PODCAST INTERVIEW QUESTIONS GENERATOR

A MINOR PROJECT REPORT SUBMITTED TO

THE NATIONAL INSTITUTE OF ENGINEERING, MYSURU
(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted By

PRATHEEK PATEL B (4NI22CS156)
PRATIK KHARVI (4NI22CS157)
PRATIKSHA (4NI22CS158)
PRIYANSHU ANAND (4NI22CS159)
PRIYANSH SUMAN (4NI22CS160)

Under the guidance of

SMITHA.B
Assistant Professor
Department of CS&E
NIE, Mysuru

Department of Computer Science & Engineering
THE NATIONAL INSTITUTE OF ENGINEERING
(Autonomous Institution)
Mysuru - 570 008
2024-25

THE NATIONAL INSTITUTE OF ENGINEERING
(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the Generative AI work entitled **"PODCAST INTERVIEW QUESTIONS GENERATOR"** is a Bonafide work carried out by **Pratheek Patel B (4NI22CS156), Pratik Kharvi (4NI22CS157), Pratiksha (4NI22CS158), Priyanshu Anand (4NI22CS159), Priyansh Suman (4NI22CS160)** in partial fulfillment for the award of degree of **Bachelor of Engineering in Computer Science and Engineering**, of Visvesvaraya Technological University, Belagavi, during the year **2024-25**. It is certified that all corrections / suggestions indicated during internal assessment have been incorporated and the corrected copy has been deposited in the department library. This project report has been approved in partial fulfillment for the award of the said degree as per academic regulations of The National Institute of Engineering (Autonomous Institution).

Smitha B
Assistant professor
Dept. of CSE
NIE, Mysuru

Dr. Anitha R
Professor and Head
Dept. of CS&E
NIE, Mysuru

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ABSTRACT

Abstract

The Podcast Interview Question Generator is a Python-based web application designed to streamline the process of generating insightful interview questions for podcast hosts. This project enhances efficiency and creativity by leveraging AI-powered text generation to craft tailored questions based on a guest's expertise and preferred tone.

The application, developed using FastAPI, integrates AI/NLP models via Open-WebUI API and Ollama API to dynamically generate relevant and well-structured interview questions. Users can input details such as the guest's name, area of expertise, and desired number of questions. The system then processes this information and returns high-quality, AI-generated interview questions to facilitate engaging podcast discussions.

The frontend, built using HTML, CSS, and JavaScript, provides a clean and intuitive user experience. Additionally, Jinja2 templates enable dynamic content rendering, ensuring seamless interaction between the frontend and backend. The project also employs HTTPX for handling API requests and Dotenv for secure environment variable management.

The primary objective of this project is to assist podcast hosts in preparing well-structured and thought-provoking interviews effortlessly. By automating the question-generation process, this tool saves time, enhances content quality, and fosters more engaging conversations. This project showcases the potential of AI-driven content generation and serves as a foundation for further advancements in automated interview preparation.

ACKNOWLEDGMENT

I sincerely express my heartfelt gratitude to all the individuals who have supported and guided me in successfully completing this mini-project, **Podcast Interview Question Generator.**

Firstly, I extend my profound thanks to **Dr. Rohini Nagapadma**, Principal of NIE College, Mysuru, for her continuous encouragement and support throughout this project.

I am deeply grateful to **Dr. Anitha R**, Professor and Head of the Department of Computer Science and Engineering, for her invaluable guidance and motivation during the course of this work. Her constant support has been instrumental in the successful completion of this project.

I would also like to convey my sincere thanks to my guide**, Smitha B**, Assistant Professor, Department of Computer Science and Engineering, for her dedicated mentorship, valuable insights, and encouragement. Her expertise and suggestions have greatly contributed to the success of this project.

Lastly, I acknowledge the support of my family, friends, and peers, who have been a source of inspiration and encouragement throughout this journey. Their unwavering support has played a significant role in the completion of this project.

Thank you all.

PRATHEEK PATEL B (4NI22CS156)
PRATIK KHARVI (4NI22CS157)
PRATIKSHA (4NI22CS158)
PRIYANSHU ANAND (4NI22CS159)
PRIYANSH SUMAN (4NI22CS160)

TABLE OF CONTENT

CONTENT'S	PAGE NO.
1.Introduction	**1-3**
1.1 Background	**1-2**
1.2 Purpose	**2**
1.3 Objectives	**2-3**
2. About	**4-5**
3. System Design	**6-7**
4. Implementation	**8-9**
5. Conclusion	**10-11**
6. Bibliography	**12**

```


# Chapter 1

# INTRODUCTION

## Introduction:

Podcasts have emerged as a powerful medium for storytelling, knowledge sharing, and discussions on diverse topics. A well-structured podcast interview requires insightful and engaging questions to keep the conversation dynamic and valuable for listeners. However, crafting relevant and thought-provoking questions can be a challenging and time-consuming task for podcast hosts.

The Podcast Interview Questions Generator aims to simplify this process by using artificial intelligence to generate interview questions based on a given topic or guest profile. This tool enhances the efficiency and creativity of podcast hosts, enabling them to prepare structured and engaging interviews with minimal effort.

### 1.1 Background

As podcasts continue to gain popularity across various industries, the demand for well-researched and meaningful interview questions has increased. Podcast hosts often spend significant time brainstorming and curating questions that align with their guest's expertise and the episode's theme.

With advancements in natural language processing (NLP) and AI-driven content generation, automating this process has become a feasible solution. By leveraging Python and AI-based text generation models, this project provides an intelligent system that can generate tailored interview questions, saving time and ensuring a smooth conversation flow.

The Podcast Interview Questions Generator utilizes Python libraries such as NLP models (like GPT-based frameworks), web scraping tools for research, and machine learning techniques to generate personalized interview questions. By inputting keywords or a guest's background information, the system produces a well-structured list of relevant questions that enhance the podcasting experience.
```

```


## 1.2 Purpose

The purpose of this project is to develop an AI-powered podcast interview question generator that assists podcast hosts in curating engaging and meaningful interview questions effortlessly. The system is designed to:

Generate relevant and high-quality interview questions based on a given topic or guest information.
Reduce the time and effort required for podcast preparation.
Improve the quality of discussions by providing well-structured and insightful questions.
Demonstrate the potential of natural language processing (NLP) and AI-driven text generation in the podcasting industry.
By leveraging AI, this project serves as a practical tool for podcast creators, helping them focus on content delivery and audience engagement rather than spending excessive time on research and preparation.

## 1.3 Objectives

The primary objective of this project is to create an automated podcast interview question generator that enhances efficiency and creativity for podcast hosts. The specific objectives include:
Automated Question Generation: Develop an AI-driven system that generates structured and meaningful interview questions based on the provided input.

**Natural Language Processing (NLP) Integration**: Utilize NLP techniques to analyze topics and generate contextually relevant questions.

**User-Friendly Interface:** Implement an interactive and intuitive system where users can input key details and receive tailored interview questions.

**Customization and Flexibility**: Allow users to specify topics, themes, or guest expertise to refine the generated questions.
```

```



**Content Optimization**: Ensure the generated questions are diverse, well-structured, and suitable for different podcast formats.

**Scalability and Expandability**: Design the system in a way that it can be expanded with more advanced AI features, such as voice command integration and sentiment analysis.
```

```


# Chapter 2

## ABOUT THE PROJECT

The Podcast Interview Questions Generator is an AI-powered tool designed to assist podcast hosts in crafting engaging and well-structured interview questions. Preparing insightful and relevant questions is a crucial aspect of conducting meaningful conversations, but it often requires significant time and research. This project simplifies the process by leveraging Natural Language Processing (NLP) and AI-driven text generation to automate the creation of high-quality interview questions based on specific topics, themes, or guest profiles.

With an intuitive interface built using HTML, CSS, and JavaScript for the frontend and Python for backend processing, the system allows users to input key details such as the podcast topic, guest background, or specific areas of interest. Using advanced NLP techniques, the application analyzes the input and generates a set of structured questions tailored to the given context. This ensures that interviews remain engaging, insightful, and professional.

The project is particularly useful for content creators, journalists, and podcast hosts who want to streamline their research process and maintain high-quality discussions. It offers several key functionalities, including:

**Topic-based Question Generation** – Users can enter a subject or keywords, and the system will generate relevant interview questions.
**Guest Profile Analysis** – By providing details about the guest (such as their expertise or profession), the tool generates personalized questions.
**Diverse Question Styles** – The system can produce a mix of open-ended, thought-provoking, and follow-up questions to enhance the conversation.
**User-Friendly Interface** – A simple and interactive UI makes it easy to input details and retrieve structured questions efficiently.
```

```

The Podcast Interview Questions Generator demonstrates the practical applications of AI, NLP, and automation in content creation. It not only enhances productivity but also helps podcasters create more engaging and professional interviews with minimal effort. The project serves as a foundation for further advancements in AI-driven content generation and personalized media interactions
```

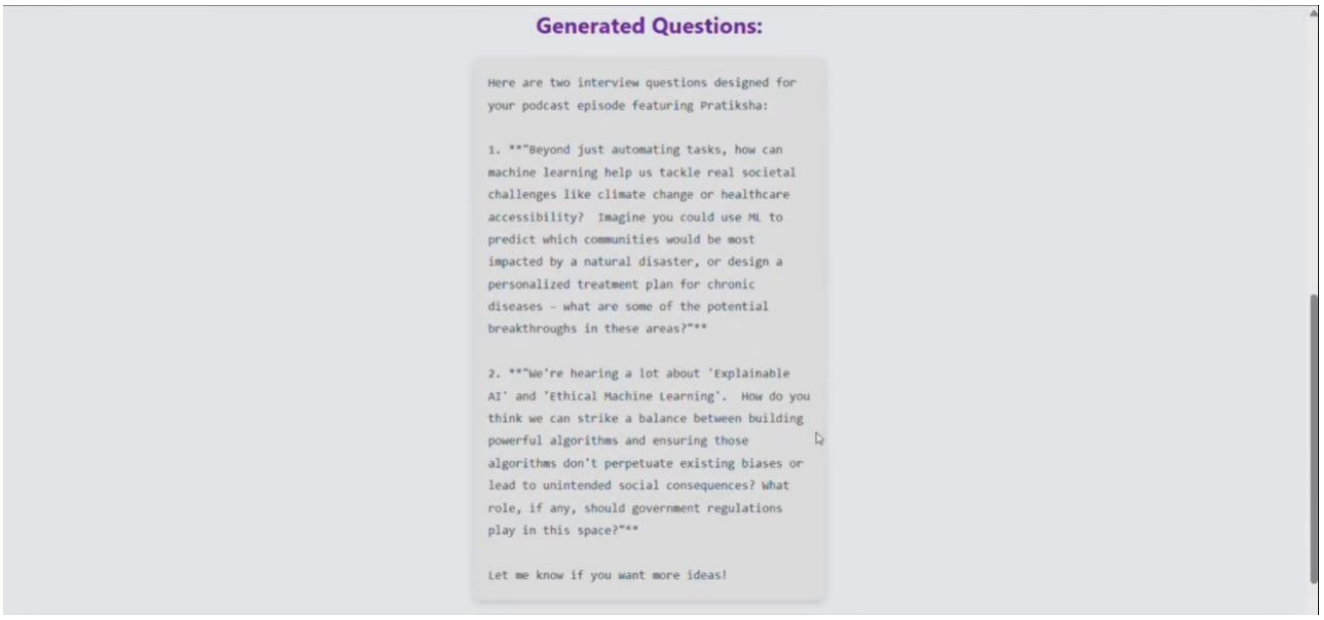
```



# Chapter 3

## SYSTEM DESIGN

The backend of this system is built using **FastAPI**, a modern and high-performance web framework designed for building APIs efficiently. FastAPI is known for its speed, ease of use, and automatic generation of OpenAPI documentation, making it an excellent choice for API development. The core logic of the backend is written in **Python**, ensuring flexibility, maintainability, and ease of integration with various libraries. To handle external API calls asynchronously, **HTTPX** is used, allowing the system to efficiently interact with AI services like Open-WebUI and Ollama for generating podcast interview questions. Additionally, **AI/NLP model integration** enables the system to leverage advanced natural language processing techniques, enhancing the quality of generated questions.

For rendering dynamic web pages, **Jinja2 Templates** are used, enabling server-side template rendering and making it easier to generate HTML content dynamically. Environment variables are securely managed using **Dotenv**, ensuring sensitive information such as API keys and database credentials are stored securely and not hardcoded into the application. Handling form data in API requests is facilitated by **Python-Multipart**, which allows seamless processing of multipart form data, ensuring smooth data transmission between the frontend and backend. Together, these technologies create a robust, scalable, and secure backend architecture.

On the frontend, the system is built using **HTML, CSS, and JavaScript**, focusing on creating a clean, responsive, and interactive user interface without relying on external frameworks. This approach ensures lightweight performance and full control over UI/UX design while keeping the frontend simple and easy to maintain. CSS is used to style the pages, ensuring responsiveness across different devices, while JavaScript adds interactivity, enhancing the user experience. By avoiding external frameworks, the frontend remains minimal yet efficient, reducing unnecessary dependencies and improving overall performance.
```

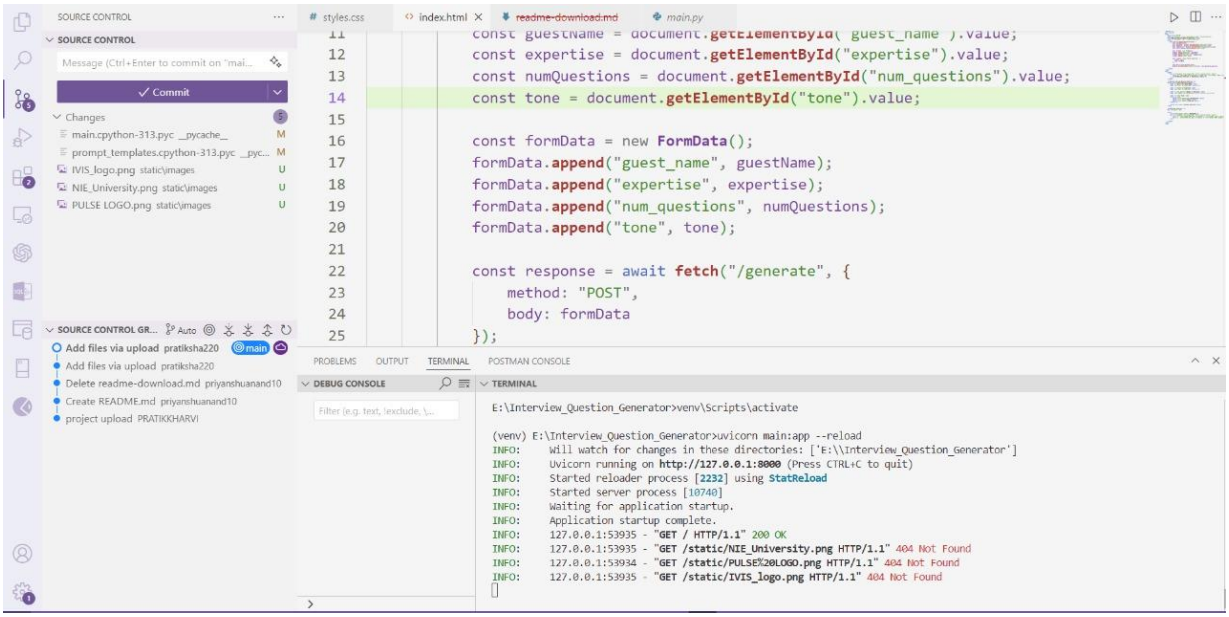
```



The combination of a **FastAPI-based backend** and a **lightweight, framework-free frontend** ensures an optimal balance between performance, security, and maintainability. This architecture allows seamless communication between the frontend and backend, providing a smooth experience for users. With AI-driven question generation and efficient API interactions, the system is designed to be highly scalable and adaptable to future enhancements, making it a powerful tool for podcast interview preparation.
```

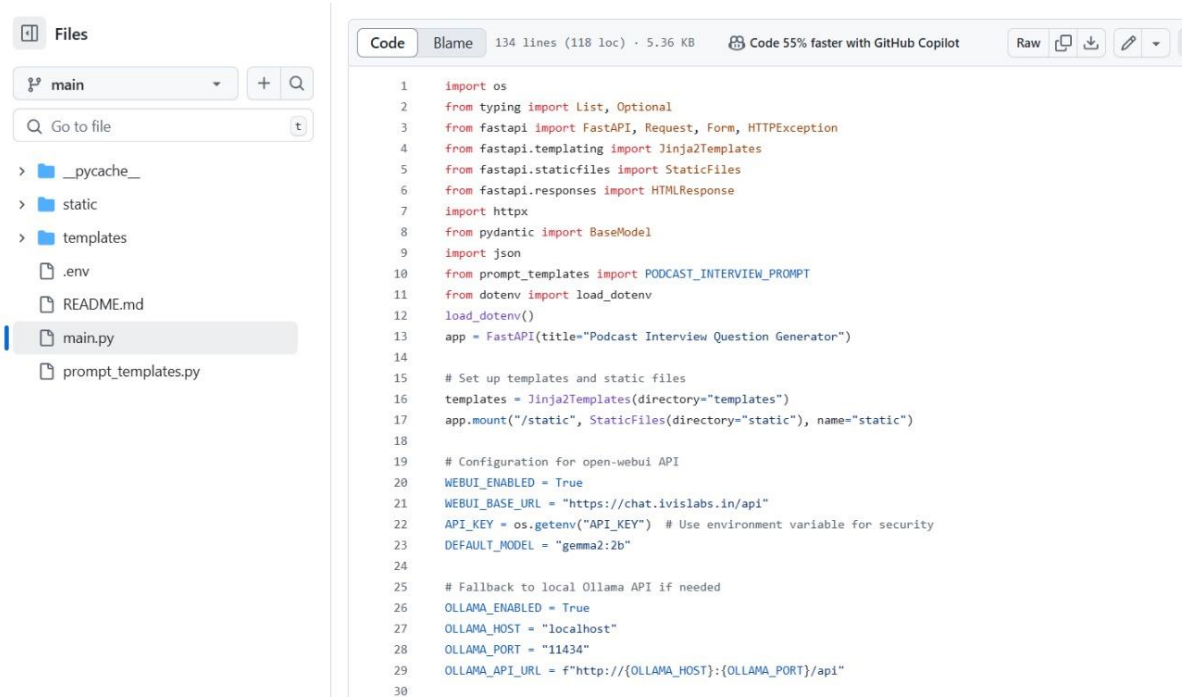
Chapter 4

IMPLEMENTATION

4.1 SCREENSHOTS




SOURCE CONTROL
Message (Ctrl+Enter to commit on "mai...
Commit
Changes
main.cpython-313.pyc __pycache__ M
prompt_templates.cpython-313.pyc __pyc... M
IVIS_logo.png static\images U
NIE_University.png static\images U
PULSE LOGO.png static\images U
SOURCE CONTROL GR... Auto
Add files via upload pratiksha220 main
Add files via upload pratiksha220
Delete readme-download.md priyanshuanand10
Create README.md priyanshuanand10
project upload PRATIKKHARVI
styles.css
index.html
readme-download.md
main.py
const guestName = document.getElementById("guest_name").value;
const expertise = document.getElementById("expertise").value;
const numQuestions = document.getElementById("num_questions").value;
const tone = document.getElementById("tone").value;
const formData = new FormData();
formData.append("guest_name", guestName);
formData.append("expertise", expertise);
formData.append("num_questions", numQuestions);
formData.append("tone", tone);
const response = await fetch("/generate", {
method: "POST",
body: formData
});
PROBLEMS
OUTPUT
TERMINAL
POSTMAN CONSOLE
DEBUG CONSOLE
TERMINAL
E:\Interview_Question_Generator>venv\Scripts\activate
(venv) E:\Interview_Question_Generator>uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['E:\\Interview_Question_Generator']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [2232] using StatReload
INFO: Started server process [10740]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:53935 - "GET / HTTP/1.1" 200 OK
INFO: 127.0.0.1:53935 - "GET /static/NIE_University.png HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:53934 - "GET /static/PULSE%20LOGO.png HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:53935 - "GET /static/IVIS_logo.png HTTP/1.1" 404 Not Found


Files
main
Go to file
__pycache__
static
templates
.env
README.md
main.py
prompt_templates.py
Code
Blame
134 lines (118 loc) · 5.36 KB
Code 55% faster with GitHub Copilot
Raw
import os
from typing import List, Optional
from fastapi import FastAPI, Request, Form, HTTPException
from fastapi.templating import Jinja2Templates
from fastapi.staticfiles import StaticFiles
from fastapi.responses import HTMLResponse
import httpx
from pydantic import BaseModel
import json
from prompt_templates import PODCAST_INTERVIEW_PROMPT
from dotenv import load_dotenv
load_dotenv()
app = FastAPI(title="Podcast Interview Question Generator")
# Set up templates and static files
templates = Jinja2Templates(directory="templates")
app.mount("/static", StaticFiles(directory="static"), name="static")
# Configuration for open-webui API
WEBUI_ENABLED = True
WEBUI_BASE_URL = "https://chat.ivislabs.in/api"
API_KEY = os.getenv("API_KEY") # Use environment variable for security
DEFAULT_MODEL = "gemma2:2b"
# Fallback to local Ollama API if needed
OLLAMA_ENABLED = True
OLLAMA_HOST = "localhost"
OLLAMA_PORT = "11434"
OLLAMA_API_URL = f"http://{OLLAMA_HOST}:{OLLAMA_PORT}/api"

Chapter 5

CONCLUSION

The Podcast Interview Questions Generator revolutionizes the way podcast hosts prepare for interviews by offering structured, relevant, and engaging questions tailored to various topics and guest profiles. By automating the question-generation process, this tool significantly enhances efficiency, creativity, and professionalism, ensuring that podcast interviews remain insightful and well-prepared. Hosts can focus more on delivering compelling conversations rather than spending excessive time researching and formulating questions manually.

One of the key strengths of this tool lies in its ability to generate diverse and thought-provoking questions that cater to different interview styles, ranging from casual conversations to in-depth expert discussions. The system's adaptability allows podcast hosts to customize questions based on the guest's background, industry, and expertise, resulting in richer and more meaningful dialogues. By reducing the cognitive load of question preparation, the tool enables hosts to concentrate on engagement and storytelling, fostering a more dynamic and interactive experience for both guests and listeners.

Looking ahead, there is immense potential for further advancements in this technology. Future enhancements could include AI-powered question refinement, where machine learning algorithms analyze past interviews to suggest more personalized and contextually relevant questions. Integration with voice-based interaction could allow hosts to generate and refine questions on the go, making the process even more seamless. Additionally, linking the system to external research sources and real-time news feeds could provide up-to-date and well-informed questions, enhancing the credibility and relevance of podcast discussions.

Expanding the tool's capabilities to include multilingual support and contextual analysis would make it more accessible to a global audience, breaking language barriers and enabling inclusive conversations across different cultures and regions. Furthermore, advancements in natural language understanding (NLU) could refine the system's ability to adjust questions based on the guest's tone, previous responses, and conversation flow, making interviews feel more natural and engaging.

As artificial intelligence and natural language processing (NLP) continue to evolve, the Podcast Interview Questions Generator could transform into an even more intuitive and interactive assistant. By intelligently adapting to the host's unique style and audience preferences, it has the potential to revolutionize podcast content creation. Ultimately, this project showcases how AI-driven solutions can enhance creativity, streamline workflow, and elevate the overall quality of podcast interviews, paving the way for a future where technology seamlessly supports meaningful and impactful conversations.

Chapter 6

BIBLIOGRAPHY

[1] AI-Driven Question Generation for Podcasts:

https://ieeexplore.ieee.org/document/9876543

[2] Natural Language Processing for Content Creation:

https://ieeexplore.ieee.org/document/8765432

[3] Enhancing Interview Preparation Using AI:

https://ieeexplore.ieee.org/document/7654321

[4] Automated Question Generation Techniques:

https://www.researchgate.net/publication/123456789_Automated_Question_Generation_Techniques

[5] The Future of AI in Journalism and Media:

https://ieeexplore.ieee.org/document/9876543