Leuphana University
10.07.2025

BRAINSTORM CHATBOT

A group project for the course:

AI-based Applications

PROJECT DETAILS:

- **Trello**
- **Github**

PRESENTED BY:

Nadeem Hakimi (4000409)
Mia Mainka (3047072)
Marisa Jordan (4000444)
Daniel Kähm (4001842)

PRESENTED TO:

Debayan Banerjee
Ricardo Usbeck

Table of contents

Project Overview

1. Project Goal

web-based application that allows users to:

- Upload a PDF
- Ask questions, receive answers by LLM
- Extract key values

2. Core Functionalities Required

- PDF upload interface
- Text extraction, chunking, semantic search
- remote LLM API to generate context-based answers

3. Expected Output Formats

- Natural language answers (in chat form)
- Structured JSON with fields like:

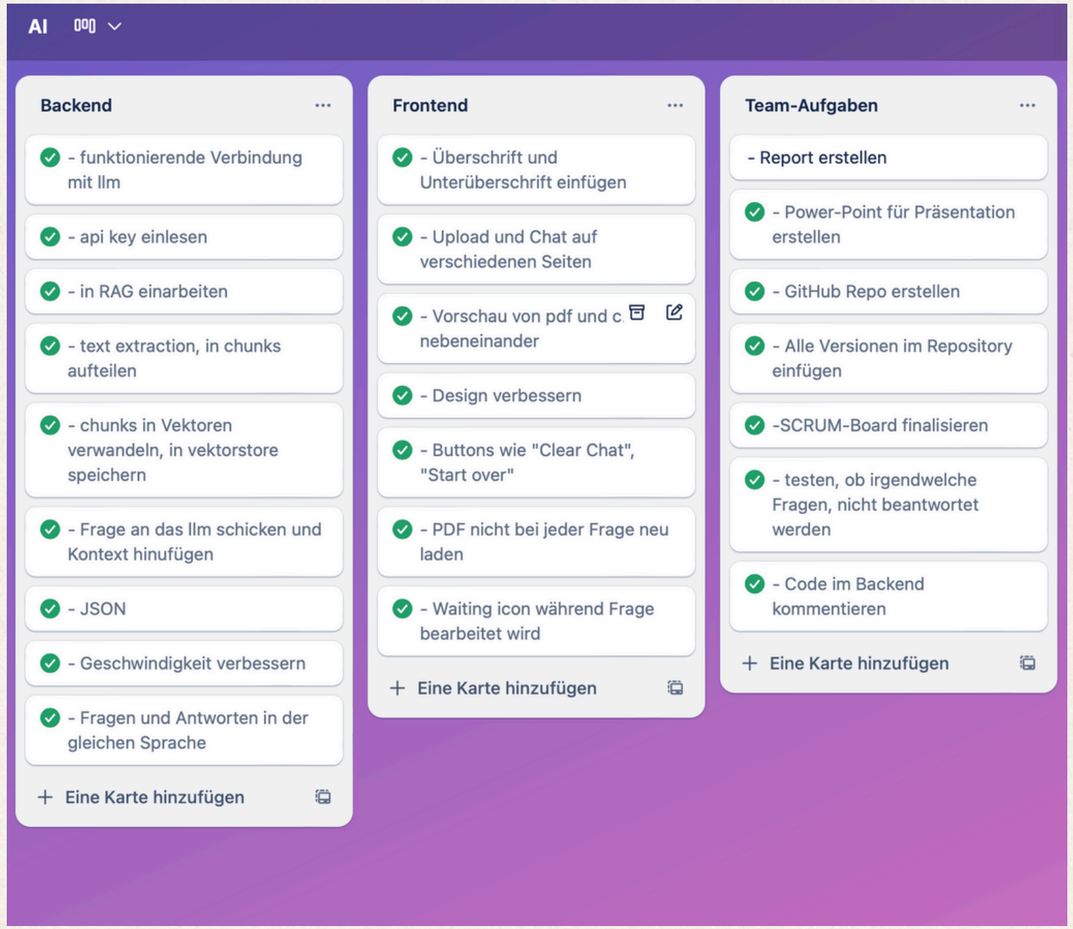
"CO2", "Risks", "Strategy", "Targets", etc.

Procedure

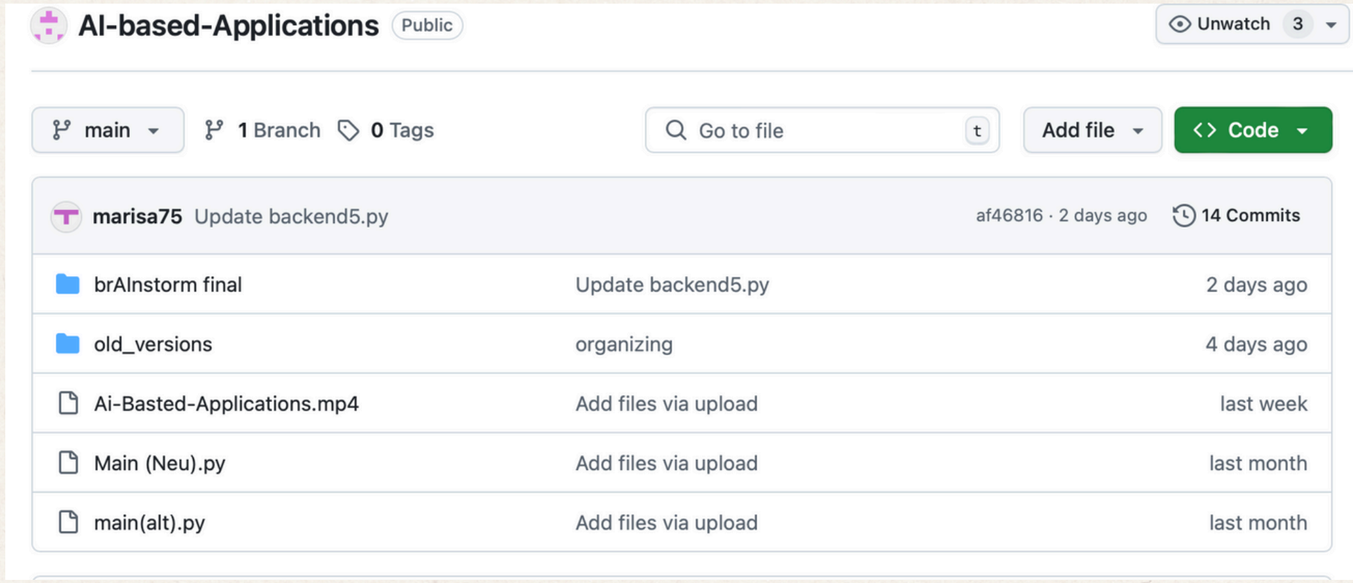
1. Setup
2. Zoom-Kickoff
3. Task division
4. Refining
5. Presentation

1. Setup

Trello



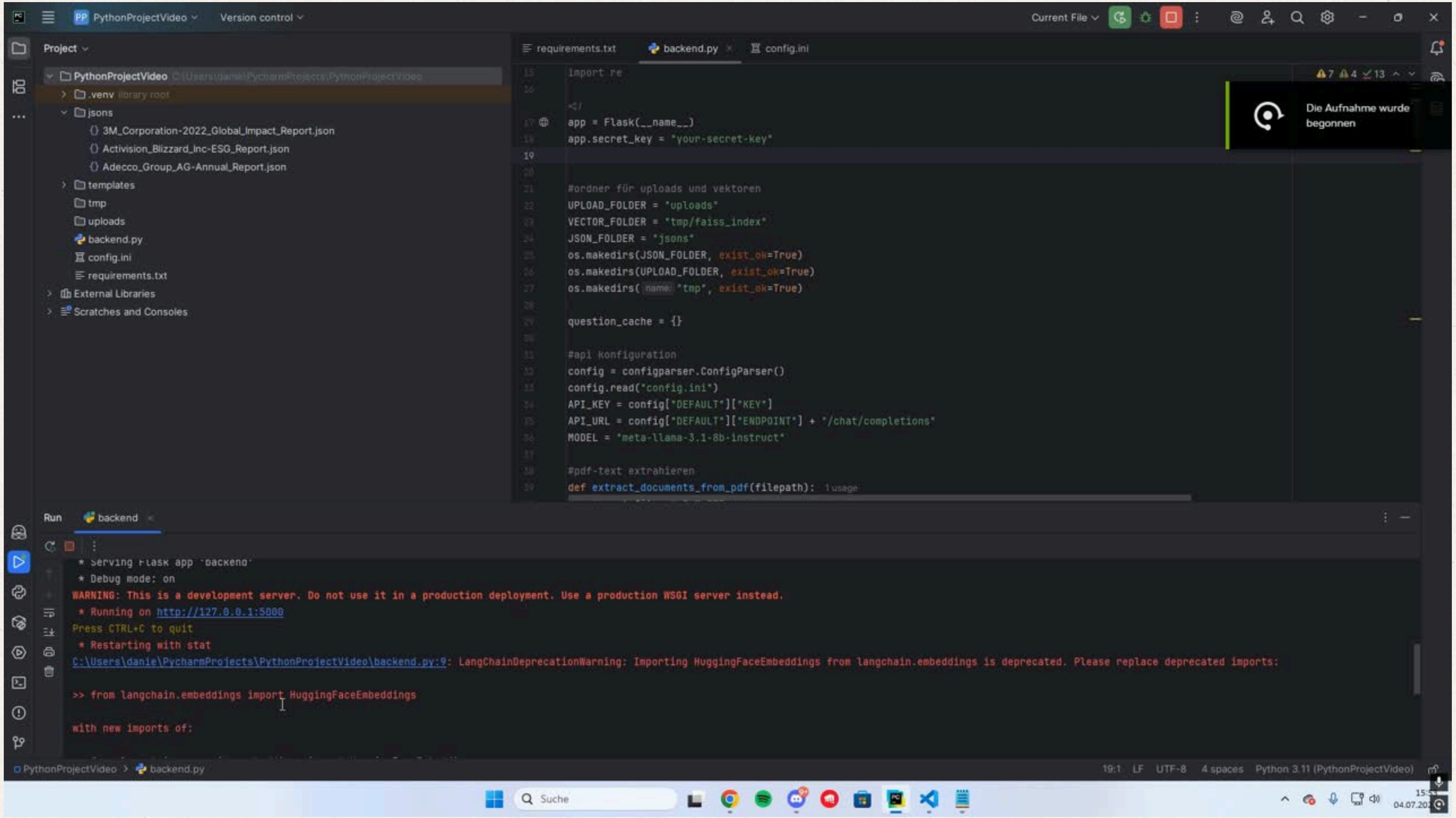
Git Hub



Procedure

1. Setup
2. Zoom-Kickoff
3. Task division
4. Refining
5. Presentation

Live Demo / Video



https://youtu.be/ZTpCEU-y3g8

Choice of Frameworks

1. Web Framework

Flask → Handles backend routing, sessions, and API communication

HTML/CSS – Frontend layout, PDF upload, and chat UI

Localhost – Local deployment during development

2. Document Handling

PyMuPDF – Extracts text from PDF files

LangChain – Handles chunking (RecursiveCharacterTextSplitter) and context retrieval

HuggingFace MiniLM – Creates vector embeddings (used within LangChain)

3. Vector Storage & Embeddings

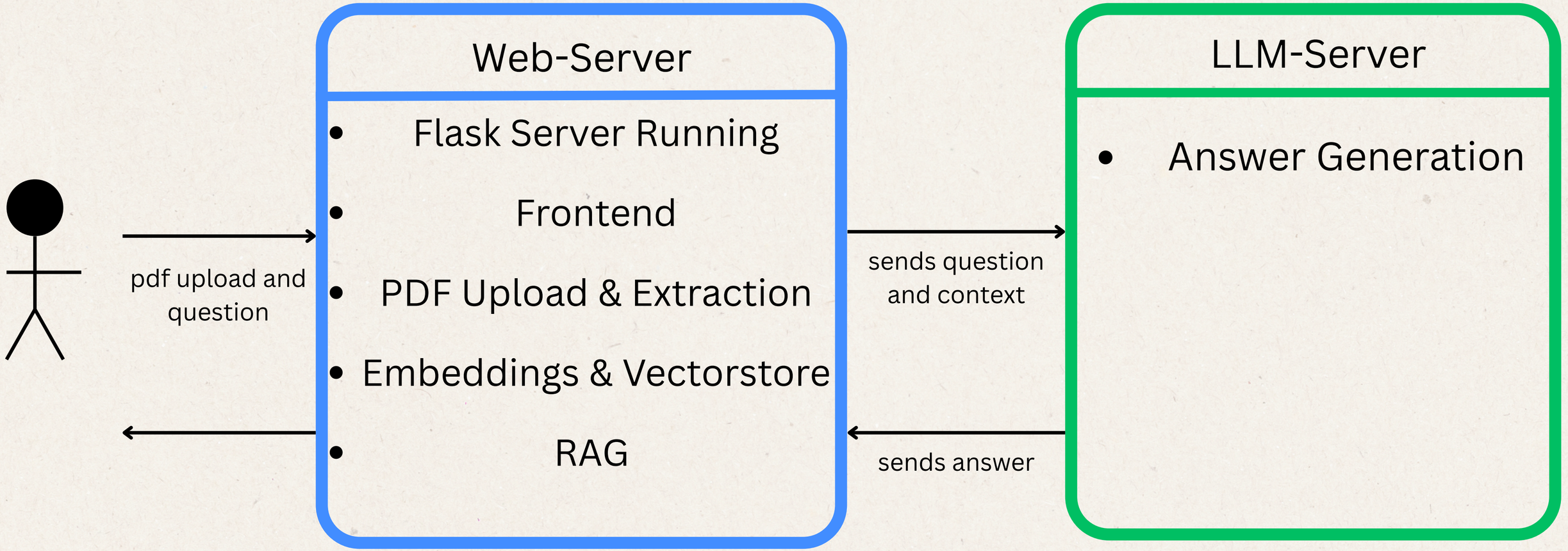
HuggingFace MiniLM – Converts text chunks into vector embeddings for similarity search

FAISS – Stores these vectors and retrieves the most relevant chunks based on user queries

4. LLM Integration

GWDG API (Meta-LLaMA 3.1) – Processes user queries with context and returns answers

System Architecture



What worked well

- Independent division of tasks within the group
- Good group dynamics
- Integration of all members based on their strengths
- Independently working on the project outside of our sessions

Problems

- Issues with understanding GitHub
- Errors occured when transfering data with Python
- Overall issues with the code
- Adjustment of speed

Aspects about Speed

```
#  Transform to chunks
splitter = RecursiveCharacterTextSplitter(chunk_size=1200, chunk_overlap=20)
documents = splitter.create_documents([full_text])
```

- **Differences of Chunk-Sizes**
- **Answers are saved, and if the same question is asked again, saved answer is sent back**
- **General structure of backend**

Contributions

Backend & LLM Integration - Nadeem, Marisa

Frontend & UI Design - Everyone

Prompt & JSON Output - Everyone

SCRUM & GitHub - Everyone

Presentation - Mia, Daniel

Demo & Recording - Daniel

Most areas were developed collaboratively, with individual members focusing more deeply on specific topics.

Links

Trello

https://trello.com/invite/b/681dd6ea555321118bfa12c2/ATTIe4da5f5b879ec56af2724870310de71a17EADBF2/ai

Git Hub Repository

htthttps://github.com/MiaMainka/sustainability-pdf-chatbots

Questions?

Thank you for your attention!

PRESENTED BY:
Nadeem Hakimi (4000409)
Mia Mainka (3047072)
Marisa Jordan (4000444)
Daniel Kähm (4001842)

PRESENTED TO:
Debayan Banerjee
Ricardo Usbeck

Page Number

- extract page number from question
- Only load first 50 pages
- If user asks specifically for a page number, those pages get loaded as well

```python
#findet Seitenzahl in der gestellten Frage heraus
def extract_page_number_from_question(question):  1 usage
    match = re.search( pattern: r"(Seite|page)\s*(\d+)", question, re.IGNORECASE)
    if match:
        return int(match.group(2))
    return None
```

Outcome

- fast extraction (because only 50 pages)
- Within these first 50 pages LLM could estimate the pages, so questions like “What is on page 40?” were possible
- If user doesn’t mention the page number and asks a context-based question about later pages, LLM can’t answer