

Test Plan Creation Date			12/04/2022					
Created By			James Wright, George Reales					
Project Name			V0.1.7 - Food Blog					
Sr.No.	Module	Sub-module	Pre-Requlite	Steps to be followed	Expected Result	Actual Result	Comments	Status (Pass / Fail)
	Testing auth.routes	N/A	Nodemailer and bcrypt library. Auth routes must be written. Nodemailer mock file set up	1. Run tests on 3 functions, mock any use of libraries. 2. Assert each function returns the expected value. 3. Check function produces no errors	authFN returns true when cookie created. False otherwise  encryptpass returns no errors	encryptpass verified to be working by comparing the hashes.  Authfn only returns true when the auth value is true in the cookie.	nodemailer is mocked.	Pass
1				1. Check function produces no errors	authenticatedEmail successfully sends an email	authenticatedEmail produced no errors and the console.loged successfully.		
			mockingoose library. Function getPosts must be written	1. Create test data 2.Mock the db library 3. Run the function getPosts to check for errors. 4. Assert the db library returns the test data	No errors found. The results from the model must match the test data object	No results can be found from the DB library	Incorrect model being mocked	Fail
2	Testing post.routes	N/A						
			mockingoose library. Function getPosts must be written	1. Create test data 2.Mock the db library 3. Run the function getPosts to check for errors. 4. Assert the db library returns the test data	No errors found. The results from the model must match the test data object	Results yielded match the test data. Correct result logged	Had to mock the model from posts instead of using the index	Pass
3	Testing post.routes	N/A						
			mockingoose library. Function checkUserExists must be written	1. Create test data 2.Mock the db library 3. Run the function checkUserExists to check for errors. 4. Assert the db library returns the test data 5.Assert the function returns true	No errors found. The function must return true and produce no errors	Results yielded match the test data. Correct result logged. Function returned true		Pass
4	Testing user.routes	N/A						
5	Login System	User Testing on the login	Server running, and user details are known	1.Enter the user details 2. Complete captcha 3. Get authentication code 4. Check for login	Quick and complete login. Should be obvious to the user they are logged in by the navbar offering a profile page	Login took 45 seconds to complete with no autofill. Page for register/login changed to Profile. 2FA code is too long and captcha was too hard.	2FA code shortened to increase usability	Fail
6	Login System	User Testing on the login	Server running, and user details are known	1.Enter the user details 2. Complete captcha 3. Get authentication code 4. Check for login	Quick and complete login. Should be obvious to the user they are logged in by the navbar offering a profile page	Login took 30 seconds to complete with no autofill. Page for register/login changed to Profile.	2FA code shortened to increase usability	Pass
7	Registration	User testing on the Registration	Server running	1. Navigate to register page 2. Enter details for a new user 3. Complete captcha 4. Click register 5. Complete Authentication via e-mail	The user should be logged in and the account should be created	Registration took ~1 minute to complete. Authentication page loaded correctly and the website logged me in after registration. The account was also created.	Everything worked as expected	Pass
8	Viewing posts	User testing on viewing posts	Server running	1. Access the home page 2. Navigate to the Create a post screen from the navbar. 3. Upload an image for the post. 4. Click the create a post button.	The user should see all the posts that have been created fill up the screen.	All the posts stored on the website display for the user.	N/A	Pass
9	Creating a post	User testing on creating a post	Server running and user is logged in.	1. Navigate to the home page 2. Find a post that you have made on the login you are logged in on. 3. If you have not made a post refer to test 9. 4. Click the create a post button.	The user should be able to easily upload information for their desired post and the post should be quickly made and made visible to the user on the home page immediately after creating the post.	The post gets created and the user is sent to the home page.	N/A	Pass
10	Deleting a post	User testing on deleting a post	Server running and user is logged in.	1. Navigate to the home page 2. Find a post that you have made on the login you are logged in on. 3. If you have not made a post refer to test 9. 4. Click the delete post button.	The user's post should get deleted and should no longer be visible.	The post gets deleted and the user can no longer see the post.	N/A	Pass
11	Editing a post	User testing on edfline a post	Server running and user is logged in.	1. Navigate to the home page 2. Click on the search bar 3. Type what you would like to find 4. Press enter and find it	The user's post should be edited with their changes.	The post is edited	N/A	Pass
12	Using the search bar	User testing on using the search bar	Server running		The user should be able to search via the search bar	They can utilise the search bar	N/A	Pass
13	Captcha system	User testing on the captcha system	Server running	1. Navigate to log in page 2. Attempt to login without captcha 3. See if you can successfully log in	The user should not be able to log in	The user cannot log in	N/A	Pass