Project Name: Cook AI	**Sprint Number:** *1*
Test Case	
Sprint Deliverable Name: *Website hosting*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)*: **Is the website accessible via URL?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.5	**Test Environment: Web browser**
Description: Test to make sure the URL to the Cook AI website is functional and takes the user to the Cook AI main page.	
Pre-condition*(What do we assume)***:** Website html is functional and hosting is set up via Firebase	
Dependencies: *(What do we need)***:** website html, Firebase, web browser, website URL	

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	Enter URL into web browser (https://cook-ai-4fc88.web.app/)	website URL	URL opens Cook AI main page			
3.						
4.						
5.						
6.	(add more steps as needed)					

Project Name: Cook AI	**Sprint Number:** *best guess*
Test Case	

Sprint Deliverable Name: Recipe Feedback	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)*:	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.5	**Test Environment: Windows PC**
Description:	
Pre-condition*(What do we assume)***:** User signed in	
Dependencies: *(What do we need)***:** git, python, web browser	

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	User receives a recipe from the kitchen page					
3.	user can give feedback based on recipe		user is able to submit feedback successfully			
4.						
5.						
6.	(add more steps as needed)					

Project Name: Cook AI	**Sprint Number:** *sprint 1*

Test Case

Sprint Deliverable Name: Sprint 1	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)*: **Do Recommended Recipes work?**	**Test executed by:** *leave blank for test plan*

SRS Corresponding *(hotlink to SRS section)* Section 3.X.X	**Test Environment:windows 11, chrome**
Description:	
Pre-condition*(What do we assume)***:** user is logged in, fridge is fully stocked	
Dependencies: *(What do we need)***:** windows, chrome	

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	User clicks generate under the “Recommended” Tab	none	10 recipes are generated using ONLY ingredients from the fridge			
3.						
4.						
5.						
6.	(add more steps as needed)					

Project Name: Cook AI	**Sprint Number:** *sprint 1*
Test Case	
Sprint Deliverable Name: *Allow items to be added to fridge from an uploaded picture*	**Test Date:** *sprint 1*
Test Title *(What are we testing)***: Does fridge work?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.3	**Test Environment: windows, chrome browser**
Description:	

Pre-condition*(What do we assume)***:** user logged in
Dependencies: *(What do we need)***:** windows, chrome, image of a grocery haul

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	Upload image to site	any picture	fridge is populated with the items from the image			
3.						
4.						
5.						
6.	(add more steps as needed)					

Project Name: Cook AI	**Sprint Number:** 1
Test Case	
Sprint Deliverable Name: *Create Database*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)***: Can a Firebase Database be created and used for storage of user information successfully?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.3	**Test Environment: Windows PC**
Description: Create a Firebase Database that will store user profile information, fridge contents, and authentication information.	
Pre-condition*(What do we assume)***:** Team member creating database has a Google account	
Dependencies: *(What do we need)***:** Python, Git, Google Firebase	

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	Create a Firebase project for CookAI.	None	Project with empty database is created on Firebase			
3.	A document “users” is initiated in the Firebase database.	None	A document “users” that will hold information for all users is created.			
4.	When the first user signs up for CookAI, their credentials will appear in the “users” collection.	User’s generated ID from Google/Firebase	Empty fields for user details such as “fridge”, “allergies”, “diet”, and “spice tolerance” will appear under the ID for that user.			
5.	As the user updates their details, they will be updated in the database.	None	User details will change as user edits them on the CookAI app/website			

Project Name: Cook AI	**Sprint Number:** *3*
Test Case	
Sprint Deliverable Name: *Profile Page - spice tolerance*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)***: Is the website accessible via URL?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.5	**Test Environment: Windows PC**
Description: Verify that the recipe generator excludes recipes containing allergens selected by the user.	

Pre-condition*(What do we assume)***:** user is logged in, fridge filled with food items
Dependencies: *(What do we need)***:** user selects their spice tolerance on their profile

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	user selects their estimated spice tolerance on the profile page	spice tolerance level (variable)	selected spice tolerance is passed to the recipe generator (back end)			
3.	user proceeds to the generate recipe page (front end)		user is transferred to the generate recipe page			
4.	user clicks “generate recipe” button	list of allergens	AI generates a recipe with the adequate spice levels			
5.						
6.						

Project Name: Cook AI	**Sprint Number:** *3*
Test Case	
Sprint Deliverable Name: *Profile Page - diet*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)***: Is the website accessible via URL?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.5	**Test Environment: Windows PC**
Description: Verify that the recipe generator excludes recipes containing allergens selected by the user.	
Pre-condition*(What do we assume)***:** user is logged in, fridge filled with food items	

Dependencies: *(What do we need)***:** user selects their diet on their profile

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	user selects their preferred diet on the profile page	diet (variable)	selected diet is passed to the recipe generator (back end)			
3.	user proceeds to the generate recipe page (front end)		user is transferred to the generate recipe page			
4.	user clicks “generate recipe” button		AI generates a recipe according to the selected diet plan			
5.						
6.						

Project Name: Cook AI	**Sprint Number:** *3*
Test Case	
Sprint Deliverable Name: *Profile Page - allergies*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)***: Is the website accessible via URL?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.5	**Test Environment: Windows PC**
Description: Verify that the recipe generator excludes recipes containing allergens selected by the user.	
Pre-condition*(What do we assume)***:** user is logged in, fridge filled with food items	
Dependencies: *(What do we need)***:** user selects their food allergies on their profile	

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	user selects their allergies on the profile page	none	list of selected allergies is generated			
3.	user proceeds to the generate recipe page (front end)		user is transferred to the generate recipe page			
4.	list of allergies is passed to the recipe generator (back end)	list of allergens	recipe generator receives the list of allergens			
5.	user clicks “generate recipe” button		AI generates a recipe without including the selected allergies			
6.						

Project Name: Cook AI	**Sprint Number:** 1

Test Case

Sprint Deliverable Name: *Create Register Page*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)*: **Can a user create a CookAI account successfully?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.2	**Test Environment: Windows PC**

Description: Allow the user to create a new CookAI account using Google sign in.

Pre-condition*(What do we assume)***:** User doesn’t have a CookAI account but has a Google account

Dependencies: *(What do we need)***:** Python, Git, Google

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	User clicks on the “login/register” button on the top right corner of the homepage	None	The user is taken to the login/register page			
3.	User clicks on the “sign up with Google” button	None	The user is taken to a Google login page			
4.	User inputs their Gmail and password to the Google sign in page	User’s Gmail/password	The user successfully signs up, assuming they provided their correct Gmail and password. If not, user will not be able to sign in until correct credentials are provided			
5.	User is logged into their new CookAI account	None	The user can start generating recipes, editing their fridge, etc.			

Project Name: Cook AI	**Sprint Number:** 1
Test Case	
Sprint Deliverable Name: *Create Login Page*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)*: **Can a user with an existing CookAI account log in successfully?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.2	**Test Environment: Windows PC**
Description: Allow the user to sign into their existing CookAI account using Google sign in.	
Pre-condition*(What do we assume)***:** User already has a CookAI/Google account	
Dependencies: *(What do we need)***:** Python, Git, Google	

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	User clicks on the “login/register” button on the top right corner of the homepage	None	The user is taken to the login/register page			
3.	User clicks on the “log in with Google” button	None	The user is taken to a Google log in page			
4.	User inputs their Gmail and password to the Google sign in page	User’s Gmail/password	The user successfully signs in, assuming they provided their correct Gmail and password. If not, user will not be able to sign in until correct credentials are provided			
5.	User is logged into their CookAI account	None	The user can start generating recipes, editing their fridge, etc.			

Project Name: Cook AI	**Sprint Number:** *1*
Test Case	
Sprint Deliverable Name: *Global Sign-In/Out Feature*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)*:	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.5	**Test Environment: Windows PC**
Description:	

Pre-condition*(What do we assume)***:** User Signed in
Dependencies: *(What do we need)***:** git, python

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	Navigate through home, kitchen, fridges pages		User stays signed while navigating the website			
3.	User signs out and tries to access all pages		Access denied			
4.						
5.						
6.	(add more steps as needed)					

Project Name: Cook AI	**Sprint Number:** *1-4*
Test Case	
Sprint Deliverable Name: *Expand Advanced Syntax Checker*	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)***: Does the check catch errors and fix them automatically?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.5	**Test Environment: Windows PC**
Description:	

Pre-condition*(What do we assume)***:** The server is running, the user is in the kitchen page, fridge is full of ingredients
Dependencies: *(What do we need)***:** python, git

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	Type in any recipe and the API key	None	Recipe and key is typed			
3.	Click Generate	None	Generate spins			
4.	The log tells us there was more than 1 attempt at generating the recipe due to syntax	None	The log says there was more than 1 attempt, but the recipe still generates. It was 'caught'.			
5.						
6.	(add more steps as needed)					

Project Name: Cook AI	**Sprint Number:** *1*
Test Case	
Sprint Deliverable Name: Enable adding/deleting ingredients from a local fridge database	**Test Date:** *sprint 1*
Test Title *(What are we testing)***: Enable adding/deleting ingredients from a local fridge database**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section)* Section 3.3	**Test Environment: Windows/chrome browser**
Description:	

Pre-condition*(What do we assume)***:** User is signed in
Dependencies: *(What do we need)***:** python, git

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	Test editing/adding/deleting/clearing ingredients list	none	fridge list is dynamically updated			
3.						
4.						
5.						
6.	(add more steps as needed)					

Project Name: Cook AI	**Sprint Number:** *1*
Test Case	
Sprint Deliverable Name: Update start_local-windows_setup_and_server.sh	**Test Date:** *leave blank for test plan*
Test Title *(What are we testing)***: Does the .sh file set the environment up?**	**Test executed by:** *leave blank for test plan*
SRS Corresponding *(hotlink to SRS section) NONE*	**Test Environment: Windows PC**
Description: to make it easy for the user, we developed a script called start_local-windows_setup_and_server.sh that will set up the environment for the user and start the server automatically with a simple double click	
Pre-condition*(What do we assume)***:** User has downloaded the project	
Dependencies: *(What do we need)***:** Python, Git	

Step	Test steps	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)	Notes
1.	*(specific step)*	*(files or data needed to execute step)*	*(what the result should be when step is executed)*	*leave blank for plan*	*leave blank for plan*	
2.	Double click the start_local-windows_setup_and_server.sh file	start_local-windows_setup_and_server.sh	The .sh file opens, runs, and sets up the environment fully for the user, and the server boots up with the address 127.0.0.1			
3.						
4.						
5.						
6.	(add more steps as needed)					