Orion Project Report

Ameer A. Almaamari
391115994

This course is taught and supervised by Dr. Mohammad Alsuhibani

About the Project

The project was accomplished in (42) days, varies from learning process to applying the knowledge ending with approaching the key-elements of the project including the GUI. It was programmed and designed quite differently than an ordinary University project, that will be noticed throughout the report.

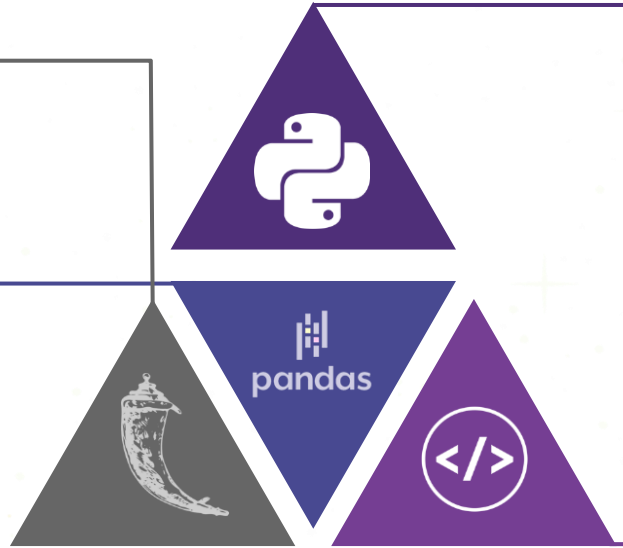
Main used tools

Flask

A web framework that provides a wide range of tools and libraries for web applications, it is more pythonic than any other framewok

Pandas

A powerful library for Python to analyze and manipulate data



Pure Python

The main programming language, it is the core of the project.

HTML , CSS , Javascript

The most popular tools for web development and front-end appearance

Main Key Features of the Project

Used Data File

CSV is a file which saves data in a tubular format, and it was used for its great compatibility with Pandas.

orion.csv

How Data is Saved and Called ?

Data is generated by using a user defined function in python and an empty data frame is built to handle the data and save them as a CSV file with using “to_csv” built in function of Pandas.

generate_data

The Way of Approaching Data

To meet the requirements, some of the most powerful data analysis tools are used. In “update_data” function, I manipulated the saved data to handle the requirements, such as the most read books, the rank of each member and a lot more.

update_data

Dummy Data ?

What if the user wants to generate their own randomized data ? no worries, they can enter the number of records and the program will create random data and use all of the features that the data file enjoys.

dummy_data.csv

The codes that bring Orion to life

How data is saved and called ?

```python
def generate_data():
    if request.method == "POST":
        names = ["Ahmed", "Ali", "Mohammed", "Sami", "Anger", "Jana", "Anwar", "Abdullah", "Tareq","Osama","Khaldon"]
        books = [{"Name":"Game Of Thrones", "Genre":"Fantasy", "Pages":350},
            {"Name":"Harry Poter", "Genre":"Fantasy", "Pages":1000},
            {"Name":"Algorithms", "Genre":"Education", "Pages":220},
            {"Name":"The Serpent Garden", "Genre":"Romance", "Pages":120},
            {"Name":"Say Yes to Yourself and No to Others", "Genre":"Self Dev", "Pages":80},
            {"Name":"Romeo and Juliet", "Genre":"Romance", "Pages":410}]

        users = [{"Name":name, "Email":f"{name}@orion.com", "Phone":f"05"+str(randint(10000000,99999999))} for name in names]

        data = pd.DataFrame(columns=["Name","Email","Phone","Book","Genre","Pages Read"])
        for sample in range(int(request.form["num_samples"])):
            user = choice(users)
            book = choice(books)
            pages_read = randint(5,book["Pages"]*0.5)
            data = data.append({'Name': user["Name"], 'Email': user["Email"], 'Phone': user["Phone"],
                                "Book":book["Name"], "Genre":book["Genre"], "Pages Read":pages_read},
                               ignore_index=True)
        data.to_csv('projectOrion/data/dummy_data.csv')
        return redirect(url_for('dashboard'))
```

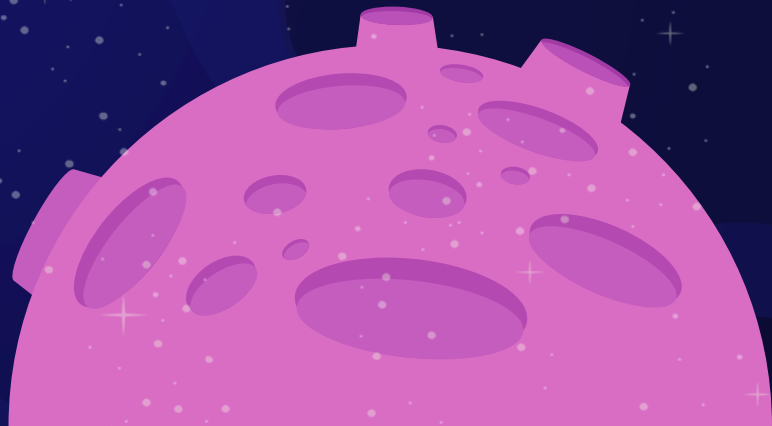
The way of approaching the data

```python
def update_data():
    # Load the old data file:
    data = pd.read_csv('projectOrion/data/data.csv', index_col=0).reset_index(drop=True)
    if os.path.isfile('projectOrion/uploads/last_upload.csv'):
        # Load the new data file:
        uploaded_data = pd.read_csv('projectOrion/uploads/last_upload.csv', index_col=0).reset_index(drop=True)
        # Concatenate old and new data together:
        data= pd.concat([data, uploaded_data]).reset_index(drop=True)
        # Save the concatenated file:
        data.to_csv('projectOrion/data/data.csv')

    # Count number of books read by members:
    num_books_read = len(pd.unique(data['Book']))

    # Count number of pages read by members:
    num_pages = sum(data['Pages Read'])

    # Put these two values in a table named Summary :
    summary = pd.DataFrame([str_x,{'Number of books read by members':[num_books_read],
                                   'Number of pages read by members':[num_pages]}])
    summary.to_csv('projectOrion/data/summary.csv')

    # Count books read by each member(Ranking of group members based on number of books read):
    temp = data[["Name","Book"]].groupby('Name').nunique()
    temp = temp.drop("Name", axis=1).reset_index(drop=False)
    book_per_user = temp.sort_values(by='Book', ascending=False).reset_index(drop=True)
    # Save this data to a table:
    book_per_user.to_csv('projectOrion/data/book_per_user.csv')
    # Plot the graph and save it to the images directory:
    book_per_user.plot(x="Name", y="Book", kind = "bar",
                       title = "Number of Books Read per User", color =(0.13, 0.12,0.48))
    plt.savefig('projectOrion/static/images/Number of Books Read per User.svg', bbox_inches='tight')

    # Ranking of group members based on number of pages read :
    user_sum = data[["Name","Pages Read"]].groupby(['Name']).sum().sort_values(by='Pages Read', ascending=False).reset_index()
    # Save this data to a table:
    user_sum.to_csv('projectOrion/data/user_sum.csv')
    # Plot the graph and save it to the images directory:
    user_sum.plot(x="Name", y="Pages Read", kind = "bar",
                  title = "Members Ranking", color =(0.13, 0.12,0.48))
    plt.savefig('projectOrion/static/images/Members Ranking.svg', bbox_inches='tight')

    # Ranking of books mostly read by the group members
    book_sum = data[["Book","Pages Read"]].groupby(['Book']).sum().sort_values(by='Pages Read', ascending=False).reset_index()
    # Save this data to a table:
    book_sum.to_csv('projectOrion/data/book_sum.csv')
    # Plot the graph and save it to the images directory:
    book_sum.plot(x="Book", y="Pages Read", kind = "bar",
                  title = "Books Ranking", color =(0.13, 0.12,0.48))
    plt.savefig('projectOrion/static/images/Books Ranking.svg', bbox_inches='tight')

    # Ranking of books categories mostly read by the group members
    genre_sum = data[["Genre","Pages Read"]].groupby(['Genre']).sum().sort_values(by='Pages Read', ascending=False).reset_index()
    # Save this data to a table:
    genre_sum.to_csv("projectOrion/data/genre_sum.csv")
    # Plot the graph and save it to the images directory:
    genre_sum.plot(x="Genre", y="Pages Read", kind = "bar",
                   title = "Categories Ranking", color =(0.13, 0.12,0.48))
    plt.savefig('projectOrion/static/images/Categories Ranking.svg', bbox_inches='tight')

# This function loads the summary csv file:
def get_summary():
    summary = pd.read_csv('projectOrion/data/summary.csv', index_col=0)
    return list(summary.iloc[0])

# This function loads the user summary csv file:
def get_user_sum():
    user_sum = pd.read_csv("
                           'projectOrion/data/user_sum.csv', index_col=0)
    return user_sum

# This function loads the category (genre) summary csv file:
def get_genre_sum():
    genre_sum = pd.read_csv('projectOrion/data/genre_sum.csv', index_col=0)
    return genre_sum

# This function loads the book summary csv file:
def get_book_sum():
    book_sum = pd.read_csv('projectOrion/data/book_sum.csv', index_col=0)
    return book_sum

# This function loads the file contains the ranking of group members based on number of books read:
def get_book_per_user():
    book_per_user = pd.read_csv('projectOrion/data/book_per_user.csv', index_col=0)
    return book_per_user
```

Used libraries and tools

```python
from flask import render_template, url_for, flash, redirect, request, jsonify
from projectOrion import app, ALLOWED_EXTENSIONS
from werkzeug.utils import secure_filename
#import pandas library's functions
import pandas as pd
#
import os
import datetime
import matplotlib.pyplot as plt
from random import choice, randint
#Using style sheets of Matplotlib
plt.style.use('fivethirtyeight')
```

what about visualization ?



Key Features of the Front-End

Web tools
HTML , CSS and Java Script are the core of converting the data to a graphical representation

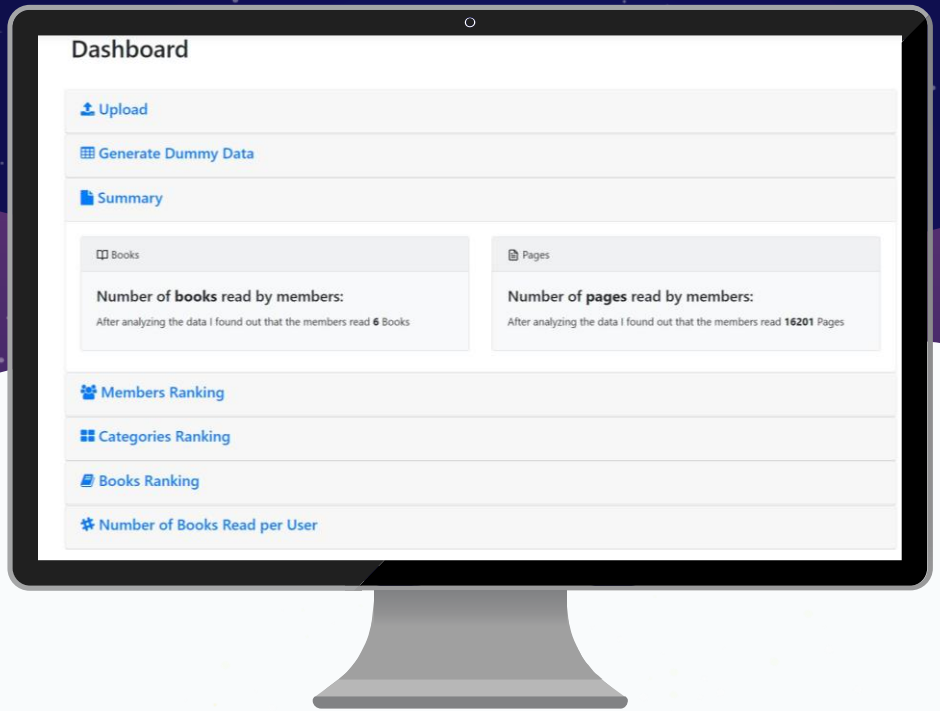
Bootstrap
A framework which used to provide a visual design for the project.

Flask
A solid microframework allows us to create servers in Python.

Matplotlib
a comprehensive library for creating static, animated, and interactive visualizations in Python

Home

This page will be shown once the server is opened, the main page contains the project's name and the developer name as well as a dynamic button to the dashboard.



Dashboard

The page that gathers all of the required data and their analysis, starting from a summary moving to the ranking of each elements and ending with the number of books read per user, also as an additional feature, generating dummy data is available if the user wants to experience different results.

note The content of this page does not fully appear in the picture, the full content will be displayed in the presentation.

note generating dummy data will be saved as a CSV file then the user has to upload it manually.

Analysed Data

Summary

Books

Number of books read by members:

After analyzing the data I found out that the members read **6** Books

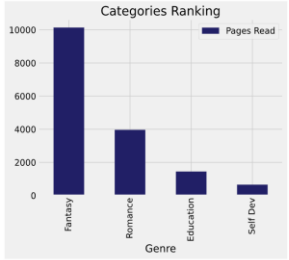
Pages

Number of pages read by members:

After analyzing the data I found out that the members read **16201** Pages

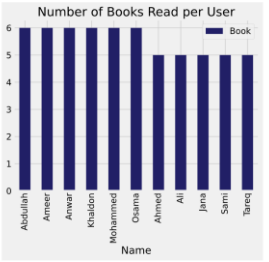
Categories Ranking

#	Genre	Pages Read
1	Fantasy	10138
2	Romance	3962
3	Education	1447
4	Self Dev	654



Number of Books Read per User

#	Name	Book
1	Abdullah	6
2	Ameer	6
3	Anwar	6
4	Khaldon	6
5	Mohammed	6
6	Osama	6
7	Ahmed	5
8	Ali	5
9	Jana	5
10	Sami	5
11	Tareq	5

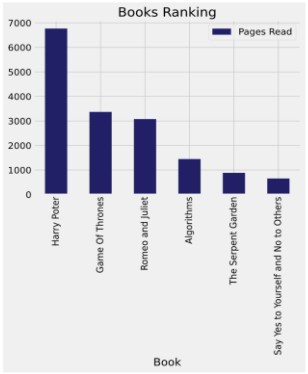


Members Ranking

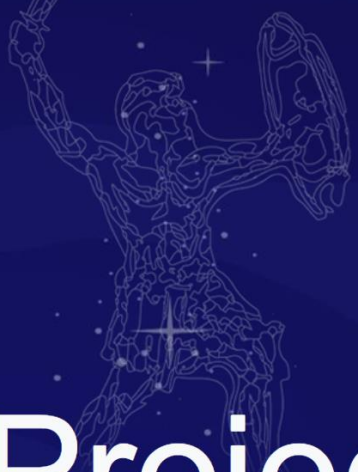
#	Name	Pages Read
1	Osama	2888
2	Khaldon	2350
3	Abdullah	1748
4	Anwar	1690
5	Ameer	1402
6	Jana	1298
7	Mohammed	1278
8	Sami	1268
9	Ahmed	1044
10	Tareq	849
11	Ali	386

Books Ranking

#	Book	Pages Read
1	Harry Poter	6769
2	Game Of Thrones	3369
3	Romeo and Juliet	3077
4	Algorithms	1447
5	The Serpent Garden	885
6	Say Yes to Yourself and No to Others	654



THE LAST PAGE, ABOUT !



Orion Project

This project is an assignment for CS213 course in Qassim University

All of the details and source code are provided in a GitHub Repository

Home | dashboard

THANKS!

Thank you for providing us with this outstanding project which enhance our experience and ability to express our thoughts in programming languages.

391115994@qu.edu.sa