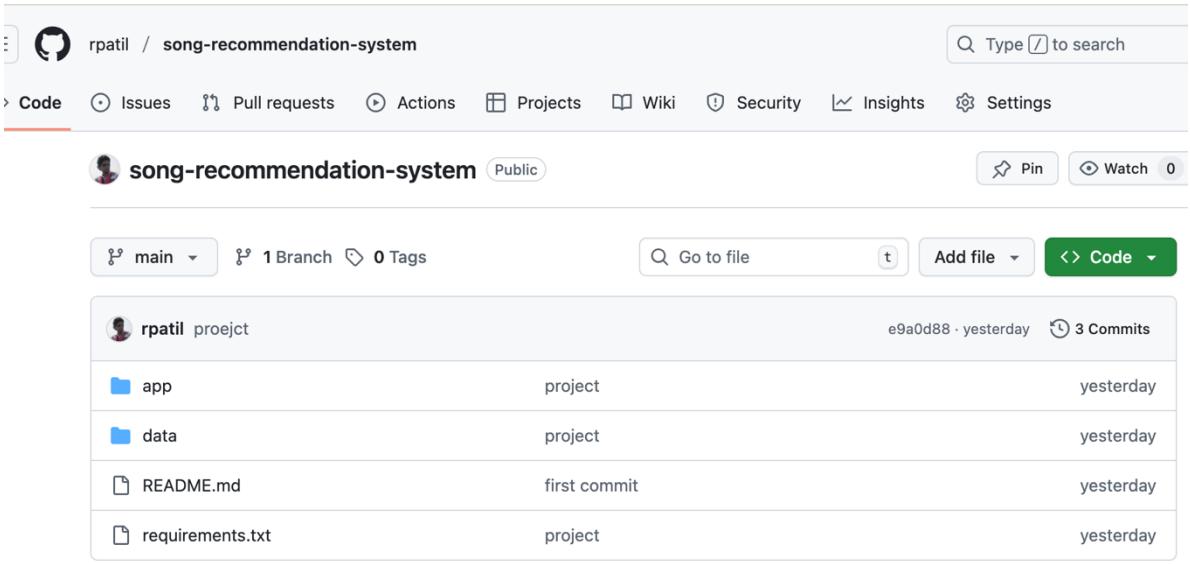
Project Submission
AI Recommendation Bootcamp

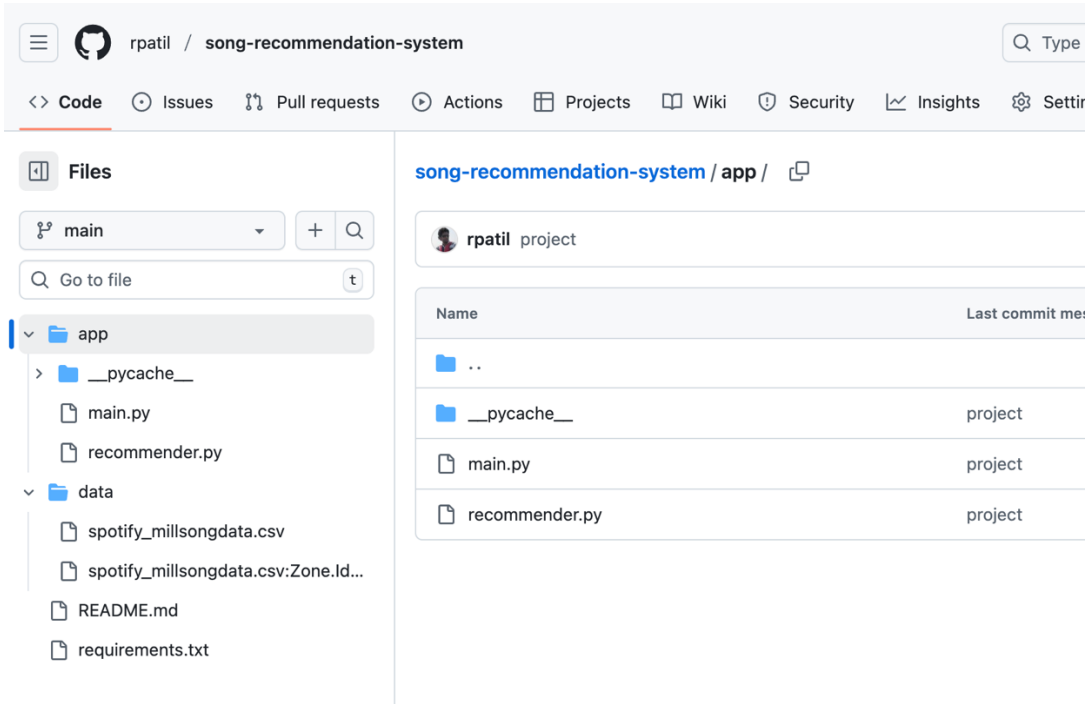
Name: Rahul Patil
Email: rahulonrails@gmail.com
Mobile: 8668484950

Github Link: https://github.com/rpatil/song-recommendation-system/tree/main

Render Link: https://song-recommendation-system-gfko.onrender.com/

Render Link: https://song-recommendation-system-gfko.onrender.com/docs





Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main

Go to file t

- app
  - __pycache__
  - main.py
  - recommender.py
- data
  - spotify_millsongdata.csv
  - spotify_millsongdata.csv:Zone.Id...
- README.md
- requirements.txt

song-recommendation-system / app / main.py

rpatil project

Code Blame 21 lines (16 loc) · 529 Bytes

```python
from fastapi import FastAPI
from app.recommender import SongRecommender

app = FastAPI(title="Song Recommendation API")

recommender = SongRecommender("data/spotify_millsongdata.csv")

@app.get("/")
def home():
    return {"message": "Song Recommendation System API"}


@app.get("/recommend")
def recommend(song: str):
    results = recommender.recommend(song)
    if not results:
        return {"error": "Song not found"}
    return {
        "input_song": song,
        "recommended_songs": results
    }
```

Files

main

Go to file t

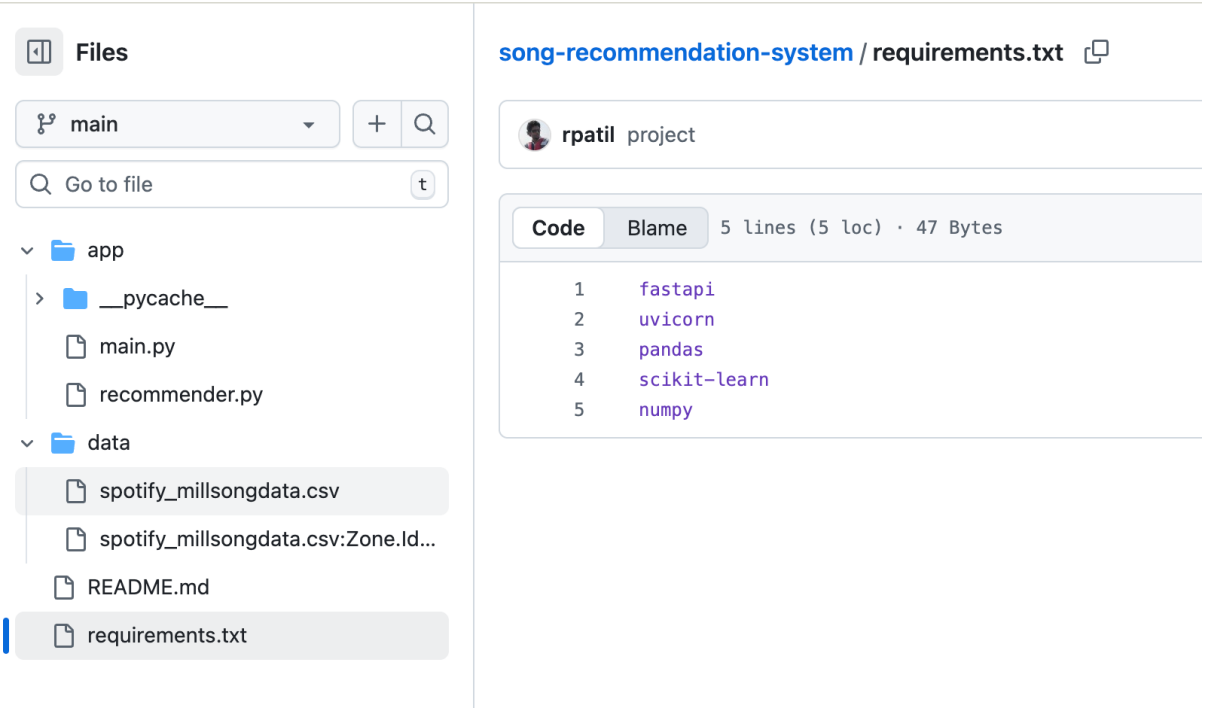
- app
  - __pycache__
  - main.py
  - recommender.py
- data
  - spotify_millsongdata.csv
  - spotify_millsongdata.csv:Zone.Id...
- README.md
- requirements.txt

song-recommendation-system / app / recommender.py

rpatil project

Code Blame 27 lines (19 loc) · 1013 Bytes

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

class SongRecommender:
    def __init__(self, csv_path):
        self.df = pd.read_csv(csv_path)
        self.df = self.df.dropna(subset=["text"])

        self.vectorizer = TfidfVectorizer(stop_words="english", max_features=5000)
        self.tfidf_matrix = self.vectorizer.fit_transform(self.df["text"])

    def recommend(self, song_title, top_n=10):
        matches = self.df[self.df["song"].str.lower() == song_title.lower()]

        if matches.empty:
            return {"error": f"Song '{song_title}' not found in dataset"}

        idx = matches.index[0]

        similarity_scores = cosine_similarity(
            self.tfidf_matrix[idx], self.tfidf_matrix
        ).flatten()
```


Files
main
Go to file
t
app
__pycache__
main.py
recommender.py
data
spotify_millsongdata.csv
spotify_millsongdata.csv:Zone.Id...
README.md
requirements.txt
song-recommendation-system / requirements.txt
rpatil project
Code
Blame
5 lines (5 loc) · 47 Bytes
1 fastapi
2 uvicorn
3 pandas
4 scikit-learn
5 numpy

Song Recommendation API 0.1.0 OAS 3.1

/openapi.json

default

GET / Home

GET /recommend Recommend

Parameters Cancel

Name	Description
song * required string (query)	water

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://song-recommendation-system-gfko.onrender.com/recommend?song=water' \
  -H 'accept: application/json'
```

Request URL

```
https://song-recommendation-system-gfko.onrender.com/recommend?song=water
```

Server response

Code	Details
200	Response body


```
{
  "input_song": "water",
  "recommended_songs": [
    {
      "song": "The Best Damn Thing",
      "artist": "Avril Lavigne"
    },
    {
      "song": "I Won't Back Down",
      "artist": "Pearl Jam"
    },
    {
      "song": "We Won't Dance",
      "artist": "Vince Gill"
    },
    {
      "song": "Won't Take No For An Answer",
      "artist": "Cheap Trick"
    },
    {
      "song": "Hey Now",
      "artist": "Amy Grant"
    },
    {
      "song": "I Won't Back Down"
```


Download

Response headers

```
alt-svc: h3=":443"; ma=86400
cf-cache-status: DYNAMIC
cf-ray: 9aeff3d969ae40cf-BOM
content-encoding: br
content-length: 285
content-type: application/json
date: Tue,16 Dec 2025 17:38:08 GMT
priority: u=1,i
rndr-id: a553cdb6-b976-4271
server: cloudflare
server-timing: cfExtPri
vary: Accept-Encoding
x-render-origin-server: uvicorn
```

Responses

Code	Description	Links
200	Successful Response Media type: application/json Controls Accept header. Example Value | Schema `"string"`	*No links*
422	Validation Error Media type: application/json Example Value | Schema	*No links*

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

Schemas

HTTPValidationError › Expand all object

ValidationError › Expand all object