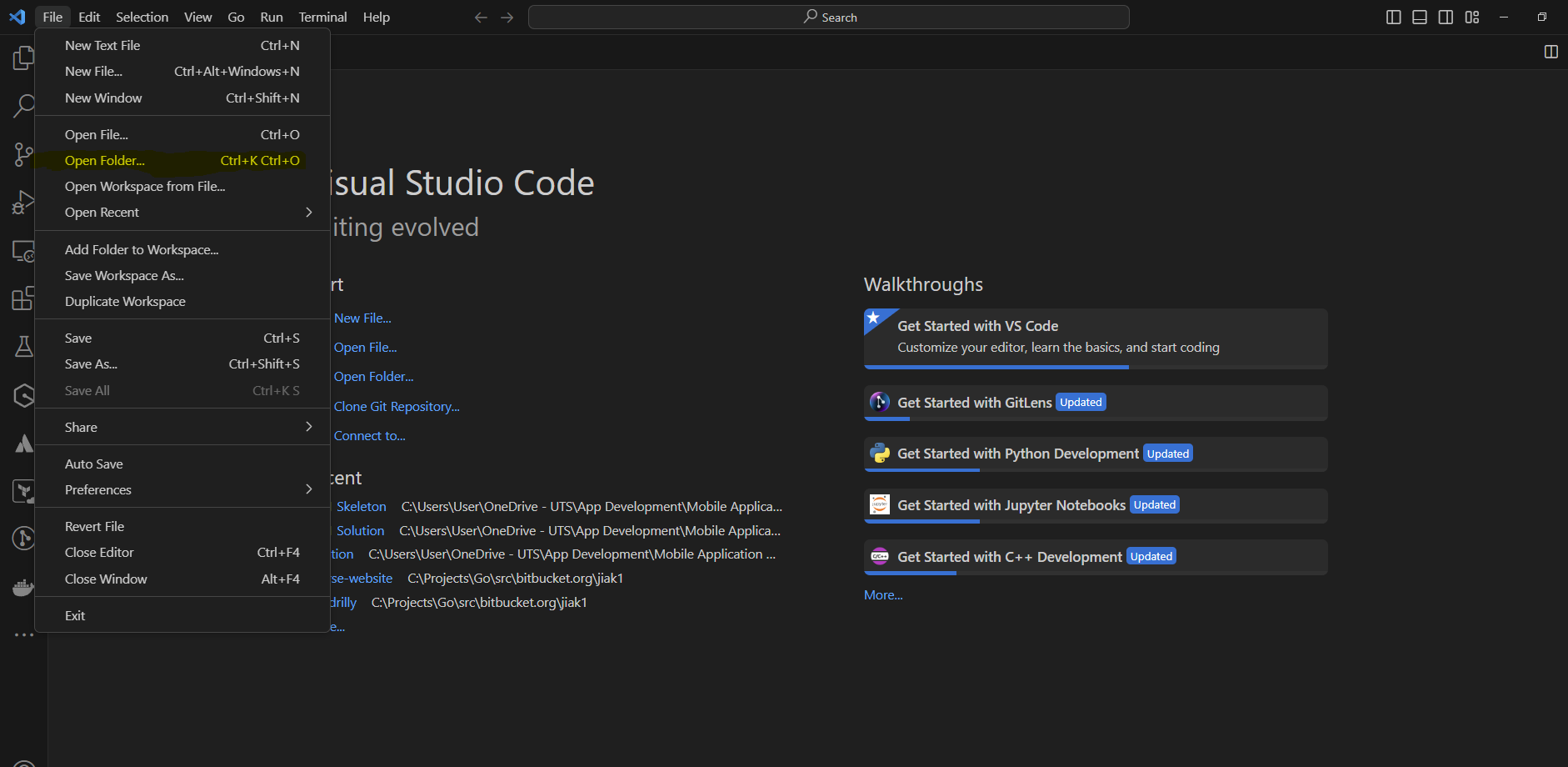
Week 5 Lab Instructions

Opening The Lab Skeleton Code

Download the skeleton code from Openlearning for Week 3 & extract it into a folder that you remember the location of, e.g. Documents.

Open VSCode, press the file “File” button (Top Left), then “Open Folder”. Select the skeleton folder you downloaded.



You should then see the files in the skeleton folder on the left hand side. If you do not see this, you most likely selected the wrong folder. Repeat step 5 again.

A screenshot of a computer

Description automatically generated

Open a terminal in VSCode by clicking the “Terminal” button, then “New Terminal”

A screenshot of a computer

Description automatically generated

If the terminal opened does not say bash, update the terminal default. To do this, click the arrow next to the plus button. Then press “Select default profile” & choose git bash. Press the plus button to then open a new bash terminal.

A screenshot of a computer

Description automatically generated

To install any dependencies, execute npm install in the terminal.

A screenshot of a computer

Description automatically generated

To run your code, type npm start in the terminal.

A screenshot of a computer

Description automatically generated

You have a couple of options on how to preview the app. You can:

* 1. Use a web browser by typing w in the terminal
  2. Use an android device by downloading the ‘Expo’ app from the Google Play Store, opening it & scanning the QR code in the terminal
  3. Use an IOS device by scanning the QR code in the terminal

Lab Exercise

1. Open the weather.ts file inside the src/app/types folder & add the properties for city, temperature, description, humidity & windspeed.

export interface CityWeather {

  /\* Add the relevant fields for city weather, specifically:

     - city which is a string

     - temperature which is a number

     - description which is a string

     - humidity which is a number

     - windSpeed which is a number

  \*/

  city: string

  temperature: number

  description: string

  humidity: number

  windSpeed: number

}

1. Open the App.tsx file and add the home screen & weather details screen components inside the stack navigation so that they are initialised correctly.

        <Stack.Navigator initialRouteName="Home">

          {/\* Add the home screen component \*/}

          <Stack.Screen name="Home" component={HomeScreen} />

          {/\* Add the weather details screen component \*/}

          <Stack.Screen

            name="WeatherDetails"

            component={WeatherDetailsScreen}

          />

        </Stack.Navigator>

1. Create a free account at <https://openweathermap.org/home/sign_up> in order to get access to the API.
2. Once created, visit <https://home.openweathermap.org/api_keys> and create a new api key. Make sure to mark it as active.
3. Open the HomeScreen.tsx file inside the ui folder & update the API\_KEY variables value to be the one you copied from your api key page:

// Get your api key from openweathermap.org & enter it here:

const API\_KEY = 'new value from the api key page'

1. Add logic to clear the city name input after a successful search:

    // Clear the input after a successful search

setCity('')

1. Add an alert notification if a city is not found when a user searches.

if (error.response && error.response.status === 404) {

// Send an alert if the city is not found

Alert.alert('City not found', 'Please enter a valid city name.')

} else {

1. Open the index.ts file inside the store folder & add logic to update the reducer when a new city is added.

const weatherSlice = createSlice({

  name: 'weather',

  initialState,

  reducers: {

    addCityWeather(state, action: PayloadAction<CityWeather>) {

      // Add the new city to the state

state.cities.push(action.payload)

    },

  },

})

1. Update the WeatherDetailsScreen.tsx file to display the current weather information for the selected city:

    {/\* Add text to display the city name \*/}

      <Text style={styles.title}>{cityWeather.city}</Text>

      {/\* Add text to display the city temperature \*/}

      <Text style={styles.detail}>

        Temperature: {cityWeather.temperature} °C

      </Text>

      {/\* Add text to display the city description \*/}

      <Text style={styles.detail}>Description: {cityWeather.description}</Text>

      {/\* Add text to display the city humidity \*/}

      <Text style={styles.detail}>Humidity: {cityWeather.humidity} %</Text>

      {/\* Add text to display the city windSpeed \*/}

      <Text style={styles.detail}>Wind Speed: {cityWeather.windSpeed} m/s</Text>

1. You should now have an application that allows you to search for a city, view it in the results & click on it to see its weather information!