

Instructions:

Evaluate the homework against the outlined criteria in the below rubric, assigning a rating to each criterion. Add points earned across all criteria and convert the total points to a letter grade, assigning a “+” or “-” letter grade designation at your discretion.

A (+/-)	90+	C (+/-)	40-64	F (+/-)	<15
B (+/-)	65-89	D (+/-)	15-39		

Notes:

The deployed assignment utilizes the **SQLAlchemy** library to retrieve data from a database which is used to generate charts and an API. The source code should also be deployed to **Github** or **Gitlab**.

Rubric for Surfs Up:

	Mastery 20 points	Approaching Mastery 15 points	Progressing 10 points	Emerging 5-0 points	Incomplete
Precipitation Analysis	<p>The submission does all of the following:</p> <ul style="list-style-type: none"> ✓ Gets the correct results for the last year of data (note that the last day in the dataset is 8/23/2017) ✓ Creates a pandas dataframe using the date and precipitation columns ✓ Sorts the dataframe by date ✓ Makes a plot using pandas with date as the x and precipitation as the y variables 	<p>The submission does 3 of the following:</p> <ul style="list-style-type: none"> ✓ Gets the correct results for the last year of data (note that the last day in the dataset is 8/23/2017) ✓ Creates a pandas dataframe using the date and precipitation columns ✓ Sorts the dataframe by date ✓ Makes a plot using pandas with date as the x and precipitation as the y variables 	<p>The submission does 2 of the following:</p> <ul style="list-style-type: none"> ✓ Gets the correct results for the last year of data (note that the last day in the dataset is 8/23/2017) ✓ Creates a pandas dataframe using the date and precipitation columns ✓ Sorts the dataframe by date ✓ Makes a plot using pandas with date as the x and precipitation as the y variables 	<p>The submission does 0-1 of the following:</p> <ul style="list-style-type: none"> ✓ Gets the correct results for the last year of data (note that the last day in the dataset is 8/23/2017) ✓ Creates a pandas dataframe using the date and precipitation columns ✓ Sorts the dataframe by date ✓ Makes a plot using pandas with date as the x and precipitation as the y variables 	<p>No submission was received</p> <p>-OR-</p> <p>Submission was empty or blank</p> <p>-OR-</p> <p>Submission contains evidence of academic dishonesty</p>
Station Analysis	<p>The submission does all of the following:</p> <ul style="list-style-type: none"> ✓ Correctly outputs the number of stations in the dataset (9) ✓ Correctly finds the most active station by using <code>count</code> (USC00519281) ✓ Gets the min, max, and average temperatures for the most active station (USC00519281) 	<p>The submission does 3 of the following:</p> <ul style="list-style-type: none"> ✓ Correctly outputs the number of stations in the dataset (9) ✓ Correctly finds the most active station by using <code>count</code> (USC00519281) ✓ Gets the min, max, and average temperatures for the most active station (USC00519281) ✓ Correctly plots a histogram for 	<p>The submission does 2 of the following:</p> <ul style="list-style-type: none"> ✓ Correctly outputs the number of stations in the dataset (9) ✓ Correctly finds the most active station by using <code>count</code> (USC00519281) ✓ Gets the min, max, and average temperatures for the most active station (USC00519281) ✓ Correctly plots a histogram for 	<p>The submission does 0-1 of the following:</p> <ul style="list-style-type: none"> ✓ Correctly outputs the number of stations in the dataset (9) ✓ Correctly finds the most active station by using <code>count</code> (USC00519281) ✓ Gets the min, max, and average temperatures for the most active station (USC00519281) ✓ Correctly plots a histogram for the 	

	<ul style="list-style-type: none"> ✓ Correctly plots a histogram for the last year of data using <code>tobs</code> as the column to count. 	the last year of data using <code>tobs</code> as the column to count.	the last year of data using <code>tobs</code> as the column to count.	last year of data using <code>tobs</code> as the column to count.	
API SQLite Connection & Landing Page	<p>The Flask Application does all of the following:</p> <ul style="list-style-type: none"> ✓ Correctly generates the engine to the correct sqlite file ✓ Uses <code>automap_base()</code> and reflects the database schema ✓ Correctly saves references to the tables in the sqlite file (measurement and station) ✓ Correctly creates and binds the session between the python app and database 	<p>The Flask Application does 3 of the following:</p> <ul style="list-style-type: none"> ✓ Correctly generates the engine to the correct sqlite file ✓ Uses <code>automap_base()</code> and reflects the database schema ✓ Correctly saves references to the tables in the sqlite file (measurement and station) ✓ Correctly creates and binds the session between the python app and database 	<p>The Flask Application does 2 of the following:</p> <ul style="list-style-type: none"> ✓ Correctly generates the engine to the correct sqlite file ✓ Uses <code>automap_base()</code> and reflects the database schema ✓ Correctly saves references to the tables in the sqlite file (measurement and station) ✓ Correctly creates and binds the session between the python app and database 	<p>The Flask Application does 0-1 of the following:</p> <ul style="list-style-type: none"> ✓ Correctly generates the engine to the correct sqlite file ✓ Uses <code>automap_base()</code> and reflects the database schema ✓ Correctly saves references to the tables in the sqlite file (measurement and station) ✓ Correctly creates and binds the session between the python app and database <p>-OR-</p> <ul style="list-style-type: none"> ✓ Flask app does not start 	
API Static Routes	<p>The static routes do all of the following:</p> <p>Precipitation route</p> <ul style="list-style-type: none"> ✓ Returns the jsonified precipitation data for the last year in the database ✓ Returns json with the date as the key and the value as the precipitation <p>Stations route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data of all of the stations in the database <p>Tobs route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data for the most active station (USC00519281) for the last year of data 	<p>The static routes do 3 of the following:</p> <p>Precipitation route</p> <ul style="list-style-type: none"> ✓ Returns the jsonified precipitation data for the last year in the database ✓ Returns json with the date as the key and the value as the precipitation <p>Stations route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data of all of the stations in the database <p>Tobs route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data for the most active station (USC00519281) for the last year of data 	<p>The static routes do 2 of the following:</p> <p>Precipitation route</p> <ul style="list-style-type: none"> ✓ Returns the jsonified precipitation data for the last year in the database ✓ Returns json with the date as the key and the value as the precipitation <p>Stations route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data of all of the stations in the database <p>Tobs route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data for the most active station (USC00519281) for the last year of data 	<p>The static routes do 0-1 of the following:</p> <p>Precipitation route</p> <ul style="list-style-type: none"> ✓ Returns the jsonified precipitation data for the last year in the database ✓ Returns json with the date as the key and the value as the precipitation <p>Stations route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data of all of the stations in the database <p>Tobs route</p> <ul style="list-style-type: none"> ✓ Returns jsonified data for the most active station (USC00519281) for the last year of data <p>-OR-</p> <ul style="list-style-type: none"> ✓ Flask app does not start 	
API Dynamic Route	<p>The dynamic route does all of the following:</p> <p>Start route</p> <ul style="list-style-type: none"> ✓ Route accepts the start date as a parameter from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the end of the dataset 	<p>The dynamic route does 3 of the following:</p> <p>Start route</p> <ul style="list-style-type: none"> ✓ Route accepts the start date as a parameter from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the end of the dataset 	<p>The dynamic route does 2 of the following:</p> <p>Start route</p> <ul style="list-style-type: none"> ✓ Route accepts the start date as a parameter from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the end of the dataset 	<p>The dynamic route does 0-1 of the following:</p> <p>Start route</p> <ul style="list-style-type: none"> ✓ Route accepts the start date as a parameter from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the end of the dataset 	

	Start/end route ✓ Route accepts the start and end dates as parameters from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the given end date	Start/end route ✓ Route accepts the start and end dates as parameters from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the given end date	Start/end route ✓ Route accepts the start and end dates as parameters from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the given end date	Start/end route ✓ Route accepts the start and end dates as parameters from the URL ✓ Returns the min, max, and average temperatures calculated from the given start date to the given end date -OR- ✓ Flask app does not start	
--	---	---	---	---	--

Rubric for Surfs Up - Bonus Analyses:

	Mastery 20 points	Progressing 10 points	Emerging 0 points	Incomplete
Optional Analyses	<p>The submission does all of the following:</p> <p>Trip Temperature Analysis ✓ Uses the calc_temps function to get the min, max, and average temperatures for a date range of their choosing ✓ Uses the calculated temperatures to generate a bar chart with an error bar.</p> <p>Daily Rainfall Average ✓ Calculates the min, max, and average temperatures for each day of their trip and appends them to a list. ✓ Creates a dataframe from the list and generates a stacked line chart plotting the min, max, and average temps for each day of their trip</p>	<p>The submission successfully does only 1 of the optional analyses:</p> <p>Trip Temperature Analysis ✓ Uses the calc_temps function to get the min, max, and average temperatures for a date range of their choosing ✓ Uses the calculated temperatures to generate a bar chart with an error bar.</p> <p>-OR-</p> <p>Daily Rainfall Average ✓ Calculates the min, max, and average temperatures for each day of their trip and appends them to a list. ✓ Creates a dataframe from the list and generates a stacked line chart plotting the min, max, and average temps for each day of their trip</p>	<p>The submission attempts one or both of the following, but fails:</p> <p>Trip Temperature Analysis ✓ Uses the calc_temps function to get the min, max, and average temperatures for a date range of their choosing ✓ Uses the calculated temperatures to generate a bar chart with an error bar.</p> <p>Daily Rainfall Average ✓ Calculates the min, max, and average temperatures for each day of their trip and appends them to a list. ✓ Creates a dataframe from the list and generates a stacked line chart plotting the min, max, and average temps for each day of their trip</p>	<p>No submission was received</p> <p>-OR-</p> <p>Submission was empty or blank</p> <p>-OR-</p> <p>Submission contains evidence of academic dishonesty</p>