

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



English Dictionary using Python with Django framework

Submitted By

Name	ID
Faria Sultana Avir	C201203
Faiza Binta Ali	C201213
Khadizatul Kubra	C201225
Farida Nusrat	C201242

Course Code: CSE-3638 + CSE-3640

Course Title: Software Engineering & Software Development Lab



Dept. of Computer Science & Engineering.

Table of Content

Serial No.	Topic	Page No.
1	Introduction	1
2	Motivation	1
3	Problem statement	1
4	Technology description	1-2
5	Our developing procedure	2-3
6	Limitations	4
7	Future scope	4
8	Discussion	5
9	Conclusions	5

1. INTRODUCTION

A **dictionary** is an application that allows users to search for a specific word and provides the meanings of the word in return.

For Students, Teachers and any individual across the world. The dictionary plays a significant role in their life in a variety of ways.

For instance, with the help of dictionaries, Students can understand their subjects better, improve their communication, and improve their grades by using the words correctly. The first ever dictionaries are in the form of books, available even now in any bookstore. But they consume a lot of time in searching and the risk of losing them or pages being torn out is high. With the evolution of technology, it is now much easier to search the unfamiliar words on search engines such as Google, etc., with an active internet connection. This advancement is much efficient than the custom book form, yet is far from the best. This is because of some factors like battery consumption, internet charges, etc., that are evident with using search engines. Thus, the need for developing an dictionary app is in excess.

We will create an English-to-English dictionary where the user will find all advanced features of the dictionary in a single App.

2. Motivation

Create an English dictionary using Python and the Django framework for a fulfilling project that improves your language abilities, gives you practical Django experience, and enables you to provide an important tool for authors and language learners.

3. Problem Statement

English Dictionary using Python with Django framework.

4. Technology Description

The English Dictionary web app project is built using –

- a) Python programming language
- b) HTML
- c) Django web framework
- d) PyDictionary module
- e) PyCharm
- f) Google Chrome

Python Programming language: Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

HTML: HTML or Hypertext Markup Language, is a markup language for the web that defines the structure of web pages. HTML consists of a series of elements. HTML elements tell the browser how to display the content.

Django Web Framework: The Django web framework provides the app's structure, handling the user interface, data storage, and application logic.

PyDictionary: PyDictionary is a Dictionary Module for Python 2/3 to get meanings words. It uses WordNet for getting meanings.

PyCharm: PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web and data science development.

Google Chrome: Google Chrome browser is a free web browser used for accessing the internet and running web-based applications.

5. Our developing procedure

1. Creating new directory
2. Creating virtual environment
3. Activating virtual environment
4. Creating the project and application
5. Installing Django

```
Command Prompt
Microsoft Windows [Version 10.0.22621.1848]
(c) Microsoft Corporation. All rights reserved.

C:\Users\asus>cd documents
C:\Users\asus\Documents>mkdir dictionary
C:\Users\asus\Documents>cd dictionary
C:\Users\asus\Documents\dictionary>py -m venv venv
C:\Users\asus\Documents\dictionary>venv\scripts\activate.bat
(venv) C:\Users\asus\Documents\dictionary>pip install django
Collecting django
  Using cached Django-4.2.2-py3-none-any.whl (8.0 MB)
Collecting asgiref<4,>=3.6.0
  Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1
  Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Collecting tzdata
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.7.2 django-4.2.2 sqlparse-0.4.4 tzdata-2023.3

[notice] A new release of pip available: 22.3.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) C:\Users\asus\Documents\dictionary>python.exe -m pip install --upgrade pip
```



```

settings.configure() before accessing settings.)).

(venv) C:\Users\asus\Documents\dictionary>django-admin startproject dictionary

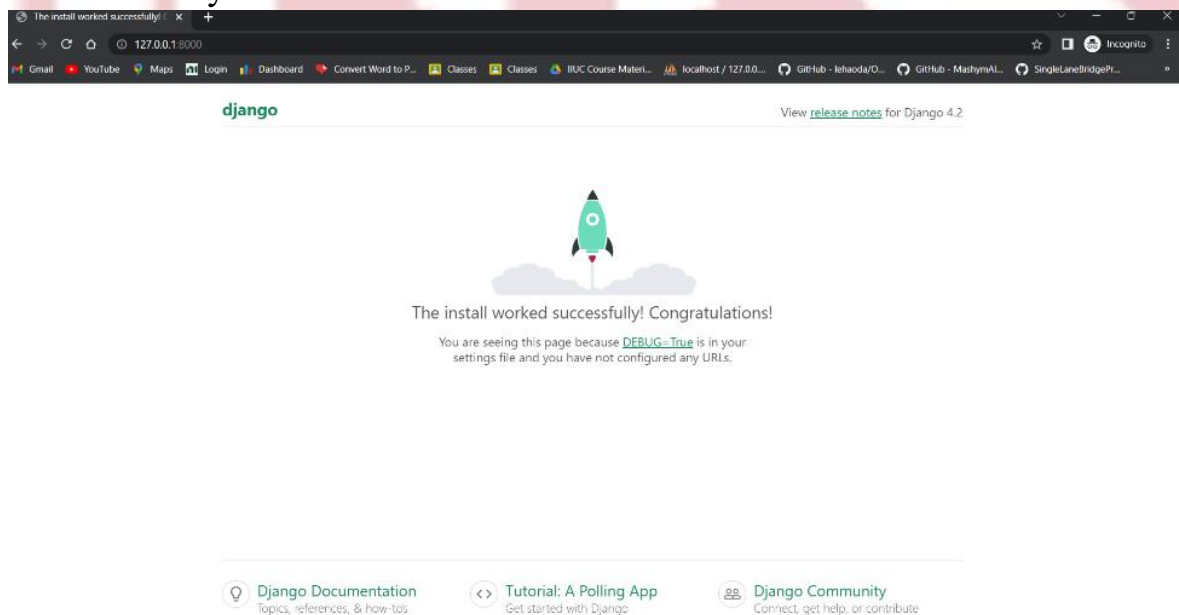
(venv) C:\Users\asus\Documents\dictionary>cd dictionary

(venv) C:\Users\asus\Documents\dictionary\dictionary>python manage.py startapp englishdictionary

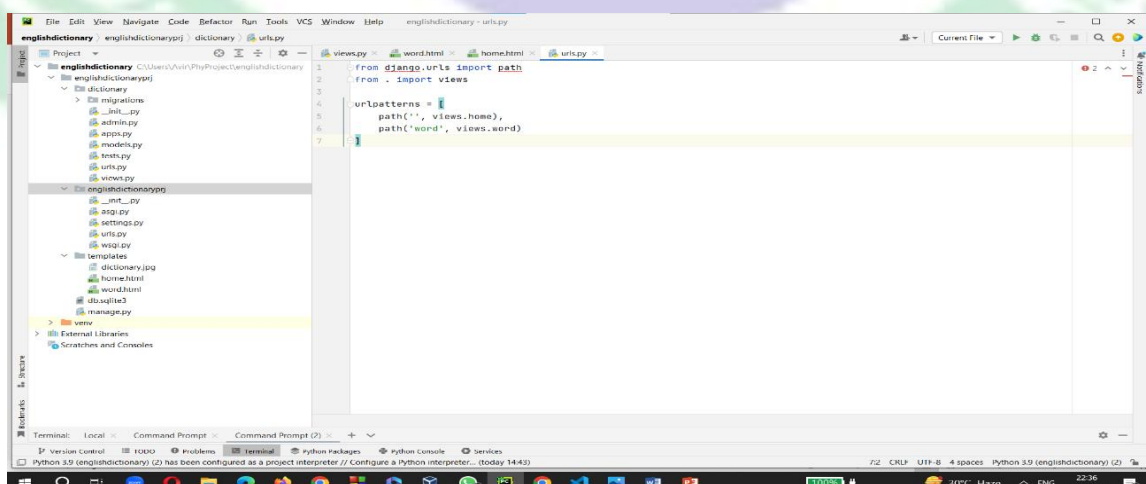
(venv) C:\Users\asus\Documents\dictionary\dictionary>

```

6. Successfully run server



7. Setting up the App in the Project Folder
8. Configuring the App URLs
9. Creating Views
10. Creating the HTML Templates
11. Integrating the Word Search Functionality



6. Limitations

Like any other technology project, the English dictionary web app project using Python and Django framework has its limitations. Here are some of the potential limitations that might encounter:

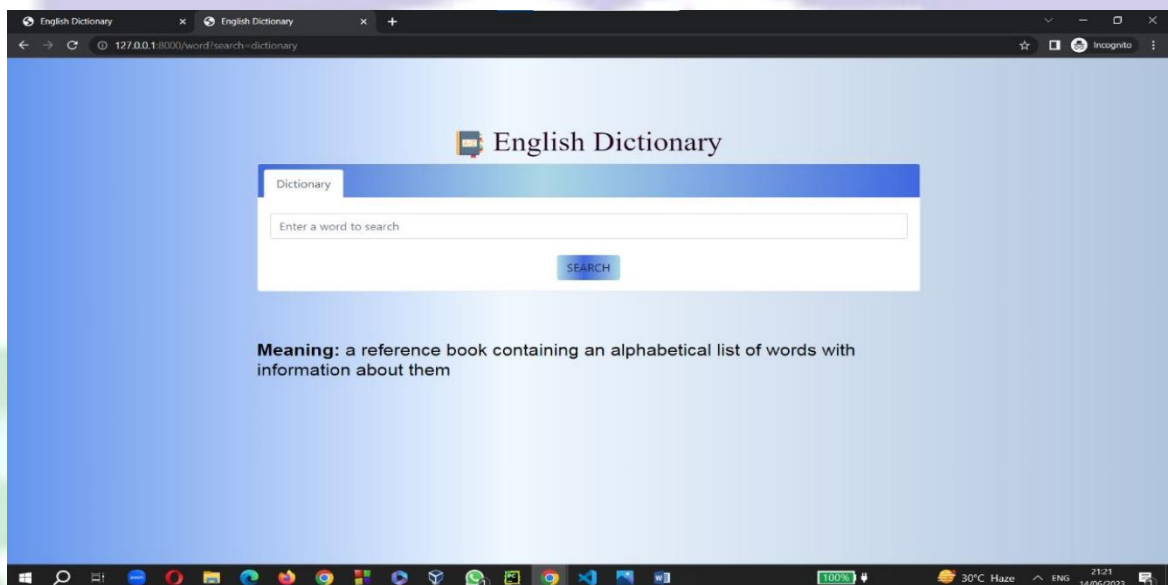
- a) Like any web scraping or data extraction method, the accuracy of the data extracted by PyDictionary or the web scraping process used in the project depends on the quality of the source website and how well the scraper is written. Inaccurate or outdated data can be displayed in the app, which can affect the user experience.
- b) The app is designed to provide only the basic information about an English word, such as its meaning. Other advanced functionalities, such as , pronunciation, parts of speech, synonyms, and antonyms or word origins, are not included in this app.
- c) The app is built for a small-scale project, and it might not be suitable for larger-scale applications. The app might encounter performance issues when handling a large number of requests or users simultaneously.

It's essential to be aware of the limitations and address them accordingly to ensure that the app functions smoothly and provides accurate and relevant information to users.

7. Future scope

- a) Advanced Search and Filtering
- b) User Profiles and Customization
- c) Audio Pronunciations
- d) Mobile Applications and API Integration
- e) Natural Language Processing Enhancements

8. Discussion



9. Conclusions

In conclusion, building an English dictionary using Python with the Django framework provides a powerful and flexible solution. Python's extensive libraries and Django's robust features allow developers to create a comprehensive and user-friendly dictionary application. With efficient search functionality, a visually appealing user interface, and secure authentication, the dictionary becomes an invaluable tool for users. The maintainable codebase and adherence to best practices ensure scalability and the potential for future feature enhancements.