Deployment Instructions for Anna AI Coach (Flask Application)

This document provides general instructions for deploying the Anna AI Coach Flask application to a cloud hosting service. The exact steps may vary slightly depending on your chosen platform (e.g., Heroku, AWS Elastic Beanstalk, Google App Engine, Vercel, Render).

Prerequisites

1. **Python 3.8+**: Ensure your deployment environment supports Python 3.8 or newer.
2. **Git**: For version control and deployment.
3. **Cloud Provider Account**: An account with your preferred cloud hosting provider.
4. **OpenAI API Key**: Set this as an environment variable (`OPENAI_API_KEY`) in your deployment environment.

General Deployment Steps

1. Prepare Your Project

Ensure your project structure is clean and includes all necessary files:

Plain Text

```
anna_coach_new/
├── src/
│   ├── main.py             # Main Flask application logic
│   ├── templates/
│   │   └── index.html      # HTML template
│   └── static/
│       └── style.css       # CSS styling
├── requirements.txt        # Python dependencies
├── knowledge_base.md       # Knowledge base for the AI coach
└── .gitignore              # (Optional) Files to ignore for Git
```

- `main.py` is the entry point for the Flask application.
- `requirements.txt` lists all Python packages needed.
- `knowledge_base.md` is read by the application to provide context.

2. Install Dependencies Locally (Optional, for testing)

It's good practice to test your application locally before deploying.

Bash

```
cd anna_coach_new
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
export OPENAI_API_KEY='your_openai_api_key_here' # Replace with your actual
key
python src/main.py
```

Then, open `http://127.0.0.1:5000` in your browser.

3. Choose a Hosting Provider and Follow Their Deployment Guide

Below are general guidelines for common platforms. Always refer to the official documentation of your chosen provider for the most up-to-date and specific instructions.

Heroku (Example)

Heroku is a popular platform for deploying Flask applications.

1. **Create a `Procfile`**: In the root of your `anna_coach_new` directory, create a file named `Procfile` (no extension) with the following content:
2. **Add `gunicorn` to `requirements.txt`**: If not already present, add `gunicorn` to your `requirements.txt` file:
3. **Initialize Git and Push to Heroku**:
4. **Set Environment Variables**: Set your `OPENAI_API_KEY` on Heroku:
5. **Open Your App**:

Other Platforms (AWS, Google Cloud, Vercel, Render)

- **AWS Elastic Beanstalk**: Provides an easy way to deploy and scale Flask applications. You'll typically use the AWS CLI or console to upload your project and configure the environment.
- **Google App Engine**: Similar to Elastic Beanstalk, it's a Platform as a Service (PaaS) that handles infrastructure. You'll need an `app.yaml` file to configure your application.
- **Vercel/Render**: These platforms are excellent for deploying web applications. They often integrate directly with Git repositories and automatically detect Flask applications. You'll typically link your GitHub repository and configure environment variables through their dashboards.

Important Considerations

- **Environment Variables**: Always configure your `OPENAI_API_KEY` as an environment variable in your hosting provider's settings, rather than hardcoding it in your application.
- **Scaling**: Most cloud providers offer auto-scaling features to handle varying traffic loads.
- **Logging**: Familiarize yourself with your provider's logging and monitoring tools to debug issues in production.
- **Database (if applicable)**: If your application were to use a database, you would need to set up a managed database service (e.g., PostgreSQL on Heroku, AWS RDS) and configure your application to connect to it.

By following these instructions and consulting your chosen provider's documentation, you should be able to successfully deploy the Anna AI Coach permanently.