**CSCI 23000 Final Project Proposal**

Prayer Timer

OVERVIEW

This tool is intended for those who want to view Islamic prayer times. It is a simple way to get all the prayer times without having to create an account or click through ads like other apps. It will be catered towards all Muslims, especially those who attend Geist Community Center. While Geist Community Center has their own website, at the time of writing this, they do not have their own app. The prayer timer will be one component of that app. For this project, I will be using an API to create functions to get each prayer times, among other things.

USE CASE

The user will be greeted by the home/menu page, which will allow them to move on to the prayer timer page. Since this project is only covering the prayer timer, the user will only be able to move to the prayer timer page. However, the entire project (my eventual honors project) will allow the user to move to other pages of the application as well.

Upon arriving at the prayer timer page, the user should be able to see:

* the current date
* a list of prayers and their respective times
* button to go back to the home page

DATA DESIGN

The key data structure will be a list of all the US capitals and states. Upon entering the app, the user will have to choose the appropriate state (and capital) from a drop-down list. This will affect the prayer times, as prayer times differ depending on where you are located.

The list of capitals and states will be used from this file: <https://github.com/jasperdebie/VisInfo/blob/master/us-state-capitals.csv>

They will then be used to create an array composed of the capitals and states. The array will remain static throughout, meaning that it will not change once it is created.

ALGORITHM

There will be several classes along with their respective methods in this program, such as:

**App() class:**

will initialize the actual app, while also using this [code](https://stackoverflow.com/questions/7546050/switch-between-two-frames-in-tkinter/7557028#7557028) to allow the user to go to other pages in the app.

**HomePage() class:**

shows the home screen to the user

* initialize required label(s), drop-down menu, and button(s) and output

**PrayerTimer() class:**

shows the prayer timer screen to the user and allows the user to select a location to get its respective prayer times.

consists of the following methods:

* **showLocMenu(): allows the user to pick a location from a dropdown menu**
  + destroys all the widgets in the frame
  + initializes and outputs necessary labels
  + submit button is created and when the user clicks on it, the program calls getPrayer(), getDate(), and showPrayerTimer()
  + home button is created and when the user clicks on it, the program changes the frame to the home page.
  + this function is usually called at the “end” of the program, when the program has already outputted prayer times and the date and the home button is clicked.
* **showPrayerTimer(): outputs the current date and prayer times depending on the location the user picked**
  + destroys all the widgets in the frame
  + initializes and outputs necessary labels, such as the labels for each prayer time
  + home button is created and when the user clicks on it, the program changes the frame to the home page.
  + this function is usually called after the user chooses a location and wants to see the prayer times for that respective city and state.
* **dropdownMenu(): creates a dropdown of locations ("capital, state" format) that the user can choose from; this DOES affect the prayer times.**
  + calls all necessary global variables
  + creates an OptionMenu using all the elements of the global variable location list and sets it to a variable named dropdown
  + dropdown is outputted, and the user’s current choice from dropdown is “traced” by using the getLocation() method
* **getLocation(): sets city and state global variables to the user's choice from the dropdown menu.**
  + calls all necessary global variables
  + city and state variables are set to the user’s choice using a for loop to filter out the comma.
  + print city and state variables to the console
* **getPrayers(): receives url of prayer time api, which will be used to output the respective prayer times depending on the city and state that the user picked.**
  + Calls all necessary global variables
  + Initializes url as the api and formats it by using the city and state the user picked
  + Gets the results of the api and prints them to the console
  + Converts results to json and prints that to the console
  + Sets each variable (fajr, thuhr, asr, maghreb, and isha respectfully) to their time using json
* **getDate(): receives url of prayer time api, which will be used to outpupt the current date ("weekday, month day year" format)**
  + calls global variable todaysDate
  + initializes url as the api and formats it by using the city and state the user picked
  + gets the results of the api, converts results to json, and sets it to apiDate
  + uses apiDate to set the weekDay, month, day, and year variables
  + sets todaysDate to a combination of those variables

KEY TECHNOLOGIES AND TECHNIQUES

* creating an app using tkinter
* using api to get data
* using the data to output appropriate prayer times + date

This project uses tkinter, which is in the default build of PythonAnywhere. The project will be written in Python 3.9 with tkinter 8.6.

UI DESIGN

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface

Description automatically generated with medium confidenceText

Description automatically generated

CITATIONS

The code for changing pages was derived from: <https://stackoverflow.com/questions/7546050/switch-between-two-frames-in-tkinter/7557028#7557028>