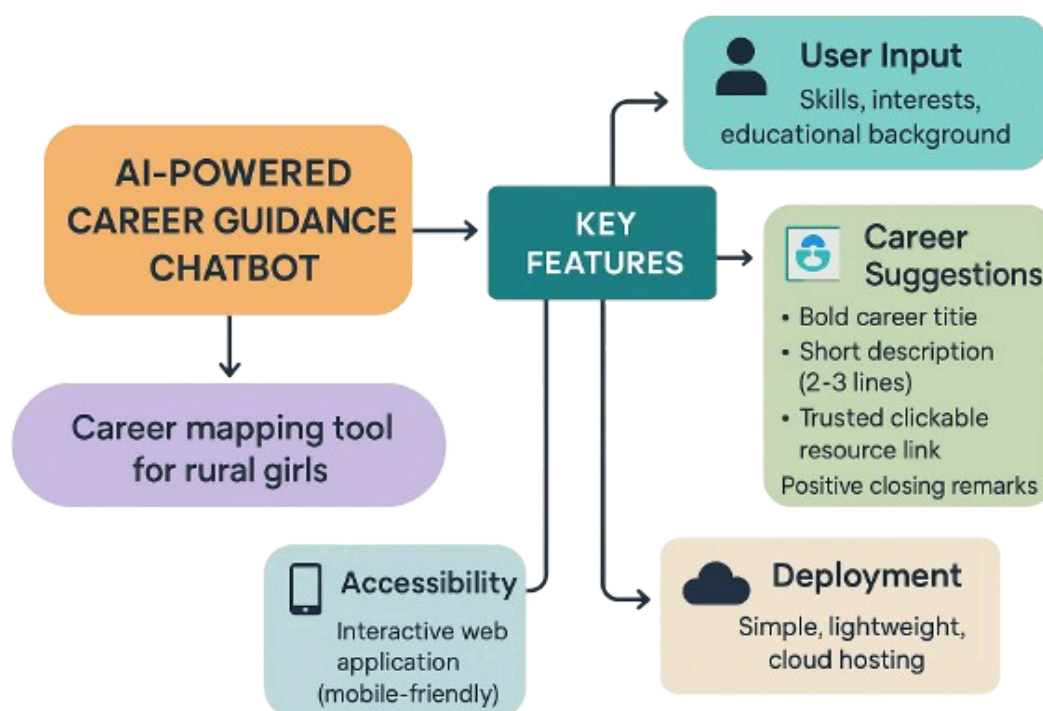


3. Proposed Solution

We propose an **AI-powered Career Guidance ChatBot**, designed as a **career mapping tool** for rural girls.

Key Features:

- Takes **user input** about their skills, interests, and educational background.
- Uses **Google Gemini API (Generative AI)** to generate personalized career suggestions.
- Each suggestion contains:
 - A **bold career title**
 - A short description (2–3 lines) of why it's suitable
 - A **trusted clickable resource link**
- Motivates the user with positive closing remarks.
- Works as an **interactive web application** accessible on mobile devices.
- Simple, lightweight, and deployable via **Hosting on Cloud**.



4. Tools and Technology Used

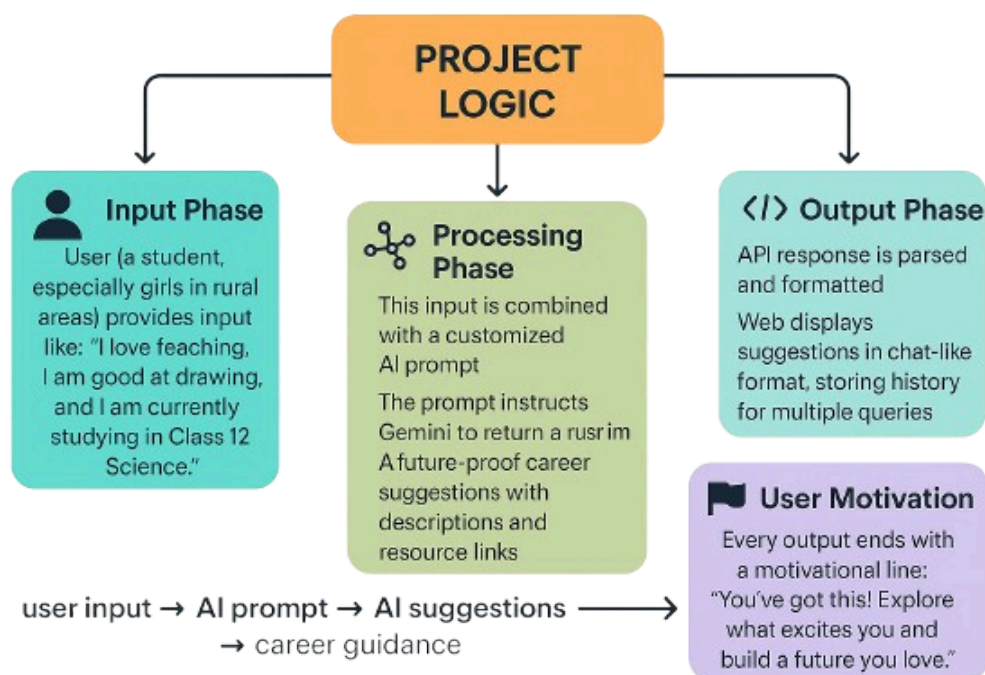
Component	Technology/Tool	Purpose
Programming Language	Python	Core logic and API integration
Web Framework	Streamlit	To create interactive web app UI
AI Model	Google Gemini API	To generate career suggestions
Libraries	requests, json, os, dotenv	Handle API calls, JSON parsing, environment variables
Deployment	Cloud	To host the application online

5. Project Logic

The **logic** behind this project is:

1. **Input Phase**
 - User (a student, especially girls in rural areas) provides input like:
“I love teaching, I am good at drawing, and I am currently studying in Class 12 Science.”
2. **Processing Phase (AI Prompting)**
 - This input is combined with a **customized AI prompt**.
 - The prompt instructs Gemini to return **3 future-proof career suggestions** with descriptions and resource links.
3. **Output Phase**
 - API response is parsed and formatted.
 - Web displays suggestions in chat-like format, storing history for multiple queries.
4. **User Motivation**
 - Every output ends with a motivational line:
“You’ve got this! Explore what excites you and build a future you love.”

Thus, the project **maps user input → AI prompt → AI suggestions → career guidance**.



6. Step-by-Step Plan

Step 1: Setup

- Install Python (3.9 or above).

Install required libraries:

```
pip install streamlit requests python-dotenv
```

•

Clone the repository:

<https://codeberg.org/bhaskarjoshi/Career-Guidance-ChatBot.git>

Step 2: Gemini API Key

- Go to **Google AI Studio** → <https://aistudio.google.com>.
- Generate a new **Gemini API Key**.
- Store it securely in **Streamlit Secrets** or **.env** file:

Example **.streamlit/secrets.toml**:

```
GEMINI_API_KEY = "your_api_key_here"
```



Step 3: Core Logic (careerBot.py)

- Code file: [careerBot.py](#)
- Key function:

```
def get_career_guidance(user_input: str) -> str:
    headers = {"Content-Type": "application/json"}
    prompt = f"""
    You are a career guidance expert.
    Analyze the user's profile: {user_input}
    Suggest 3 suitable careers with:
    - Bold career title
    - 2–3 line description
    - Resource link
    End with a motivational line.
    """

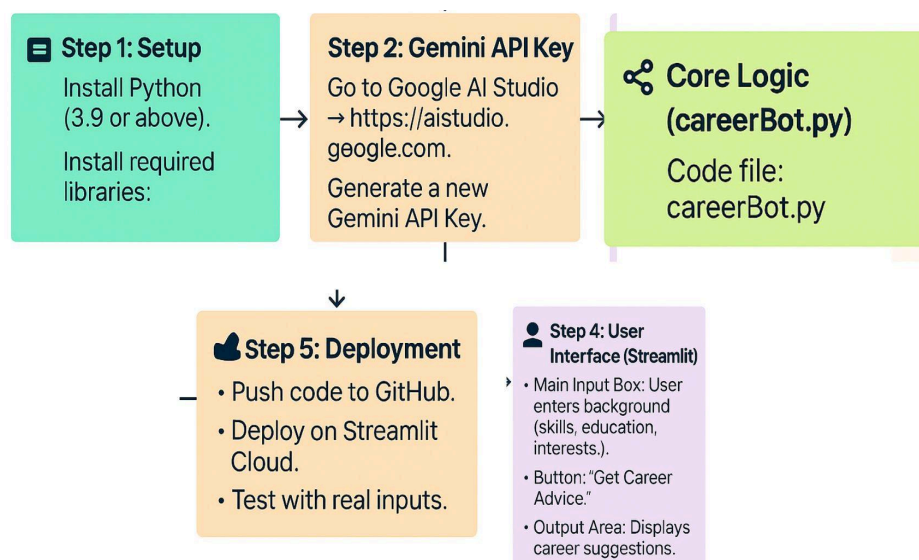
    data = {"contents": [{"parts": [{"text": prompt}]}]}
    response = requests.post(API_URL, headers=headers, data=json.dumps(data))
    return response.json()["candidates"][0]["content"]["parts"][0]["text"]
```

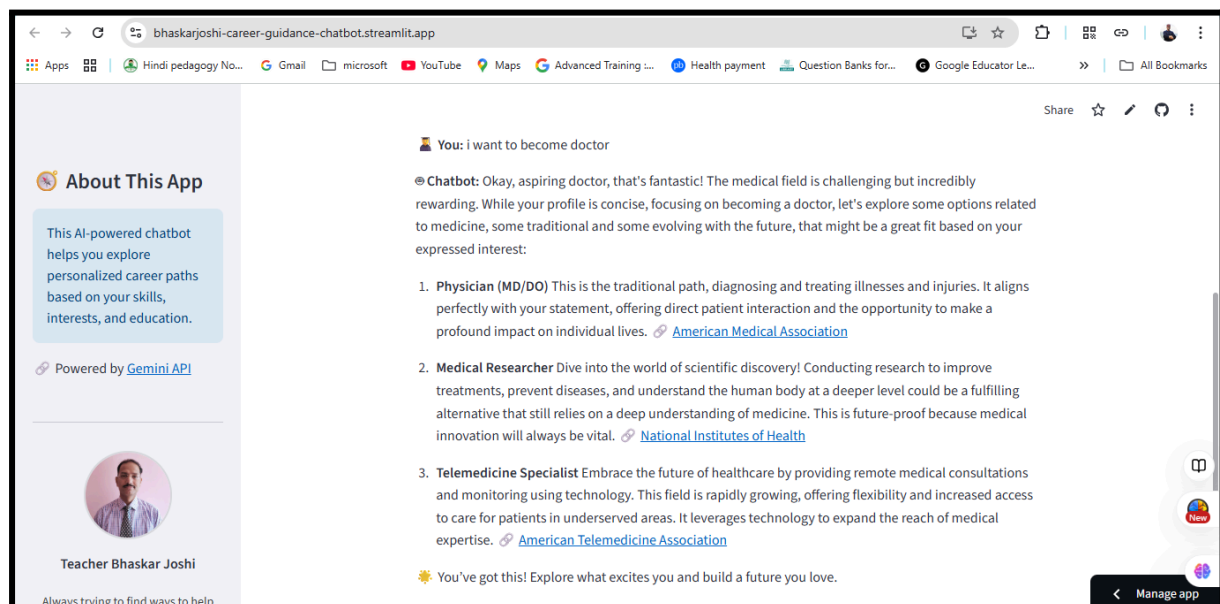
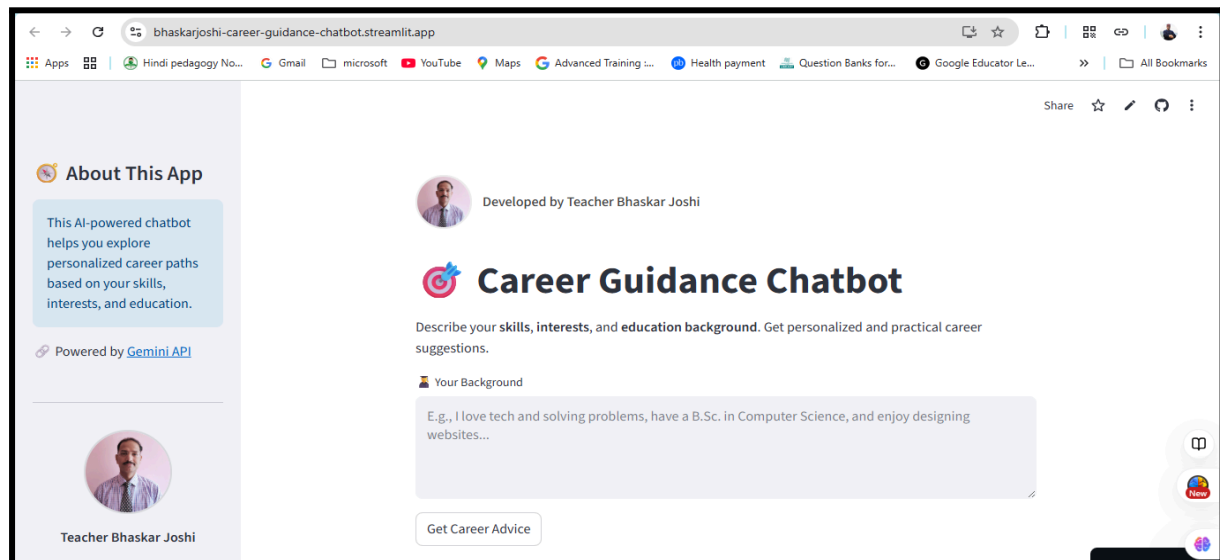
Step 4: User Interface (Streamlit)

- **Header:** Developed by Teacher Bhaskar Joshi (with circular profile photo).
- **Main Input Box:** User enters background (skills, education, interests).
- **Button:** “Get Career Advice”.
- **Output Area:** Displays career suggestions.
- **Sidebar:** About the app + developer bio with photo.

Step 5: Deployment

- Deploy on **Streamlit Cloud / Hugging Face**
<https://bhaskarjoshi-career-guidance-chatbot.streamlit.app/>
- Test with real inputs.





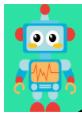
7. Learning Outcomes

By replicating this project, a student will learn:

1. Technical Skills

- Python programming and API integration.
- Web application development with Streamlit.
- Using environment variables and secrets for security.

2. AI/ML Understanding



- Prompt engineering for AI models.
- How AI can solve real-world problems.

3. Design & Usability

- Building an interactive, user-friendly web app.
- Using images, markdown, and formatting in Streamlit.

4. Project Development Skills

- GitHub version control.
- Cloud deployment.
- Testing and debugging.

5. Social Impact Awareness

- Linking technology with **empowerment of rural girls**.
- Understanding how digital tools can contribute to **SDGs**.

8. Link to Hackathon Theme / SDGs

★ I want to learn,
but what
should I learn?

LINK TO HACKATHON THEME

- SDG 4: Quality Education**
→ Career awareness for girls.
- SDG 5: Gender Equality**
→ Equal opportunity in education and career mapping.
- SDG 10: Reduced Inequalities** → Support for marginalized rural



This project directly aligns with **Hackathon themes** and **Sustainable Development Goals (SDGs)**:

- **SDG 4: Quality Education** → Career awareness for girls.
- **SDG 5: Gender Equality** → Equal opportunity in education and career mapping.
- **SDG 10: Reduced Inequalities** → Support for marginalized rural communities.

Impact:

- Encourages girls to dream beyond traditional roles.
- Provides them with resources and guidance.
- Acts as a low-cost, scalable career counseling solution.

9. How Can Students Replicate This Project?

Any student—whether from a city or a remote village—can build this meaningful AI-powered project in just six simple steps. First, start by learning the basics of **Python** and **Streamlit**, two beginner-friendly tools that make it easy to build interactive web apps. Next, head to [Google AI Studio](#) and generate your **Gemini API key**, which will power the chatbot's career suggestions using generative AI. Once you have your key, **clone the codeberg repository** using the command:

<https://codeberg.org/bhaskarjoshi/Career-Guidance-ChatBot.git>

Then navigate into the project folder and **install the required dependencies** with:

```
pip install -r requirements.txt
```

Now you're ready to **run the project locally** using:

```
streamlit run careerBot.py
```

From here, the fun begins! You can **customize the chatbot** by adding your own profile photo, updating the developer bio, and even modifying the prompt logic to serve different user groups—like engineering students, tribal youth, or primary school learners. This flexibility makes the project deeply personal and socially impactful.

Finally, you can **deploy your chatbot for free** using platforms like **Streamlit Cloud**, **Render**, **Vercel**, or **GitHub Pages** (for static content). These platforms offer generous free tiers, making it easy for students to showcase their work to the world without spending a rupee.

10. Video link YouTube

video link <https://youtu.be/YpBS6TTYAh4>