Hack the Future: A Gen AI Sprint Powered by Data

Work at the heart of c*h*ange

Data and AI Week

Problem Statement 5: Enhancing Job Screening with AI and Data Intelligence

Challenge Overview:

The recruitment process often involves manually reviewing numerous job descriptions (JDs) and CVs, which can be time-consuming and prone to human error. The goal of this hackathon is to develop a multi-agentic AI system that can automatically read and summarize job descriptions (JDs), match candidate qualifications with the JD, shortlist candidates, and send interview requests based on the match

Current Process:

- **Job Description Summarizer**: Reads and summarizes key elements from the JD, including required skills, experience, qualifications, and job responsibilities.
- **Recruiting agent**: Extracts key data from CVs, such as education, work experience, skills, certifications, and other relevant information. Compares the extracted data from the CV to the summarized JD and calculates a match score based on the relevance of the candidate's qualifications, experience, and skills.
- **Shortlisting Candidates**: Based on the match score, candidates who meet or exceed a defined threshold (e.g., 80% match) are shortlisted for interviews.
- **Interview Scheduler**: Sends personalized interview requests to shortlisted candidates, including potential dates, times, and interview format written in email.

Expected Technical Output: Multiagent framework and SQLite Database for long term memory

Technology for AI

- Ollama based on-prem LLMs
- Custom tools for agents– API , web srapper, ML model etc..
- Ollama based embedding models
- SQLite DB
- Multi agent framework

Ollama

Step-by-Step Guide to Using Ollama with Python:

System Requirements

OS	Processor	RAM	Storage	Python Version
Windows	x86_64 (Intel/AMD)	8GB (16GB recommended)	10GB+ free space	3.8 or higher
macOS	Apple Silicon (M1/M2) or Intel x86_64	8GB (16GB recommended)	10GB+ free space	3.8 or higher
Linux	x86_64 or ARM64 CPU	8GB (16GB recommended)	10GB+ free space	3.8 or higher

Ollama

Step1: **Install Ollama**

Ollama allows you to run AI models locally on your system.

- **Go to the official Ollama website**: https://ollama.com
- **Download the installer** for your operating system:
  - **Windows**: Download the .exe file and run it.
  - **Mac**: Download the .dmg file and install it.
  - **Linux**: Open a terminal and run:
  - curl -fsSL https://ollama.com/install.sh | sh
- **Follow the installation steps** to complete the setup.
- **Verify the installation** by running:
- ollama version

If a version number appears, the installation was successful.

Step2: **Download a Model**

Now, download a lightweight AI model for your system.

- Open a terminal or command prompt.
- Run the following command to download the model:
- ollama pull mistral

(You can replace "mistral" with another lightweight model from the suggestions below.)

Step3: **Install Required Python Package**

To use Ollama with Python, install its Python package:

- pip install ollama

Step4: **Run a Model Using Python**

Now, let's write a simple Python script to interact with the AI model.

- Open a text editor or IDE (like VS Code, PyCharm, or even Notepad++).
- Create a new Python file (e.g., run_model.py).
- Copy and paste the following code:

```
import ollama
# Define the model name (use a lightweight model like "mistral")
model_name = "mistral"
# Define the user input prompt
prompt = "Explain what Agent AI is in simple words."
# Run the model and get a response
response = ollama.chat(model=model_name, messages=[{"role": "user", "content": prompt}])
# Print the response
print("AI Response:", response["message"]["content"])
```

- Save the file and run it using:

```
python run_model.py
```

Ollama

Here are a few recommended models:

Model Name	Description
TinyLlama-1.1B	Very lightweight, fast, good for quick demand forecasting.
Gemma-2B	Optimized for efficiency, handles structured data.
Phi-2	Great for reasoning and small-scale predictions.
Flan-T5 Small	Good for fine-tuning and specific NLP tasks.
DistilBERT	Super compact, great for text classification and embeddings.

To explore more models, visit: https://ollama.com/library.

We are looking for following from you:

- Agents' interaction design
- Technical approach slide
- Code structure
- Demo Video
- Final presentation which includes Problem statement, Final approach, Team, Solution, Benefit and Impact.

Research Agent Guidelines

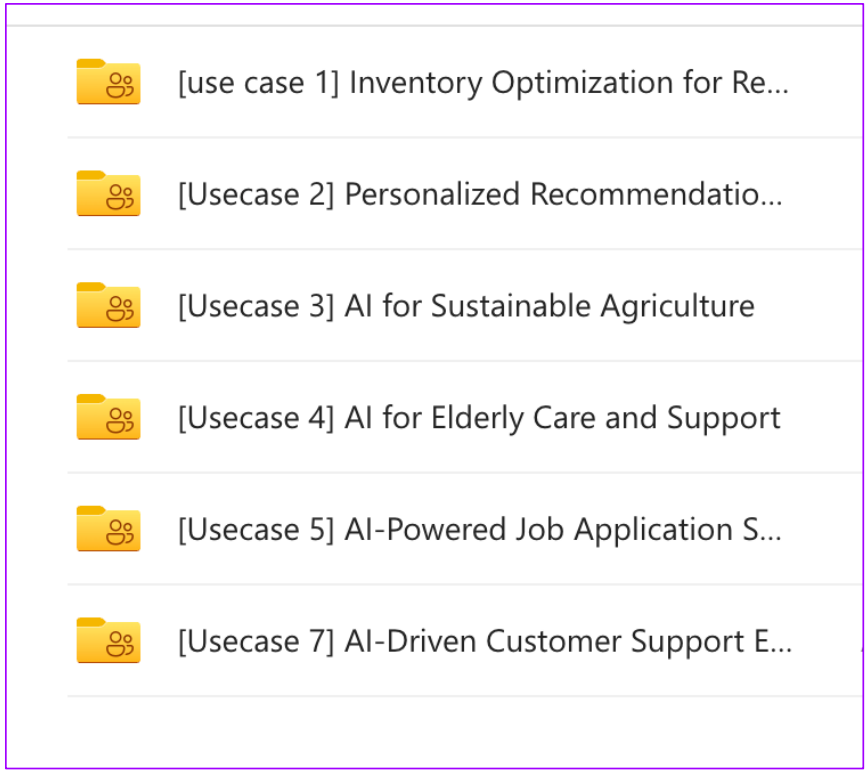
Guidelines for Ethical Web Scraping

When using web scraping to gather data for personalizing e-commerce experiences, it is important to follow ethical guidelines to respect the privacy and rights of website owners, users, and the public. Here are some key ethical principles to follow:

1. Respect Website Terms of Service
    - Always review and adhere to a website's Terms of Service (ToS) and robots.txt files to ensure that web scraping is allowed. If the site explicitly prohibits scraping in its terms, avoid scraping that site.
2. Data Privacy and User Consent
    - Ensure that the data you collect does not violate user privacy or data protection regulations such as GDPR (General Data Protection Regulation) or CCPA (California Consumer Privacy Act).
    - If personal data is scraped, avoid using sensitive information (like emails, passwords, etc.) unless explicit consent is given.
3. Scraping Frequency and Load
    - Do not overwhelm the website's server by sending too many requests in a short period. Scrape at a reasonable frequency to avoid causing performance issues or downtime.
    - Use appropriate delays between requests to avoid overloading the server and to minimize impact on the website's normal operation.
4. Data Anonymization
    - When collecting personal data, ensure that any personally identifiable information (PII) is either not scraped or is anonymized to prevent misuse or breaches of privacy.
    - Avoid scraping data that can directly identify individuals
5. Fair Use and Avoiding Copyright Violation
    - Avoid scraping copyrighted content, especially if it is not in the public domain, without permission. For example, scraping product descriptions or images may violate intellectual property rights.
    - Ensure that the data you scrape is used fairly and that it doesn't infringe on copyright laws or harm the website owners' business model.

Synthetic Data link

Access the Synthetic Dataset:
Click here to access



Thank You