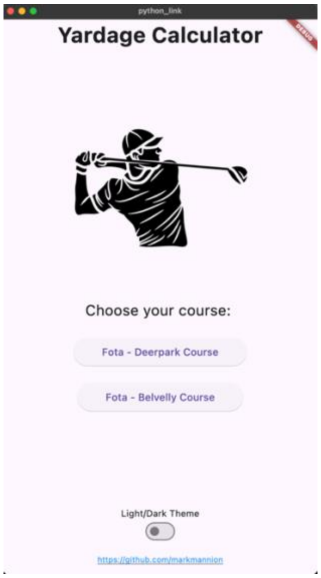
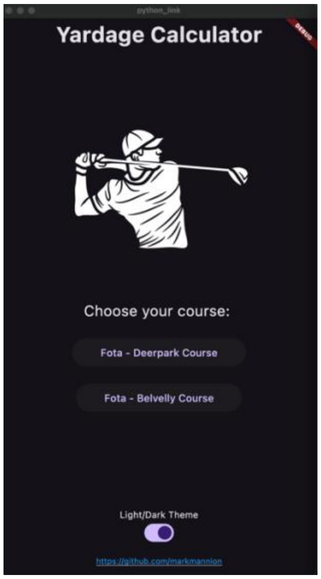
ISE Entrance Submission – Golf Yardage Calculator – Mark Mannion

Please find all code and images relating to my submission in the following GitHub Repository – https://github.com/markmannion/MarkMannion_ISE





Introduction

My entrance submission for the ISE 2025 entry is a personal passion project that I created which makes it easier for people to play better golf.

One of the main parts of improving your golf game is knowing what club to hit relative to the distance you have left on a hole. This enables you to hit better shots. Oftentimes the yardage that you have left into the green is a flat yardage without taking the slope of the hole, the wind speed or the temperature into account. My main aim was to create a quick and easy to use tool to bring all of these variables into the equation and make the task of gauging distance on the course a much simpler one.

Intitial Research

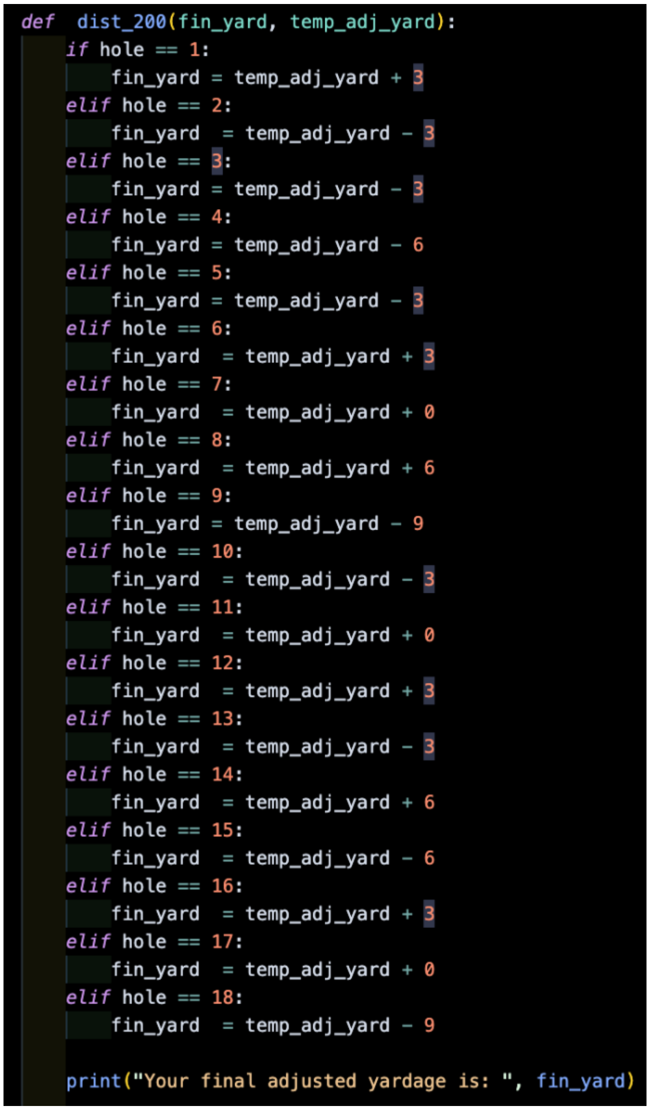
I conducted my own research as well as using secondary sources. To measure slope on the two courses at Fota Island Golf Club, I used a laser rangefinder (accurate to ± 0.5m over 1000m) to measure the slope of each of the holes on the courses from four set yardages: 200/150/100/50 yards. The data collection showed up to a 10 yard increase/decrease in yardage depending on slope.

I then investigated the effect temperature has on the flight of the golf ball. According to my research the standard carry distance for each golf club is when the temperature is at

approximately 24°C. For every increase/decrease of 5.5/6°C from that temperature, the carry distance increases/decreases by 2 yards.
The final variable investigated was wind, which has a significant impact on the carry distance of the golf ball. I discovered that for every 10 mph of tail wind/head wind the ball will carry 10 yards further or shorter.

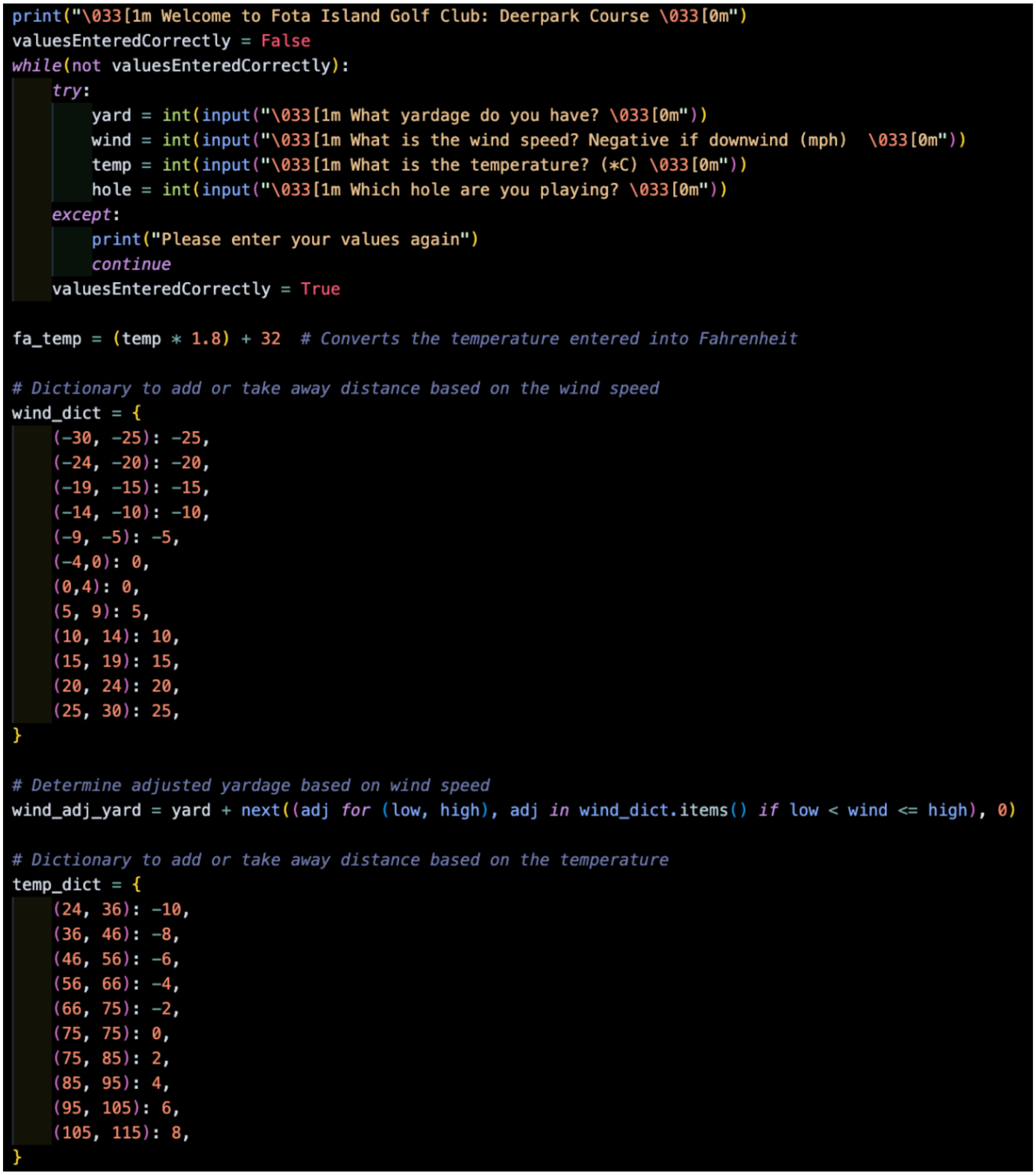
Creating the Programme

After gathering all of the data I required, I created my Python programme. I started by creating user inputted variables for temperature, wind speed, hole number and yardage. I then created If statements for wind and temperature. Also, I created four functions based on different length yardages - 200/150/100/50 yards. Finally, the programme suggests which club you should hit depending on the set clubs and corresponding distances that are in the programme. For example if you hit your 9 iron 160 yards and your 8 iron 170 yards the programme will recommend which club you should hit for the final adjusted yardage.



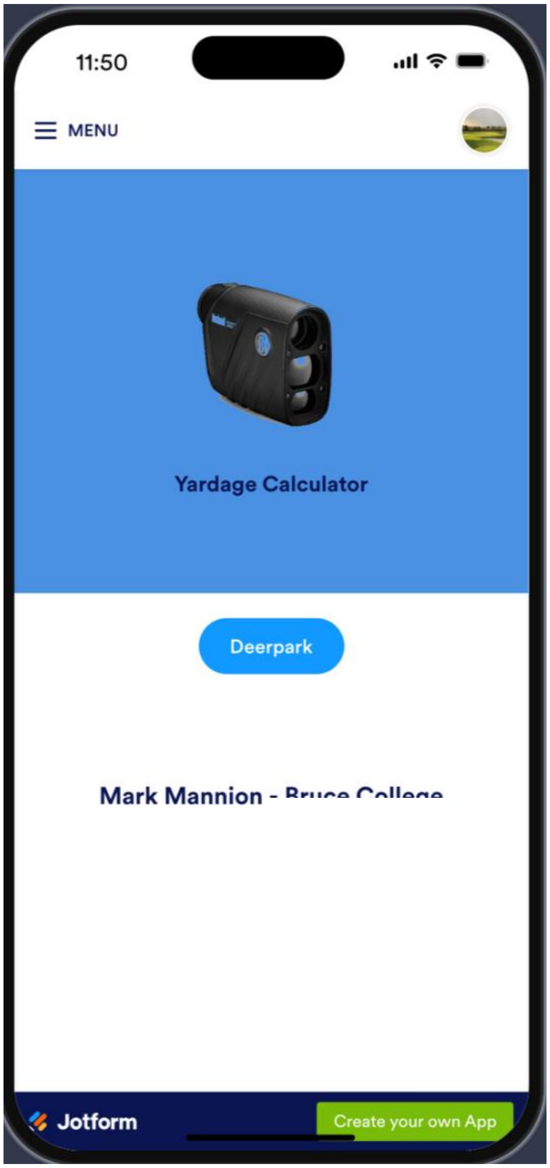
```
def  dist_200(fin_yard, temp_adj_yard):
    if hole == 1:
        fin_yard = temp_adj_yard + 3
    elif hole == 2:
        fin_yard  = temp_adj_yard - 3
    elif hole == 3:
        fin_yard = temp_adj_yard - 3
    elif hole == 4:
        fin_yard = temp_adj_yard - 6
    elif hole == 5:
        fin_yard = temp_adj_yard - 3
    elif hole == 6:
        fin_yard  = temp_adj_yard + 3
    elif hole == 7:
        fin_yard  = temp_adj_yard + 0
    elif hole == 8:
        fin_yard  = temp_adj_yard + 6
    elif hole == 9:
        fin_yard = temp_adj_yard - 9
    elif hole == 10:
        fin_yard  = temp_adj_yard - 3
    elif hole == 11:
        fin_yard  = temp_adj_yard + 0
    elif hole == 12:
        fin_yard  = temp_adj_yard + 3
    elif hole == 13:
        fin_yard  = temp_adj_yard - 3
    elif hole == 14:
        fin_yard  = temp_adj_yard + 6
    elif hole == 15:
        fin_yard  = temp_adj_yard - 6
    elif hole == 16:
        fin_yard  = temp_adj_yard + 3
    elif hole == 17:
        fin_yard  = temp_adj_yard + 0
    elif hole == 18:
        fin_yard  = temp_adj_yard - 9

    print("Your final adjusted yardage is: ", fin_yard)
```

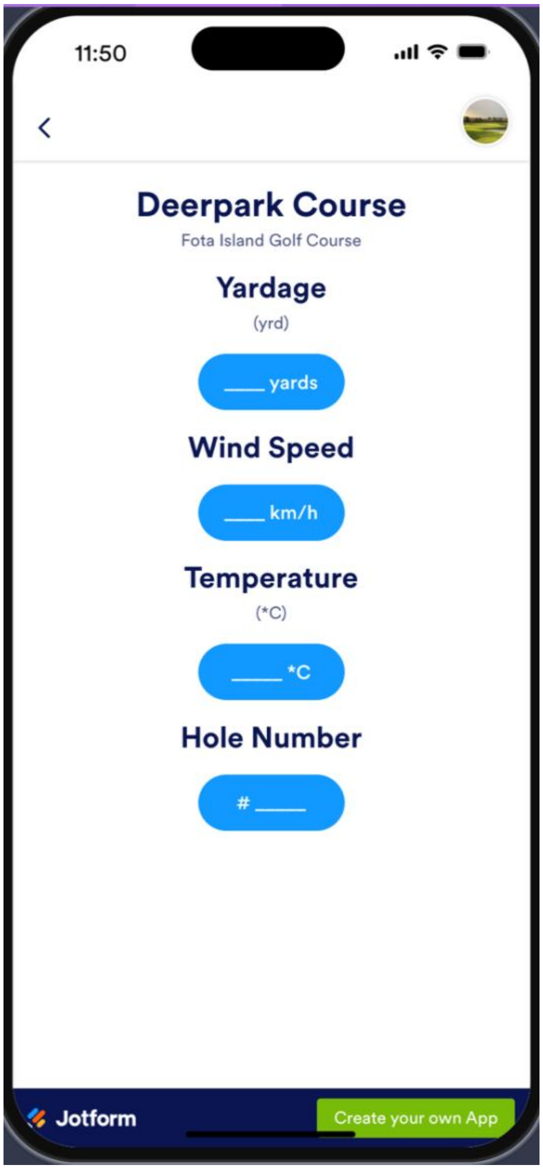
However, this code was inefficient and unwieldly so I replaced my If statements and functions with dictionaries. These dictionaries held my slope, temperature and wind information. I then created variables which determined which club to suggest. This was much more efficient and compact.



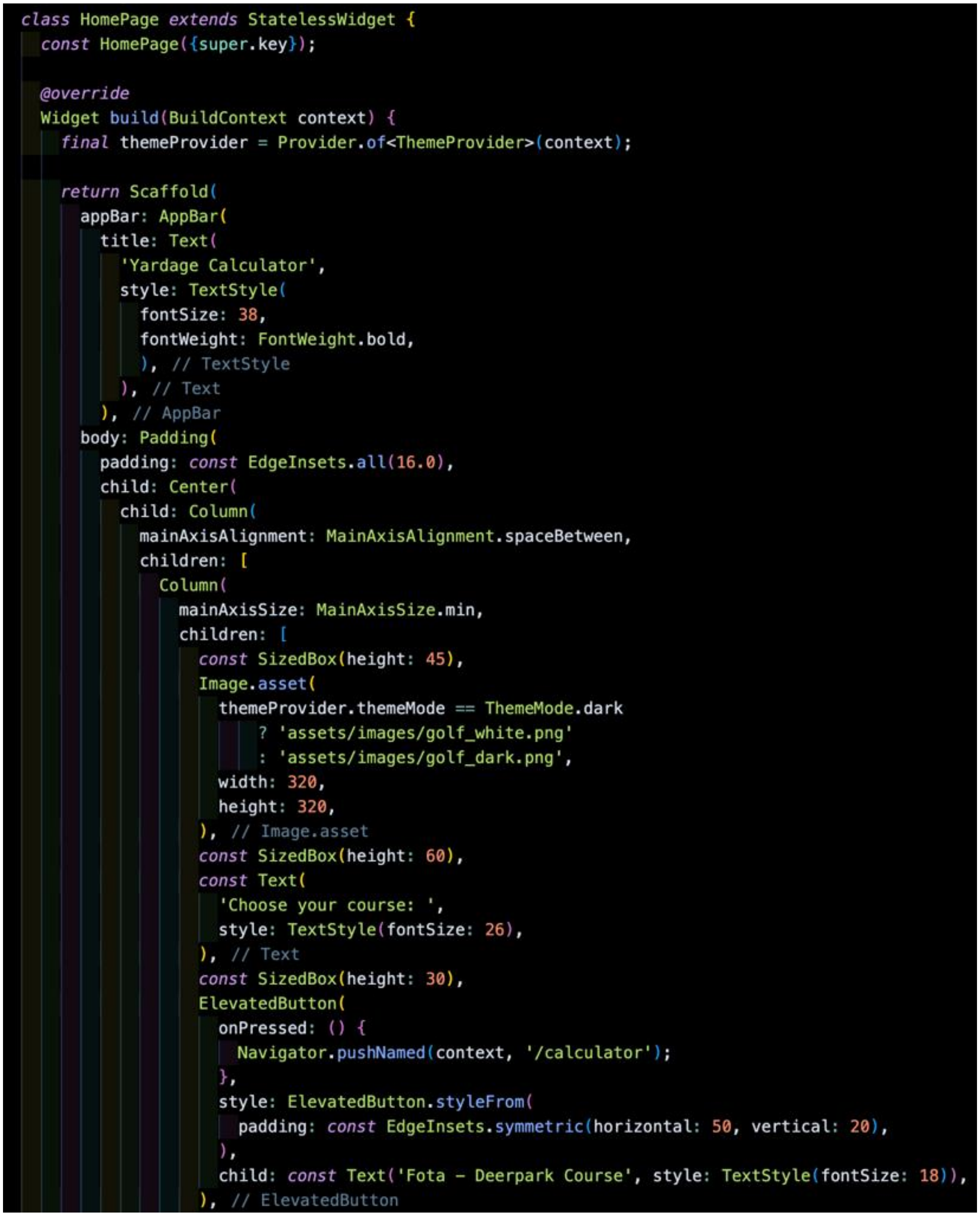
```
print("\033[1m Welcome to Fota Island Golf Club: Deerpark Course \033[0m")
valuesEnteredCorrectly = False
while(not valuesEnteredCorrectly):
    try:
        yard = int(input("\033[1m What yardage do you have? \033[0m"))
        wind = int(input("\033[1m What is the wind speed? Negative if downwind (mph)  \033[0m"))
        temp = int(input("\033[1m What is the temperature? (*C) \033[0m"))
        hole = int(input("\033[1m Which hole are you playing? \033[0m"))
    except:
        print("Please enter your values again")
        continue
    valuesEnteredCorrectly = True

fa_temp = (temp * 1.8) + 32  # Converts the temperature entered into Fahrenheit

# Dictionary to add or take away distance based on the wind speed
wind_dict = {
    (-30, -25): -25,
    (-24, -20): -20,
    (-19, -15): -15,
    (-14, -10): -10,
    (-9, -5): -5,
    (-4,0): 0,
    (0,4): 0,
    (5, 9): 5,
    (10, 14): 10,
    (15, 19): 15,
    (20, 24): 20,
    (25, 30): 25,
}

# Determine adjusted yardage based on wind speed
wind_adj_yard = yard + next((adj for (low, high), adj in wind_dict.items() if low < wind <= high), 0)

# Dictionary to add or take away distance based on the temperature
temp_dict = {
    (24, 36): -10,
    (36, 46): -8,
    (46, 56): -6,
    (56, 66): -4,
    (66, 75): -2,
    (75, 75): 0,
    (75, 85): 2,
    (85, 95): 4,
    (95, 105): 6,
    (105, 115): 8,
}
```

Upon creating these python scripts I decided to create an application using this yardage calculator. To do this I decided to use Flask for Python. This enabled me to insert my Python Flask App into whichever development interface I liked. I initially went with React Native (JS) as my interface of choice for app development, however I found that Flask integrated better with Flutter (Dart). Before beginning on the programming I created some prototypes on Jotform to give myself a design to work towards.



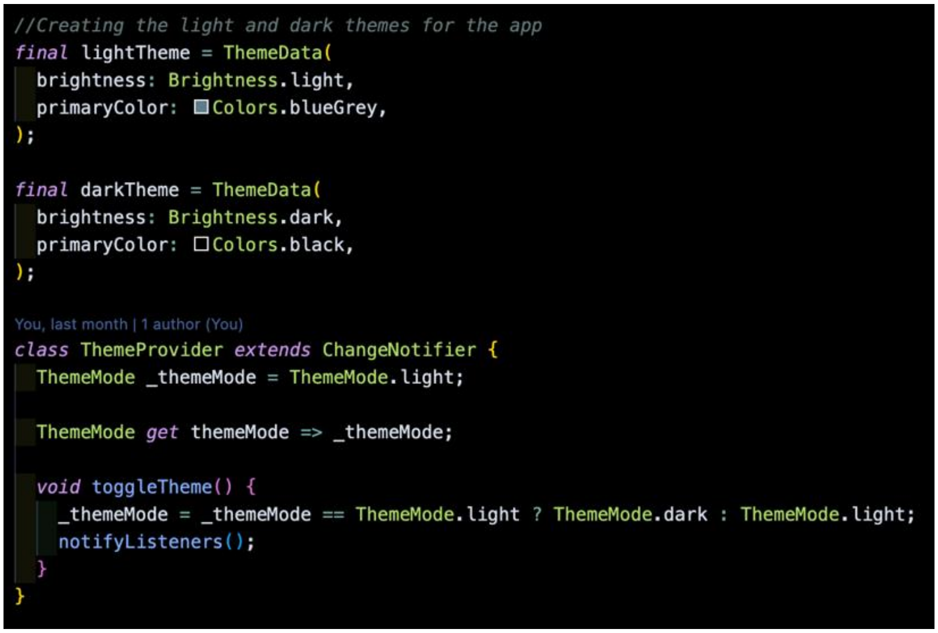


Firstly, I integrated my Flask app into my Flutter app by running both on a local network. This enabled me to use my Flask app as the backend to run the calculations. I then created multiple classes for my various pages.

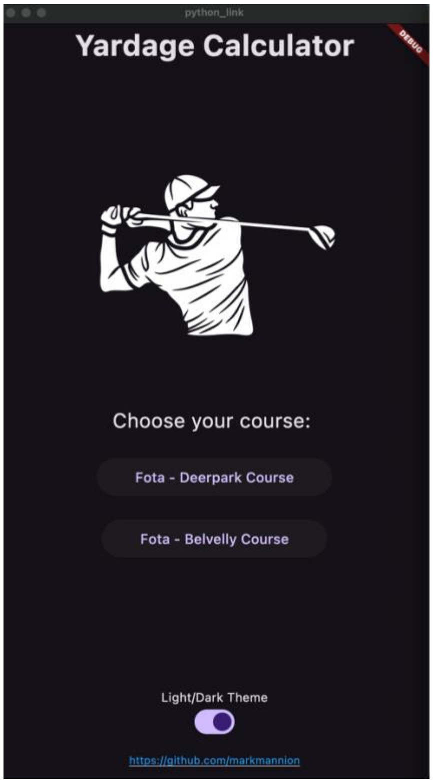


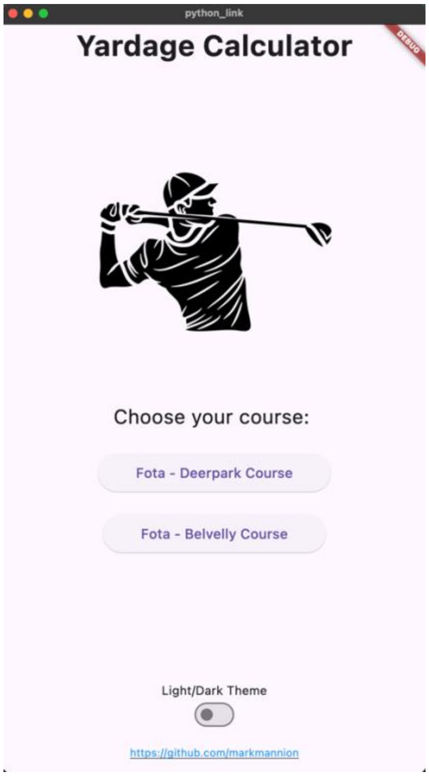
```dart
class HomePage extends StatelessWidget {
  const HomePage({super.key});

  @override
  Widget build(BuildContext context) {
    final themeProvider = Provider.of<ThemeProvider>(context);

    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Yardage Calculator',
          style: TextStyle(
            fontSize: 38,
            fontWeight: FontWeight.bold,
          ), // TextStyle
        ), // Text
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.spaceBetween,
            children: [
              Column(
                mainAxisSize: MainAxisSize.min,
                children: [
                  const SizedBox(height: 45),
                  Image.asset(
                    themeProvider.themeMode == ThemeMode.dark
                        ? 'assets/images/golf_white.png'
                        : 'assets/images/golf_dark.png',
                    width: 320,
                    height: 320,
                  ), // Image.asset
                  const SizedBox(height: 60),
                  const Text(
                    'Choose your course: ',
                    style: TextStyle(fontSize: 26),
                  ), // Text
                  const SizedBox(height: 30),
                  ElevatedButton(
                    onPressed: () {
                      Navigator.pushNamed(context, '/calculator');
                    },
                    style: ElevatedButton.styleFrom(
                      padding: const EdgeInsets.symmetric(horizontal: 50, vertical: 20),
                    ),
                    child: const Text('Fota - Deerpark Course', style: TextStyle(fontSize: 18)),
                  ), // ElevatedButton
```

After creating these classes, I created a light and a dark theme for the app to suit the user’s own preferences. These colour themes also affected the images in the app which changed to blend in with the current theme.

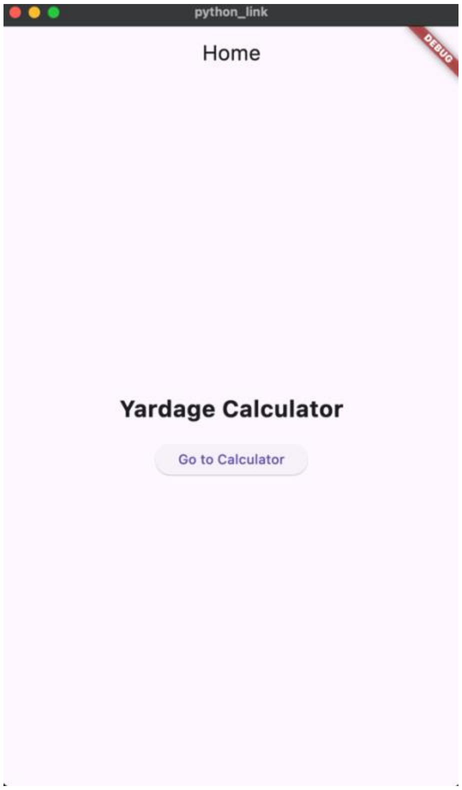


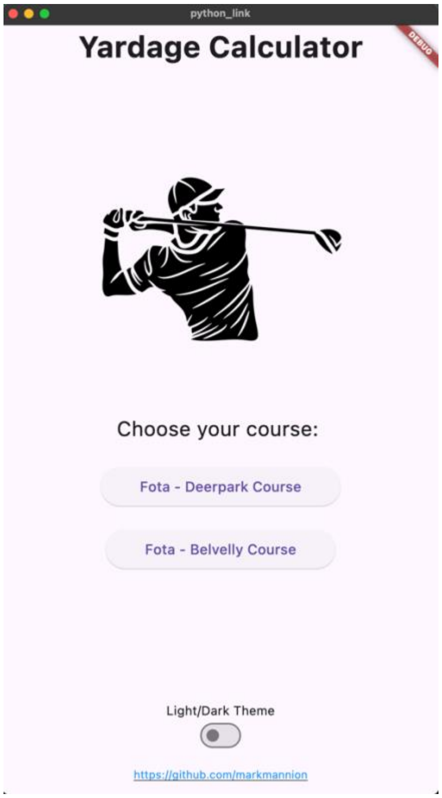
```
//Creating the light and dark themes for the app
final lightTheme = ThemeData(
  brightness: Brightness.light,
  primaryColor: ■Colors.blueGrey,
);

final darkTheme = ThemeData(
  brightness: Brightness.dark,
  primaryColor: □Colors.black,
);

You, last month | 1 author (You)
class ThemeProvider extends ChangeNotifier {
  ThemeMode _themeMode = ThemeMode.light;

  ThemeMode get themeMode => _themeMode;

  void toggleTheme() {
    _themeMode = _themeMode == ThemeMode.light ? ThemeMode.dark : ThemeMode.light;
    notifyListeners();
  }
}
```





Finally, I improved the layout and design of the homepage to enhance the user experience. This was done by changing text and interactive feature positioning.





Conclusion

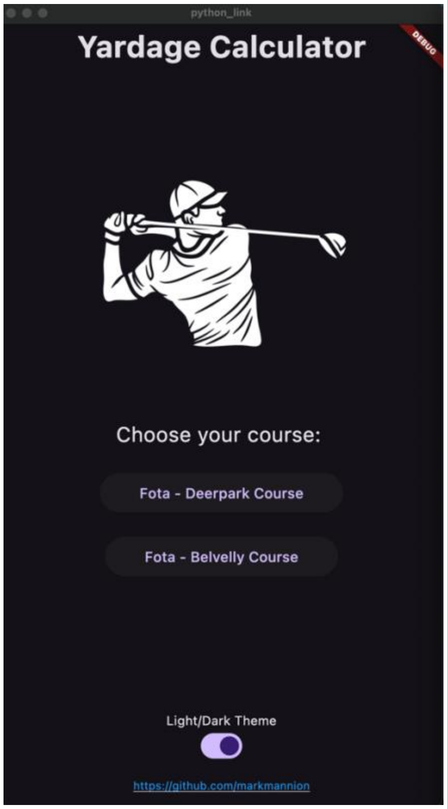
Overall I really enjoyed this project and learning how to use new languages. This was my first experience building an app, using Flask and also programming in Dart. Another aspect of using Flutter and Dart is that it enables the application to be built for both Android and iOS enabling cross platform development.

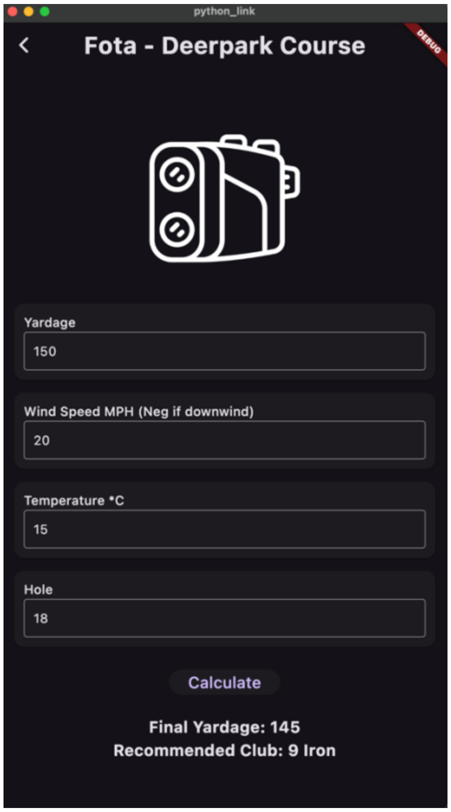
I showed this app to others and the feedback was that it was really interesting and intuitive to use. This helped refine the user design and experience.

I could also adapt and introduce a few things if I were to continue on with this project. I could introduce another page where the users would be able to set their own clubs and corresponding yardages and also integrate multilingual support for global use. Another improvement that could be introduced would be a method for users to submit their own courses into a community section. This would enable people around the world to build a database of downloadable course modules for others to use when playing new courses. With these additions the app could have commercial potential.

https://github.com/markmannion/MarkMannion_ISE

Finished Application





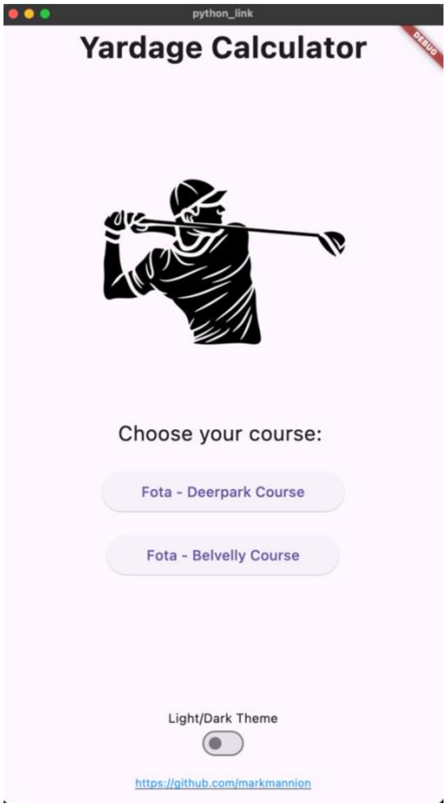
Yardage: 150

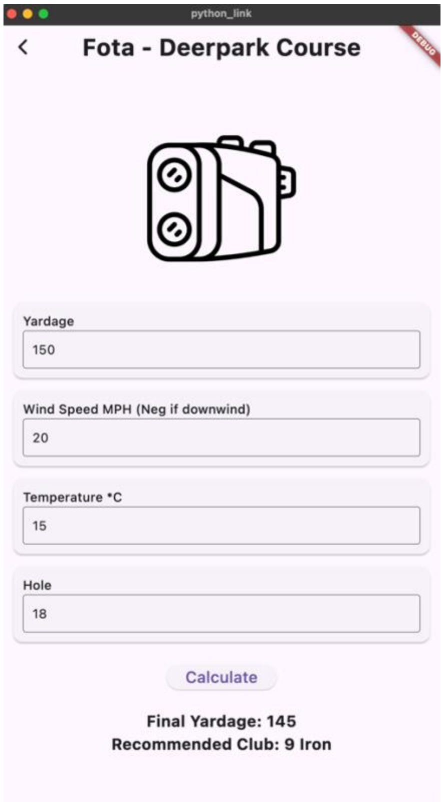
Wind Speed MPH (Neg if downwind): 20

Temperature *C: 15

Hole: 18

Final Yardage: 145
Recommended Club: 9 Iron





Yardage: 150

Wind Speed MPH (Neg if downwind): 20

Temperature *C: 15

Hole: 18

Final Yardage: 145
Recommended Club: 9 Iron

References:

https://www.sciencedirect.com/science/article/pii/S1877705812017213
https://mygolfspy.com/labs/hot-versus-cold-golf-balls/
https://seedgolf.com/blogs/news/how-does-cold-weather-effect-golf-ball-performance
https://www.andrewricegolf.com/andrew-rice-golf/2023/4/what-effect-does-temperature-have
https://www.golfmonthly.com/tips/golf-swing/how-to-calculate-distance-in-the-wind-108215
https://www.andrewricegolf.com/andrew-rice-golf/tag/wind+formula
https://www.paulhurrion.com/tuition/blowin-in-the-wind/
https://www.scottsdalegolf.co.uk/blog/60-second-technique-tips/into-the-wind-vs-down-wind-the-effect-on-your-ball
https://www.golfwrx.com/318416/how-the-wind-affects-your-golf-ball/
https://www.sciencedirect.com/science/article/pii/S187770581100991X
https://journals.sagepub.com/doi/abs/10.1177/1754337118812618
https://www.f22labs.com/blogs/how-to-implement-dark-mode-in-your-flutter-app/