

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

Assignment 3 - Whe Whe Lottery Ticket Machine

Submission Checklist

This is an individual assessment.

For your submission to be graded, you must complete/submit the following:

- Zip your **entire project**, excluding the **node_modules/** folder.
- Rename your zip file: a3-studentname-studentid.zip. Replace studentname and studentid with your name and id.
- Upload the zip file to the assignment dropbox.
- **Only projects submitted as zip files will be graded. 7z, rar, or other file formats will not be graded.**

Due Date

- See course webpage
- Late submissions are accepted and deductions applied at 10% per day up to a maximum of 3 calendar days. No submissions accepted after 3 calendar days have passed.

Academic Integrity

- Please ensure that you have complied with the College's Academic Integrity Policy.
- Providing or obtaining "references" to/from others is not okay
- Using partial or full solutions found on the internet is not okay
- Recruiting another person/entity to complete your assignment is not okay

Problem Description

In this assignment, you will develop a lottery ticket machine for the fictional lottery game, WHE WHE.

In WHE WHE, a player must choose a maximum of 3 numbers and an "amount" they want to bet. At a future point in time, the lottery company will draw the winning numbers for the week. If a customer's ticket contains some or all of the winning numbers, the customer wins a multiplier of the amount they bet. In other words, the more money you bet, the more money they will win.

You have been hired as an Angular Developer to develop the user interface that will allow the lottery ticket company to generate a customer' ticket (ie: the UI to allow the customer to select up to 3 numbers and assign a money value of it).

You are only required to develop the user interface to generate the ticket. You are not required to implement the logic to choose winning numbers, determining winners, or calculating payouts.

The user interface you must develop should be similar to the following screenshots:

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

Figure 1 - Overall interface

- Notice that initially, the "GENERATE TICKET" button is disabled

Who Who Ticket Generator

Choose Bet Amount

\$0 \$1 \$5 \$10 \$20

Total so far: \$0

Choose 3 numbers

1 2 3 4 5 6 7 8 9 10

Random Clear

Cash Out

Generate Ticket

Figure 2 - Selecting numbers

- The "GENERATE TICKET" button is enabled after both numbers and bet amount are selected

Who Who Ticket Generator

Choose Bet Amount

\$0 \$1 \$5 \$10 \$20

Total so far: \$31

Choose 3 numbers

1 2 3 4 5 6 7 8 9 10

Random Clear

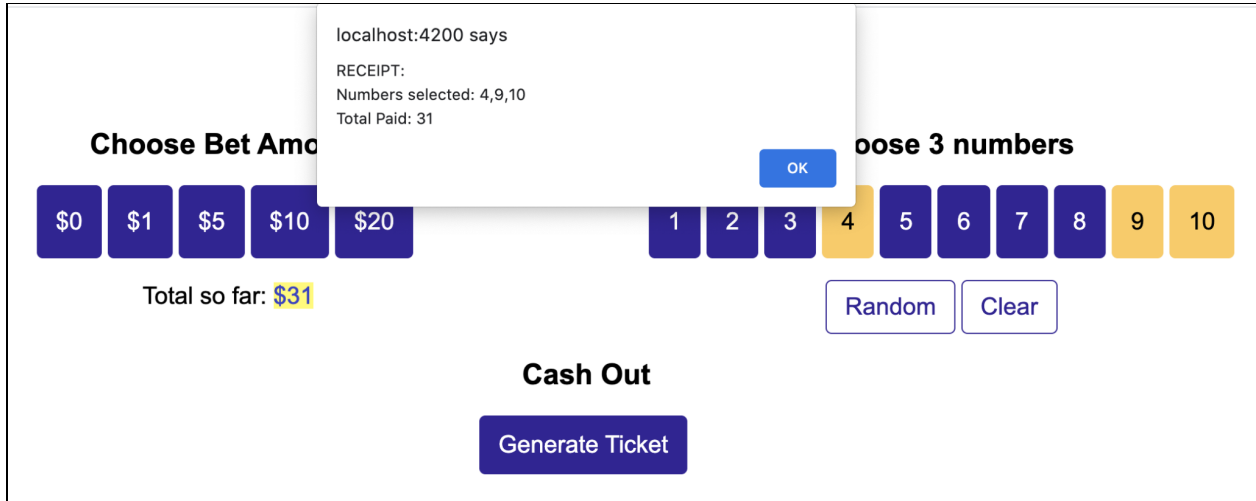
Cash Out

Generate Ticket

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

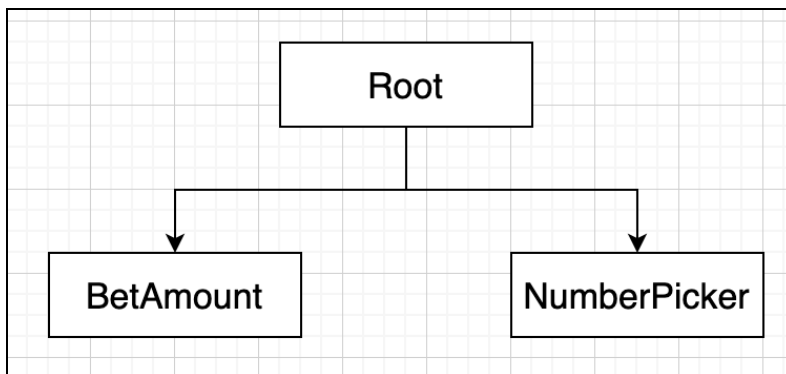
Figure 3 - Generating a Ticket

- After tapping **GENERATE TICKET**, an alert box displays the customer's selected numbers and desired bet amount

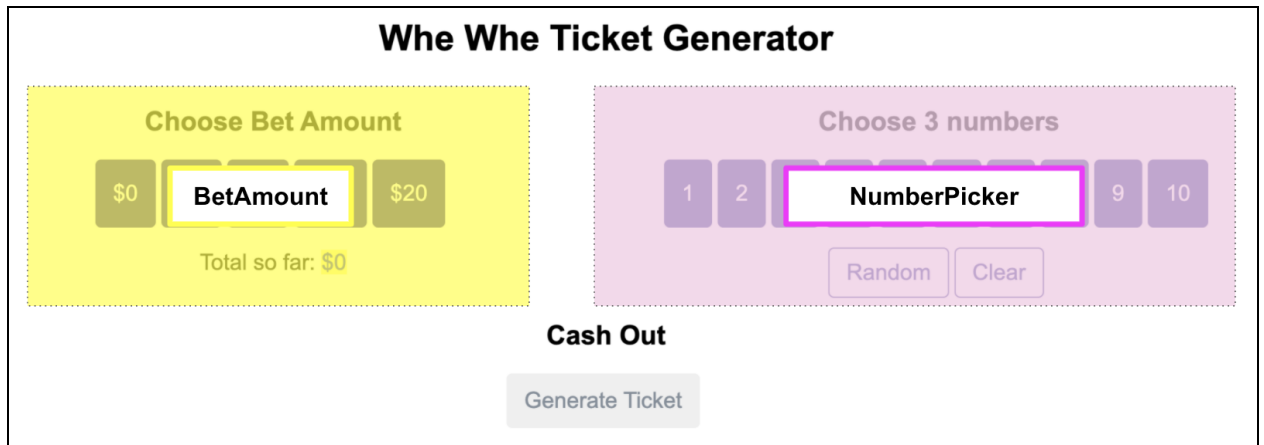


Detailed Technical Guidelines

1. The app **MUST** be built with Angular. The design of the data flow between components must make appropriate use of property binding and event emitters.
2. The app must be composed of the components in the following component diagrams.



This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.



The “Whe Whe Ticket Generator” header, “Cash Out” header, and “Generate Ticket” button are part of the Root component.

3. The look and the feel of the app does not have to be exactly the same as the screenshots shown Figures 1 to 3. However, the app must be reasonably pretty and must be obvious to the user where everything is (eg: number picker, bet selector, etc). It must also be clear what is clickable vs. disabled.
4. When the app loads, the user must be presented with a screen similar to **Figure 1**. By default, the GENERATE TICKET button is **disabled**. The button becomes **enabled** when the user selects a bet amount **AND** a minimum of 1 lottery number.
5. The user **MUST** choose **up to 3 numbers** between 1 to 10. This is done by hitting the appropriate buttons displayed in Figure 1 and 2. If the user tries to choose a fourth number, they cannot. The user can only choose a number **ONCE** (no duplicate numbers)
6. Each time the user selects a number, the user interface must update to clearly indicate that the number has been selected. For example, you can change the color of the number button, or add a border around the button.
7. If the user cannot decide what numbers they want to select, they may press the **RANDOM** button. When the **RANDOM** button is pressed, the app should randomly select 3 random numbers and update the UI accordingly. The app will **NOT** select duplicate numbers.
8. The user can reset their number choices by pressing the **CLEAR** button. When **CLEAR** is pressed, the user interface of each number button must reset to its default styling.

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

9. In addition to selecting numbers, the user must assign a bet value to their chosen numbers. For example, if the user wants to bet \$19 on their ticket, they can press the \$10 button once, the \$5 button once, and the \$1 button four times.
10. The total amount the user wants to bet must be shown in the **Choose Bet Amount** section.
11. The user can reset the amount of money they want to bet by pressing the \$0 button.
12. After selecting 3 numbers (either manually or randomly) AND placing a bet amount, the user generates a ticket by pressing the GENERATE TICKET button. When the button is pressed, the app will display an alert() indicating:
 - The numbers that were selected, sorted, in ascending order
 - The total amount the person bet
13. You are **NOT** allowed to use local or session storage for this assignment.
14. You are **NOT** allowed to use vanilla Javascript or JQuery DOM manipulation. No document.querySelector(...), \$(...), addEventListener(...), etc permitted. Your job is to demonstrate your competency with Angular and Angular's paradigms, eg: component based design, data passing/sharing, directives, etc.
15. When performing equality operations, you must use ===. Marks will be deducted for usage with == .
16. When declaring variables, you must use let/const syntax. Marks will be deducted for inappropriate (and general) usage of var
17. You may use either traditional function() syntax or arrow ()=>{} syntax. However, you must select a single style and use it consistently throughout your application. For example, if you choose to declare your functions using traditional function() syntax, then all your functions must be declared in this style. Marks deducted for inconsistent syntax.

This assessment contains materials that may be subject to copyright and other intellectual property rights. Modification, distribution or reposting of this document is strictly prohibited. Learners found reposting this document or its solution anywhere will be subject to the college's Academic Integrity policy.

Rubric

Criteria	Marks
• App user interface is complete and polished.	3
• App architecture conforms to the components described in the requirements	3
• When user presses a number, the number button's UI must update to clearly indicate that it is selected. Max 3 numbers can be selected	5
• Random button generates numbers per requirements	3
• Pressing the CLEAR button removes all previously selected numbers	2
• User can select a bet amount and value is displayed	5
• Generate Ticket button correctly displayed required results	5

Total : 26 MARKS