

San Diego State University

Final Project Technical Documentation



**SAN DIEGO STATE
UNIVERSITY**

Omid Gholizadeh and Ridge Del Rio

MIS 315

Professor Doiron

15 December 2022

Table of Contents

Problem	3
Technical Summary	3
How to use	4
Folder and File Information	6
How to use cloud_script.py	8

Problem

Our project aims to assist users with identifying the high and low times for the Aztec Recreation Center (ARC). The ARC is an on-campus facility that provides students with fitness equipment and sport facilities. The ARC is very popular among students and can get busy during certain hours of the day. The problem that our program aims to solve is the issue with not knowing how busy the gym is going to be at a given time. Attending a workout facility that is jam-packed leads to longer workout time due to the unavailability of certain equipment or facilities. The majority of college students have limited time and a small window in their day to exercise. Our program gives the user information to find the most optimal time for them to attend the ARC, allowing them to work around their busy schedule. The high and low times can be looked at based on specific venues that the user may specify. For instance, if the user wants to look at the high and low times for just the climbing area, they can do that.

Technical Summary

To ensure the reliability of our data collection process, we have implemented a solution with triple redundancy by deploying three separate cloud servers. The servers are hosted by Amazon Web Services (AWS), Microsoft Azure, and Oracle Cloud Infrastructure (OCI). Each is running a copy of our data collection script. This approach ensures that our data remains secure and can be accessed in the event our servers experience downtime. To collect and process the data for this project, we leveraged Selenium Chrome Driver and BeautifulSoup to scrape and parse the HTML from the ARC's website. This allowed us to download occupancy data for each of the 14 venues and use it for our analysis. From there, we organized our data into a database optimized to be easily read and analyzed using a Pandas data frame, which we use to filter and sort through our data. To facilitate ease of use for end users, we have implemented a state-of-the-art graphical user interface (GUI). The GUI aids in guiding a user to generate a report, which includes data in both CSV and line plot formats.

How to use

1. If this is your first time using Python please look at this video:
Make sure to download and run Python 3.11 or later
<https://youtu.be/YYXdXT2l-Gg> and come back to this document
If you're running MacOS Catalina or newer, you'll need to edit the .zprofile file instead of .bash_profile to add the Python alias*
2. How to install an IDE:
 - a. Windows: <https://youtu.be/lM5Y7BnP56k>
 - b. Mac: <https://youtu.be/K5cAu-Wro3M>
3. Once familiar with the basics of running python and have a working IDE
4. Open the MIS_Final_Project folder in a Python IDE of your choice:
 - a. If your IDE has a built-in Terminal:
 - i. Navigate to it and run the following commands in the terminal:
pip3 install numpy==1.23.5
pip3 install pandas==1.5.2
pip3 install -r requirements.txt
pip3 install matplotlib==3.5.3
python3 age_verification.py
These must be done in the correct order
 - ii. Please ensure there are no spaces before or after the command
 - b. If your IDE has NO built-in Terminal:
 - i. Windows
 1. In the bottom left corner of your screen please locate the search prompt and type: Command Prompt
 2. Press enter to open a new Command Prompt window
 3. Type in "cd" followed by the path to the project folder
 4. For example:
cd C:\Users\USERNAME\PycharmProjects\MIS_Final_Project
 5. Please ensure the appropriate path is utilized for your specific setup and press enter
 6. Next run the following commands in the terminal one by one,
pip3 install numpy==1.23.5
pip3 install pandas==1.5.2
pip3 install -r requirements.txt
pip3 install matplotlib==3.5.3
python3 age_verification.py
These must be done in the correct order

7. Once done please close the command prompt window
- ii. Mac
 1. On the keyboard please press, Command and Spacebar at the same time and type: Terminal
 2. Press enter to open a new terminal window
 3. Type in "cd" followed by the path to the project folder
 4. For example:
`cd /Users/USERNAME/PycharmProjects/MIS_Final_Project`
 5. Please ensure the appropriate path is utilized for your specific setup and press enter
 6. Next run the following commands in the terminal one by one,
`pip3 install numpy==1.23.5`
`pip3 install pandas==1.5.2`
`pip3 install -r requirements.txt`
`pip3 install matplotlib==3.5.3`
`python3 age_verification.py`
 These must be done in the correct order
 7. Run the command and wait for all packages to finish installation
 8. Once done please close the command prompt window
5. Select an valid interpreter if needed
6. Within the project folder, navigate to "age_verification.py" and run the file
7. The "Age Verification" window will display, please enter appropriate birthdate using dropdown menus
8. Select either the "All Dates" or "Date Range" button
 - a. If "All Dates" button is selected:
 - i. Analysis will be ran on all applicable dates
 - b. If "Date Range" button is selected:
 - i. Additional dropdown menus and Buttons will appear
 - ii. Select a starting date
 - iii. Click "Submit Start Range" button
 - iv. Select an ending date
 - v. Click "Submit End Range" button
9. Once ALL Dates or Date Range menus are completed:
 - a. Select a venue from the dropdown menu
 - b. Click "Submit" button to run analysis
10. Output data gets stored in "data.csv" inside the output folder
 - a. The data is organized in order from least occupied to most occupied times
 - b. Times are in 24hr format
11. A graph will be generated in "graph.png" inside the output folder with a visual representation of the report (Times are in 24hr format)

Folder and File Information

1. Contents of the “ProjectFolder” folder:
 - a. “database” folder:
 - i. ARC_Image.png
 1. This file contains the image used in the ARC Gym Analysis Tkinter window
 - ii. cloud_script.py
 1. cloud_script.py is not utilized for the normal operation of the program and was only used to gather data from the ARC’s website
 2. If you would like to continue adding data to your database you may use this script for such purpose
 3. Detailed information on how to use cloud_script.py will be provided towards the end of this document
 - iii. database.csv
 1. This is a .csv file that contains all the information that cloud_script.py is gathering
 - b. “output” folder:
 - i. data.csv
 1. Contains the output data based on the user specified parameters
 2. This data is organized by count
 - ii. graph.png
 1. Contains the graphical visualization of the data analysis
 - c. “documentation” folder:
 - i. Complete_Documentation.pdf
 1. Contains a copy of the technical documentation of this project
 - ii. README.txt
 1. Contains a quickstart guide on how to quickly use the program, specifically designed for users that are proficient in python
 - d. age_verification.py
 - i. Script that allows the user to enter their birthdate through a Tkinter window
 - ii. If the user is under 18 years of age, the program will exit and prompt the user with a message box that informs them of not meeting the age requirement
 - iii. If the user is above 18 years of age, the Tkinter window will update with the ARC Gym Analysis window that is scripted in main.py
 - e. main.py
 - i. Script that, upon meeting the age requirement, displays to the user the main GUI that is intended to be interacted with for this project

- ii. Used for allowing the user to pick the dates and venues for their analysis
 - iii. Calls statistics.py once completed
 - iv. Passes along user specified parameters
- f. requirements.txt
 - i. Includes a list of modules and their corresponding version number for the user to download to ensure full program functionality
- g. statistics.py
 - i. Loads our database into a pandas dataframe while dropping duplicates
 - ii. Formats the columns into an optimized format for statistical analysis
 - iii. Filters through parameters through status, venue, start range, and end range
 - iv. Calculates bottom and top quartiles, along with the average
 - v. Plots data on a line plot, with time on the x-axis and occupancy count on y-axis
 - vi. Sorts data by low to high occupancy and saves the output as a .csv file

How to use cloud_script.py

1. Please ensure that google chrome is installed on your computer
 - a. For instructions on how to install chrome please click the appropriate link below
 - i. Windows: https://youtu.be/Ew_ReoK1zMc
 - ii. Mac: <https://youtu.be/x8ZA2Cquo4Q>
2. Once chrome is installed please open chrome and open the settings menu by clicking the three stacked dots on the upper right hand corner of the chrome window and select the settings tab
3. Once chrome settings is opened navigate to the About Chrome tab in the bottom left hand menu of the settings site
4. Take note of the first three digits of the version
5. Navigate to this website: <https://chromedriver.chromium.org/downloads>
6. Navigate to the current releases section and click on the link associated with your current chrome version you obtained from previous steps
7. Click on the appropriate download button for your machine.
 - a. If you have a Windows machine please select, “chromedriver_win32.zip”
 - b. If you have an Intel-based mac please select, “chromedriver_mac64.zip”
 - c. If you have an Arm-based M1 or M2 Mac please select, “chromedriver_mac_arm64.zip”
8. Unzip the file if necessary and place the executable file in a location of your choice
9. On line 14 of cloud_script.py in the database folder please paste in the absolute file path of your chromedriver executable inside the parentheses marked PATH TO CHROMEDRIVER
10. Run cloud_script.py
 - a. It will obtain occupancy data from the ARC’s website every 10 minutes and dump the data into database.csv with built in error mitigation and handling
 - b. On linux machines you can run the script with the following command to run the script in the background, without being terminated if the terminal window closes, and suppresses any output:
nohup python3 PATH TO CLOUD_SCRIPT.PY >/dev/null 2>&1 & disown