	TO DO	IN PROGRESS	COMPLETE
Satoshi			
			Task 1: Remove google authentication from the login page.
			Task 1.1: For removing google auth just make sure to only have the login be email/username and password based
		Task 2: Add logout functionality to the website.	
		Task 2.1: Add a logout confirmation popup	
		Task 2.2: Add a logout button/link on the nav bar on all pages (except for sign up and log in)	
		Task 3: Add redirects based on log in and log out	
		Task 3.1: When website loads, if no user logged in make sure every page redirects to login	
		Task 3.2: Adter a user signs up or logs in, they should be redirected to the home page	
		Task 3.3: If there are sign up or log in issues, make sure the program does redirect but handles the validation errors	
		Task 4: For the UI on the Sign Up page, make sure that the submit button for signing up reads "Sign Up" instead of "Sign Up"	
		Task 5: Signing up should sign in the user and redirect.	
	Task 6: Ensure that the user can not access any webpages except for sign up and log in prior to signing up or logging in		
	Task 6.1: Remove all guest functionality for the website		
	Task 6.2: Add a valid account check to the login code so that if invalid account info is entered, the website won't think a user has logged in		
	Task 7: Have a way to reflect the user logged in across all pages		
	Task 7.1: . On the search history page, list out the users name, to show it is their search history		
	Task 8: Implement unit and acceptance tests for all features of the Log in, Log out , and sign up pages.		
			Task 9: Remove the nav bar from signup/login pages
	Task 10: Implement and test input validation for the Activity and Vacation pages.		
Alex			
		Task 1: Add acceptance tests for the database to show which information is being saved	
		Task 1.1 (Example) Acceptance test for search for something, show that it shows up in one user's search history and not another.	
		Task 2: Implement the database to maintain the user information over all webpages in the program	
		Task 2.1: Track the user logged in, their search history, likes, etc.	
	Task 3: Implement a liking system for the cities		
	Task 3.1: Do based on favorites and add it to the database, with a list of favorited cities.		
		Task 4: Add tests to ensure that the correct number of likes is added to the favorited cities across multiple user accounts and searches.	
John			
		Task 1: Implement the Prior Search Page	
		Task 1.1: Implement the functionality to get the prior search from the database	
	Task 1.2: Ensure that the right city info is properly loading into the field on the search page.		
	Task 1.3: Write acceptance tests for the search history page, ensuring that it loads the correct information about the cities.		
		Task 2: Add the functionality of defining the radius for the search field	
			Task 2.1: Make sure that the correct distance is being gathered from the API
	Task 2.2: Make sure that no cities outside that radius are being added.		
	Task 2.3: Add tests for the radius of the search field to the acceptance tests.		
Lauren			
			Task 1: Change the nav bar color to be darker so that it is more visible and distinct on the page.
	Task 2: Implement the updated ui on all webpages, so that the appearance of the website is consistent and cohesive.		
			Task 3: Make sure the show me the weather button on the main search page is on one line
			Task 4: Change the "Change Units" button to a toggle between celsius and fahrenheit
		Task 5: Make sure that when the results load on the pages, there is not a scroll bar on the side of the page (aka all content should be in "one" page)	
		Task 6: Add acceptance tests for all aspects of the UI	
		Task 7: Make sure to update pages to reflect the features needed from project 1 (graph and images on analysis).	
Jung			
			Task1: Update the numbers in the tables to start at 1 and continually increase without resetting to one on the next page.
		Task 2: Add a next and previous button to the pagination on the vacation and activity pages.	
		Task 3: If there are more than 5 pages of results, make sure to only show 5 pages at a time.	
		Task 4: Add more acceptance tests for the pagination for next and previous and show that it is updating which pages are on the screen (when there are more than 5)	
	Task 5: Once the favoriting system is implemented, add a column for the number of favorites to the table.		
	Task 5.1: This column should also have the functionality to sort the results (like distance)		
	Task 6: When sorting results, make sure that all pages are updated, so reload the page as a whole with the sorted results in the paginated pages.		