

Project Submission

AI Recommendation Bootcamp

Name: Rahul Patil

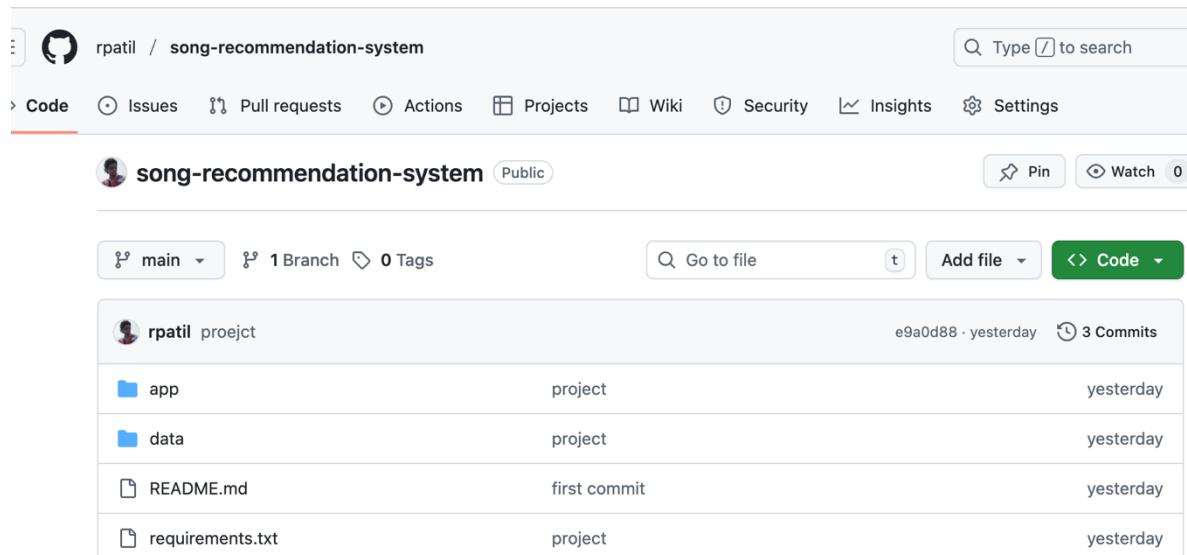
Email: rahulonrails@gmail.com

Mobile: 8668484950

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

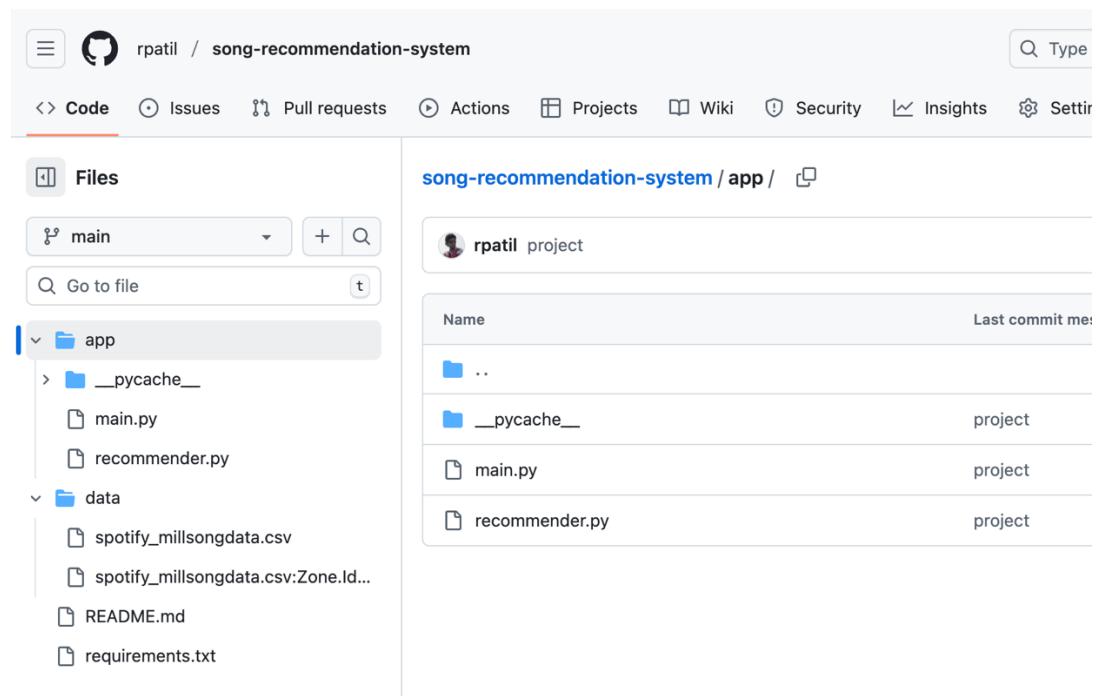
Render Link: <https://song-recommendation-system-gfko.onrender.com/>

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



The screenshot shows the GitHub repository 'song-recommendation-system' for user 'rpatil'. The repository is public and has 1 branch and 0 tags. The main branch is selected. Recent commits are listed, all made by 'rpatil' yesterday, with commit IDs like e9a0d88 and 3 commits. The files listed are app, data, README.md, and requirements.txt.

File	Type	Last Commit
app	project	yesterday
data	project	yesterday
README.md	first commit	yesterday
requirements.txt	project	yesterday



The screenshot shows the 'app' directory within the 'song-recommendation-system' repository. The directory structure includes 'app', 'data', and 'main'. The 'app' directory contains '_pycache_', 'main.py', and 'recommender.py'. The 'data' directory contains 'spotify_millsongdata.csv' and 'spotify_millsongdata.csv:Zone.Id...'. The 'main' directory contains 'README.md' and 'requirements.txt'. The right panel shows the contents of the 'app' directory, listing '_pycache_', 'main.py', and 'recommender.py'.

Name	Last commit
...	project
main.py	project
recommender.py	project

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

main Go to file

- 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

```
1  from fastapi import FastAPI
2  from app.recommender import SongRecommender
3
4  app = FastAPI(title="Song Recommendation API")
5
6  recommender = SongRecommender("data/spotify_millsongdata.csv")
7
8  @app.get("/")
9  def home():
10     return {"message": "Song Recommendation System API"}
11
12
13 @app.get("/recommend")
14 def recommend(song: str):
15     results = recommender.recommend(song)
16     if not results:
17         return {"error": "Song not found"}
18     return {
19         "input_song": song,
20         "recommended_songs": results
21     }
```

Files

main Go to file

- 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

```
1  import pandas as pd
2  from sklearn.feature_extraction.text import TfidfVectorizer
3  from sklearn.metrics.pairwise import cosine_similarity
4
5  class SongRecommender:
6      def __init__(self, csv_path):
7          self.df = pd.read_csv(csv_path)
8          self.df = self.df.dropna(subset=["text"])
9
10         self.vectorizer = TfidfVectorizer(stop_words="english", max_features=5000)
11         self.tfidf_matrix = self.vectorizer.fit_transform(self.df["text"])
12
13     def recommend(self, song_title, top_n=10):
14         matches = self.df[self.df['song'].str.lower() == song_title.lower()]
15
16         if matches.empty:
17             return {"error": f"Song '{song_title}' not found in dataset"}
18
19         idx = matches.index[0]
20
21         similarity_scores = cosine_similarity(
22             self.tfidf_matrix[idx], self.tfidf_matrix
23         ).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

rpatali 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

Name Description

song required string (query) water

Cancel

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-cdn: cloudflare
cf-ray: 3aef53d989a4e4cf-80M
content-encoding: br
content-length: 285
content-type: application/json
date: Tue, 16 Dec 2025 17:38:08 GMT
priority: u1,i
rndr-id: a3d955-8978-4271
server: cloudflare
server-timing: cfFrontPr
vary: Accept-Encoding
x-render-origin-server: unicorn
```

Responses

Code Description Links

200 Successful Response No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
"string"
```

422 Validation Error No links

Media type

application/json

Example Value Schema

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

Schemas

HTTPValidationError > Expand all object

ValidationError > Expand all object