

<b>Course: Database Connectivity using Java</b>	<b>Midterm</b>	<b>Contribution : 20% of course</b>
<b>Instructor:Mehrnaz Zhian</b>	<b>Date Given: Thursday June5, 2025</b>	<b>Date Due: June12, 2025 @ 11:59pm</b>

**Notes for the Student:** This Midterm is designed to test your ability to build simple client side web applications using React.

**Background:** You will need to have access to a Code Editor and have a thorough understanding of React.

### ***Midterm Submission Requirements***

- Please upload the following:
  - The URL of your Github repository if you have an account
  - Upload the source code to Blackboard (delete the node\_modules folder before uploading).
  - A video (with audio) of you demonstrating your application in accordance to the requirements.

### ***Midterm Regulations***

- **NO LATE SUBMISSIONS WILL BE ACCEPTED.**

**Presentation will take place in person – live -No videos is allowed for presentation**

**I will ask questions during the presentation, if no reply → No mark**

- Please review Seneca's policies on Academic Integrity, specifically:

*"Each student should be aware of the College's policy regarding Cheating and Plagiarism. Seneca's Academic Policy will be strictly enforced. To support academic honesty at Seneca College, all work submitted by students may be reviewed for authenticity and originality, utilizing software tools and third party services. Please visit the Academic Honesty site on <http://library.senecacollege.ca> for further information regarding cheating and plagiarism policies and procedures.*

*." Thus, ensure that your code or any part of it is not duplicated by another student(s). This will result in a percentage of zero (0%) assigned to all parties involved*

## Detailed Specification

You have been asked to develop a **Cash Register System** that is designed to function within a fictional lottery system called “WHE WHE”.

In the realm of “WHE WHE”, to participate, a customer MUST choose 5 numbers and then assign a money value to their ticket. The more money a customer attaches to their ticket, the higher their pay out would be if their 5 selected numbers are drawn.

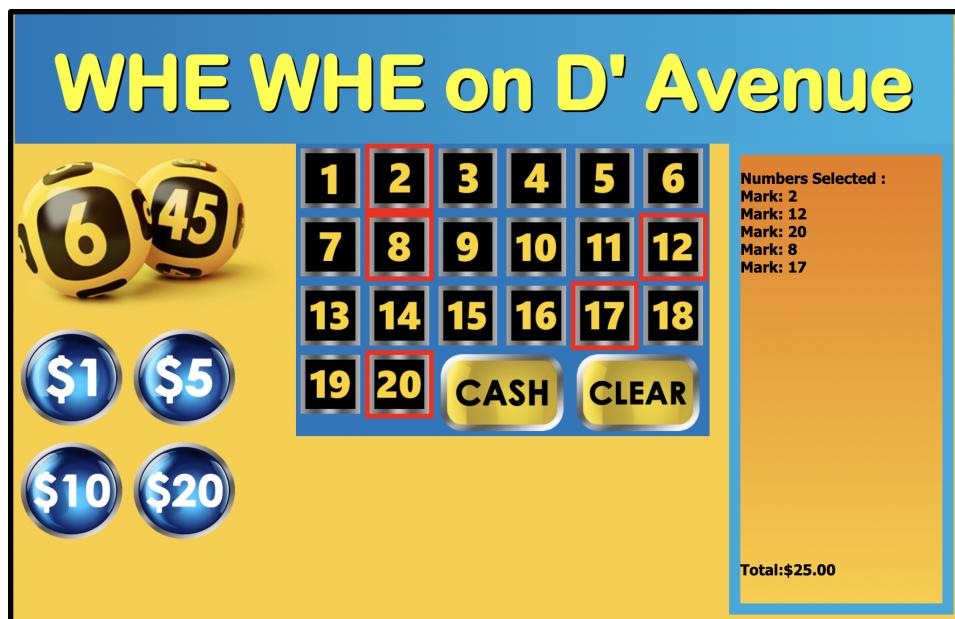
You have been “hired” as a React Developer to develop the “Cash Register System” that will allow a cashier to generate a customer’s ticket, i.e, select 5 numbers and assign a money value to it.

Well-designed web interfaces must be developed similar to the ones presented below:

**Figure 1.1**



**Figure 1.2**



#### SEE BELOW FOR PROGRAMMING GUIDELINES:

1. This app **MUST** be built using React.
2. Ensure that you break up your app into pertinent components. You are not allowed to have only 1 component. Note, your components must be meaningful and useful.
3. With regards to the look and feel, your app does not have to look exactly like the above screenshots but must be well structured.
4. When the app loads, the cashier must be presented with a screen similar to **Figure 1.1**.
5. A cashier **MUST** choose 5 numbers between 1 to 20. This is done by hitting the appropriate buttons displayed in Figure 1 and & 2. If the cashier tries to choose a 6th number, they must be alerted that they can't do so.
6. Everytime the cashier presses a number, the number must change in color or a border is placed around the number. Additionally, as shown in **Figure 1.2**, the number must then be added and listed to the **Numbers Selected section**, to the right of the screen. This is to indicate to the cashier which numbers have been selected thus far.
7. The cashier is allowed to unselect numbers. If a number is unselected, it must be removed from the Numbers Selected Section and the appropriate button must change back to its initial color.
8. **After** the 5 numbers are selected, the cashier has to assign a money value to the customer's ticket. For example, if the customer wants to play \$23.00 on their ticket, then the cashier can hit the \$20 money value button once and then hit the \$1.00 money value button 3 times. Note, this can ONLY be done after the cashier selects the 5 numbers. If

the cashier tries to assign money before selecting the 5 numbers, they must be alerted that they can't do so.

9. The Total money value must also be added to the **Numbers Selected section**, as shown in **Figure 1.2**
10. When the cashier hits the “**Clear**” button, all selected numbers must be cleared and unselected. Also, if money value was assigned, that should be cleared too.
11. Add a “Random” button. When the cashier presses this button. The system must randomly select 5 random numbers, select them and them to the “**Numbers Selected Section**”
12. Display the numbers selected and the money value assigned when the cashier presses the “**Cash**” button
13. You are **NOT** allowed to use local or session storage for this assignment.
14. You are not allowed to build this application as if you were building it using Vanilla JS. Marks will be significantly deducted if you do so. You are required to demonstrate your competence with React by adequately using props, state, styling your elements the React way and breaking up your U.I into reasonable and useful components.

## Rubric

Criteria	Not Implemented	Fully Implemented
<ul style="list-style-type: none"> <li>• The app loads like Figure 1.1 upon start up.</li> <li>• Everytime the cashier presses a number, the number must change in color or a border is placed around the number.</li> <li>• Each pressed number must then be added and listed to the Numbers Selected section, to the right of the screen.</li> <li>• The cashier is allowed to unselect numbers.</li> <li>• Unselected numbers must be removed from the Numbers Selected Section.</li> <li>• Cashier has to assign a money value to the customer's ticket (5 numbers).</li> <li>• The Total money value must also be added to the <b>Numbers Selected section</b>.</li> <li>• Clear Button functionality</li> <li>• Random Button Functionality</li> </ul>		1 2 3 2 3 2 2 2 2 2 5
<ul style="list-style-type: none"> <li>• Relevant and meaningful Components</li> <li>• Look and Feel</li> </ul>		<b>3</b> <b>5</b>

**Total : 30 MARKS**