

42Amman AI Hackathon: Domain-Specific Chatbot

Objective

Build a domain-specific chatbot answering questions about 42Amman and Jordan using DistilGPT-2 with Hugging Face Transformers and Flask.

Tools & Libraries Required

- - Python 3.8+
- - Flask, Transformers, Torch, python-dotenv
- - VSCode or PyCharm
- - Git for version control
- - Vanilla JavaScript for frontend

Setup Instructions

- - Create and activate virtual environment:
- - `python3 -m venv venv`
- - `source venv/bin/activate`
- - Install dependencies:
- - `pip install flask transformers torch python-dotenv`
- - Run the app:
- - `python app.py`
- - Access the app at `http://localhost:5000`

Project Structure

- - chatbot/
- - |— app.py
- - |— .env
- - |— requirements.txt
- - |— templates/
- - |— index.html

Timeline

- - Day 1 Morning: Domain knowledge & prompt engineering

- - Day 1 Afternoon: Backend & DistilGPT-2 integration
- - Day 2 Morning: Frontend & end-to-end testing
- - Day 2 Afternoon: Debugging, enhancements, final preparations

Expected Deliverables

- - Working chatbot app with integrated DistilGPT-2 model
- - Functional prompt builder and domain knowledge dictionary
- - /ask endpoint and frontend interface
- - Well-documented code and version control

Instructions to Participants

- - Follow setup instructions to prepare your dev environment
- - Develop backend first, then integrate frontend

Submission Guidelines

- - Submit your GitHub repo link
- - Include working chatbot demo and documentation
- - Ensure code is clean and version controlled

Evaluation Criteria

- - Code Quality: 20 pts
- - Functionality & Accuracy: 30 pts
- - UI/UX Design: 20 pts
- - Creativity: 20 pts
- - Teamwork: 10 pts

Optional Enhancements

- - Retrieval-Augmented QA
- - Conversation History
- - Styling & UX features like message bubbles or typing indicators

Resources & References

- - [Hugging Face Transformers documentation](#)
- - [Flask documentation](#)
- - [Python official documentation](#)