

Tic-Tac-Toe Manual Testing

- **Tic-Tac-JS: Test Plan:**

Overview: Test functionality of the app.

Links: <http://localhost:4000> (can be found within VSCode)

Test Criteria:

Does the start button initiate a game of Tic-Tac-Toe?

Does each square work?

Does the 'AI' play back?

Is the real winner declared?

Entry Criteria:

Test criteria provided,

Running version provided for testing,

Computer for testing.

Exit Criteria:

All high priority tests have passed,

The allotted time is up

Other Details:

Should be able to run on other browsers and be fully functional (minimum requirement is Chrome).

- **Test case: Completed**

-Test Case:(Start Button)

Description: This test will verify if the app starts a game upon a click of the start button.

Steps: Navigate to browser → User sees game and a start button → user clicks on start

Post Conditions: Button must be able to register a request and send a response back to the user.
A game should initiate upon the click.

- Test case: Failed

-Test Case: (Square Functionality)

Description: This test will verify that each square works upon initiation of the game.

Steps: Navigate to browser → User sees and clicks on start button → click on as many different squares as possible → refresh browser to test more squares

Post Conditions: Each square must register a click and send a response back to the user, upon each click the user must continue playing until the AI/User wins or a tie game occurs

-Test Case: (AI play)

Description: This test verifies that the 'AI' responds to the user's plays and fills in squares

Steps: Navigate to/Refresh browser → User sees and clicks on start button → User plays → Repeat process

Post Conditions: Each movement by the player should initiate a response from the AI, The AI should play until it wins.

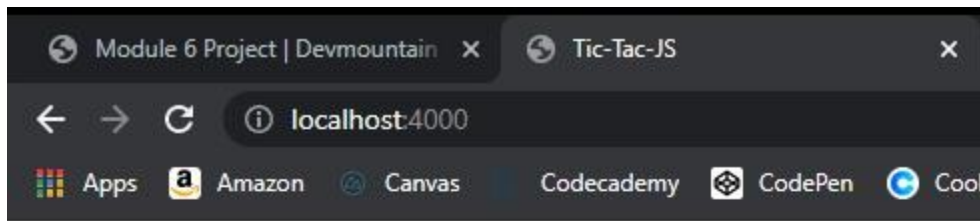
-Test Case: (Winner status)

Description: Upon winning/losing a game, the victor/loser must be announced at the top.

Steps: Navigate to/Refresh browser → User sees and clicks on start button → User plays → Repeat process

Post Conditions: At the end of every game, a winner/loser should be announced if there are no more available squares or if a line of three has been made.

- Bug Report:



X lost

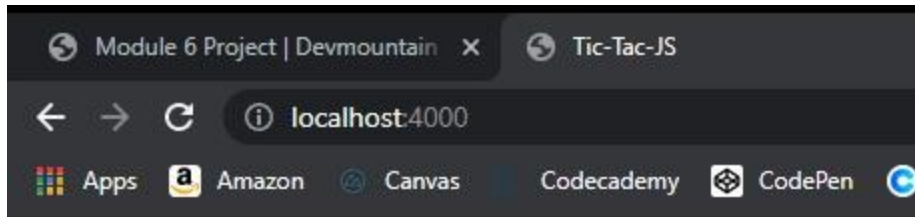
O	O	X
o	X	
X	X	

Winner Bug: Winner status shows as a loss.

Steps to reproduce: Navigate to site, start game, play to win, win, view loss.

Expected Result: To win and see that “X Won”

Actual result: “X Lost”



Tic Tac JS

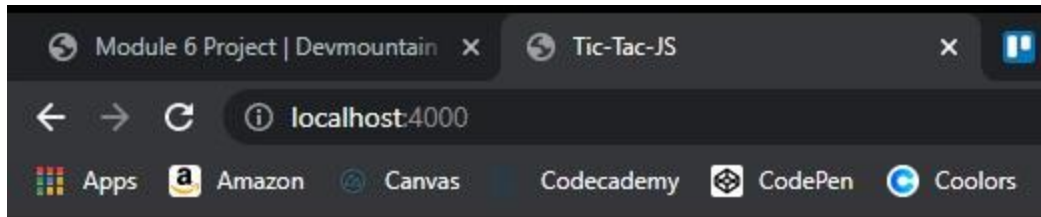
O	X	X
o	?	X
X	X	?

AI Bug: AI stops playing and user plays on their own.

Steps to reproduce: Navigate to site, start game, block the AI movement, allow it to win, fill in any spaces that deny your own victory.

Expected Result: To see the AI fill in empty spaces and win

Actual result: AI gives up after three moves, if one move is blocked, it will give up.



Tic Tac JS

O	X	O
X O	X	

X=O?

Square middle left Bug: User cannot place X on middle left square without an O replacing it.

Steps to reproduce: Navigate to site, start game, block the AI movement on square 2 (top middle), place and X on the center square, place X on left middle.

Expected Result: To be able to place an X on left middle square.

Actual result: AI takes over your square, furthermore, an X can be placed over any O (except left middle).