> N/A N/A N/A

> N/A N/A N/A N/A N/A N/A N/A

nequii	cilicitis	

Title/Notes:			
Commit at which tests were ran:  Note: Some features that passed in one testrun may be n/a on a following test as team member was testing on a different branch.			
Note: some reatures that passed in one testrun may be n/a on a following test as team member was testing on a different branch.  When testing on development all previously passed tests should pass again, if falled, implement fix and rerun tests.			
Functional Requirements  1- User			
1.1 Verify that a local user view weather on app opened when connected to Wi-Fi or cellular network with data enabled.	S. A	P	JS.
1.2 Users must be able to view a five day forecast	S. A	P	JS
1.3 Users must be able search a location by name	S. A	P	JS
1.4 User must be able to create an account if doesn't have one already	O, A	P	AN/SS
1.5 User must login to access the following features	S, A	P	AN
1.6 User must be able to login to access their dashboard information	S, A	P	AN/SS
1.7 User must be able to login to edit their dashboard information	S, A	P	AN/SS
1.8 User must be able to save up to five locations	S, A	P	TN/JS
1.9 User must be able to share their dashboard information	S, A	P P	AN/TN
1.10 User must receive error message if location is not sent	S, A	P	JS
1.11 User must be able to customize dashboard from given options 1.12 User must be able to log out and lose access to previous features	S, A S. A	P	JS SS/AN
1.12 Oser must be able to togout and use access to previous readules	3, A		SS/AIN
2 - Weather Data Retrieval			
2.1 The system must connect to the external WeatherApp API and retrieve data for the current weather.	I, S	P	TN
<ol> <li>The system must connect to the external WeatherApp API and retrieve 5-day forecast data.</li> </ol>	I, S	P	TN
2.3 The system must update weather data on the dashboard at regular intervals.	UN	P	
2.4 The system must handle API errors and display appropriate messages to the user.	O, A	P	TN TN
2.5 The system must format and display temperature, weather conditions, and weather icons. 2.6 The system must switch between C/F upon the user prompt	A S	P	IN IS
2.6 The system must switch between C/F upon the user prompt	5	P	ZS.
3 - Dashboard Customization			
3.1 The system must allow users to customize the dashboard by adding or removing locations	US, A	P	TN/JS
3.2 The system must allow users to change the color of the dashboard.	US, A	F	TN/JS
3.3 The system must allow users to set a preferred format for displaying weather data.	US, A	F	TN
3.4 The system must save dashboard preferences upon user action.	S, A	P P	TN
3.5 The system must load saved preferences when the user logs in.	S, A	P	
4 - Firebase Authentification	Firebase was scra	pped and replaced with	Database Authentication
5 - CI/CD Pipeline			
5.1 The system must automatically build and test on every new commit.	s	F	TN
5.2 The system must deploy successfully without errors.	s	F	TN
5.3 The system must send notification for failed builds or tests.	S	F	TN
6 - Automated Testing			
6.1 The system must run unit tests on critical components like API requests and data formatting.	UN, S	F	TN
6.2 The system must run integration tests for end-toend user flows.	UN, S	F	TN
6.3 The system must run User Interface tests to ensure a responsive and accesible interface.	UN, S	F	TN
7 - Database Testing			
7.1 The database connection/disconnection is successful	s,o	P	AN
7.2 User information can be retreived from database	s,o	P	AN
7.3 Users can be added/deleted from database	S,O	P	AN
7.4 User information can be updated in database	S,O	P	AN
7.5 Locations can be added in database	S,O	P	AN
7.6 Locations can be retreived from database	S,O	P	AN
7.7 Dashboard locations can be added to database	s,o	P	AN
7.8 Dashboard locations can be removed from database 7.9 Dashboard locations can be queried per user	s,o s.o	P P	AN AN
7.9 Dashboard locations can be queried per user	5,0	r	AN
	_		
Non Functional Requirements			
1 - Security  1.1 The parton must present unsufficient account of user accounts, data and personal information	_		ee

Non Functional Requirements			
1 - Security			
1.1 The system must prevent unauthorized access to user accounts, data and personal information.	0	F	SS
1.2 The system must use secure connections (HTTPS) for API requests.	O, UN	F	SS
2 - Usability			
2.1 The system must have a user-friendly interface that is intuitive for all users	US		
2.2 The system must allow users to recover from all errors and exceptions without crashing	US		
2.3 The system must provide help messages for key user actions.	US		
2.4 The application must be able to run on various browsers	US		
2.5 The application must be able to run on various devices	US		
3 - Reliability			
3.1 The system must use recover from API errors and inform users.	0, US	F	
3.2 The system must maintain session data for the duration of user interaction.	O, A	F	
4 - Performance			
4.1 The system must load weather data within 5 seconds of request.	O, A, US	F	
4.2 The system must function optimally on moderate internet connection.	O, A, US	F	
5 - Maintainability			
5.1 The system must be modular to support easy updates and bug fixes.	UN, S	F	
5.2 The codebase must follow a consistent style guide to improve readability.	S, I	Code inspection TBD	
5.3 Every section of code must be documented to support easy updates and bug fixes.	S, I	TBD	
6 - Avaitability			
6.1 The system must be available 100% of the tiume during peak usage hours.	US, A, O		

Non Functional Requirements end here

| 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.100/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.1000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.10000/2024 | 1.100000/2024 | 1.100000/2024 | 1.100000/2024 | 1.100000/2024 | 1.1000000/2024 | 1.1000000/2024 | 1.1000000/2024 | 1.100000000000000000000000000000000