Government Polytechnic Himatnagar
Computer Engineering Department
Nr Motipura Circle, Gadhoda Road, Himatnagar(S.K.)-383001 Gujarat.
Email: gphcomputer@gmail.com Website: computer.gphimatnagar.in

A

PROJECT REPORT

ON

"Battleship Console Game Using JAVA"

FOR COURSE
ADVANCED OBJECT ORIENTED PROGRAMMING
(4340701)

GUIDED BY:

Ms. A.R.SHAH

PREPARED BY:

Suthar Manthan Sumantkumar: 226240307068

DIPLOMA IN 4TH SEMESTER

COMPUTER ENGINEERING DEPARTMENT

Government Polytechnic Himatnagar
Computer Engineering Department
Nr Motipura Circle, Gadhoda Road, Himatnagar(S.K.)-383001 Gujarat.
Email: gphcomputer@gmail.com Website: computer.gphimatnagar.in

CERTIFICATE

This is to certify that the micro - project entitled “Battleship Console Game Using JAVA” submitted by Suthar Manthan Sumantkumar(226240307068) towards the partial fulfillment of the requirement for Continuous Assessment (CA) to facilitate integration of Course Outcomes (COs) in Advanced Object Oriented Programming (4340701) is a record of the work carried out by him/her under my guidance and supervision. The work submitted, in my opinion, has to a level required for being accepted for the examination.

Guided By:

Mrs. A. R. Shah
Government Polytechnic,
Himatnagar – 380001

Head of the Department:

Mrs. M. P. Mehta
Government Polytechnic,
Himatnagar - 380001

TABLE OF CONTENTS

No.	Point	Page
1	Brief Description	1
2	Aim of Micro Project	1
3	Course Outcomes Integrated	1
4	Actual Procedure Followed	2-9
5	Outputs of the Micro-projects	10-11
6	Actual Resources Used	11
7	Skill Developed/ learning out of this Micro-Project	12

1.0 Brief Description

"Battleship Console Game Using JAVA" is a micro project will involve creating a console-based interface where players can interact with the game by inputting their guesses for the ship locations (as row and column) and receiving feedback on whether their guesses were hits or misses. The game will be one player, where the player will play and the result will be count as per how many attempt the user get to hit the number of ship as per the rules given by in the game already. The game will include features such as displaying the game board, keeping track of the numbers of how many ships were hit and multiple options of the grid of game board i.e. 4x4, 5x5, 6x6, 8x8; each layout contain different rules.

2.0 Aim of Micro Project

"Battleship Console Game Using JAVA" is a micro project aimed at implementing the classic Battleship game in the Java programming language. In this project, players will engage in a strategic naval battle type game where they have to guess the coordinates of the ship in game board grid and attempt to sink the ships by guessing their coordinates and if the guessed coordinates will contains the ship then the ship will sink which means hit; and if the ship does not found at coordinates then it is miss.

3.0 Course Outcomes Integrated

4340701.1	Write simple java programs for a given problem statement.
4340701.2	Use object oriented programming concepts to solve real world problems.
4340701.3	Develop an object-oriented program using inheritance and package concepts for a given problem statement.
4340701.4	Develop an object oriented program using multithreading and exception handling for a given problem statement.
4340701.5	Develop an object-oriented program by using the files and collection framework.

4.0 Actual Procedure Followed

```
import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;

public class BattleshipGame {

   public static Scanner input = new Scanner(System.in);
   public static void main(String[] args) {
      boolean playAgain = false;
      do{
         int userSize = 0;
         System.out.println("\n<-- Welcome to Battleship console game -->");
         System.out.println("\nEnter '1' for 4x4 board\nEnter '2' for 5x5 board\nEnter '3' for
6x6 board\nEnter '4' for 8x8 board");
         System.out.println("Note: The more size of matrix of game board, the more hard the
game will.\n");
         do{
         System.out.print("Enter the number of Game board you want to play in: ");
         userSize = input.nextInt();
         } while(userSize < 1 || userSize > 4);

         switch (userSize) {
            case 1:Game4x4();
               break;

            case 2:Game5x5();
               break;

            case 3:Game6x6();
               break;

            case 4:Game8x8();
               break;
         }
         char playAgainChoice='y';
         do{
            System.out.print("Wanna play again? [Y]yes or [N]no:");
            playAgainChoice = Character.toUpperCase(input.next().charAt(0));
         } while (playAgainChoice != 'Y' && playAgainChoice != 'N');
         if(playAgainChoice == 'Y'){
            playAgain = true;
         } else {
            playAgain = false;
         }
      }while(playAgain == true);
   }


   //method to create a game for 4x4
   public static void Game4x4(){
      String boardName = "Game4x4";
```

```
        int Attempts = 0;
        int gameBoardLength = 4; //lenght of x and y axis of board
        char water = '-'; //char to fill water in board
        char ship = 'S'; //char to fill ship in board
        char hit = 'X'; //char to fill hit symbol after hiting a ship
        char miss = 'O'; //char to fill miss symbol after missing hit
        int shipNumber = 3; //total number of ships to be hit

        System.out.println("\n___YOU SELECT THE 4x4 BOARD TO PLAY GAME___");
        System.out.println("------------------RULES--------------------");
        System.out.println("> You have to enter row and column numbers to hit at the particular target in game board.");
        System.out.println("> If the ship found in that target it will hit otherwise its miss.");
        System.out.println("> You have to hit 3 ships in 4x4 game board to win!\n");

        char[][] gameBoard = createGameBoard(gameBoardLength, water, ship, shipNumber); //creating game board
        showGameBoard(gameBoard, water, ship); //displaying Game Board
        int undetectedShipNumber = shipNumber;
        while(undetectedShipNumber > 0){
            Attempts++;
            int[] guessCoordinates = getUserCoordinates(gameBoardLength); //user's row and column choices stored in one-dimentional array
            char[] locationViewUpdate = evaluateUserGuessAndGetTheTarget(guessCoordinates, gameBoard, ship, water, hit, miss); //this char is to store what contains in the targated place in game board
            if (locationViewUpdate[0] == hit && locationViewUpdate[1]=='H') { //if the user's targated place contains ship the value of locationVeiwUpadte will be 'X' which is hit that means ship is hit
                undetectedShipNumber--; //decrementing whenever ship found to confirm how many ship is left to hit
                /*
                 * when undetectedShipNumber will be 0 it means there is no ship is left to hit
                 * and all the given ships are hitted or found
                 * so this while loop will terminate
                 */
            }
            gameBoard = updateGameBoard(gameBoard, guessCoordinates, locationViewUpdate); //update method calling to update the game board after users guessed place
            showGameBoard(gameBoard, water, ship); //displaying game board after users guess
        }
        winOutput(Attempts, boardName, shipNumber);//displaying won message after user succesfully hit all the ships
    }

    //method to create a game for 5x5
    //all comments from 4x4 is same here
    public static void Game5x5(){
        String boardName = "Game5x5";
        int Attempts = 0;
        int gameBoardLength = 5;
        char water = '-';
        char ship = 'S';
```

```
        char hit = 'X';
        char miss = 'O';
        int shipNumber = 4;

        System.out.println("\n___YOU SELECT THE 5x5 BOARD TO PLAY GAME___");
        System.out.println("-----------------RULES--------------------");
        System.out.println("> You have to enter row and column numbers to hit at the particular target in game board.");
        System.out.println("> If the ship found in that target it will hit otherwise its miss.");
        System.out.println("> You have to hit 4 ships in 5x5 game board to win!\n");

        char[][] gameBoard = createGameBoard(gameBoardLength, water, ship, shipNumber); //creating game board
        showGameBoard(gameBoard, water, ship); //displaying Game Board
        int undetectedShipNumber = shipNumber;
        while(undetectedShipNumber > 0){
            Attempts++;
            int[] guessCoordinates = getUserCoordinates(gameBoardLength);
            char[] locationViewUpdate = evaluateUserGuessAndGetTheTarget(guessCoordinates, gameBoard, ship, water, hit, miss);
            if (locationViewUpdate[0] == hit && locationViewUpdate[1] == 'H') {
                undetectedShipNumber--;
            }
            gameBoard = updateGameBoard(gameBoard, guessCoordinates, locationViewUpdate); //update method calling to update the game board after users guessed place
            showGameBoard(gameBoard, water, ship); //displaying game board after users guess
        }
        winOutput(Attempts, boardName, shipNumber);//displaying won message after user succesfully hit all the ships
    }

    //method to create a game for 6x6
    //all comments from 4x4 is same here
    public static void Game6x6(){
        String boardName = "Game6x6";
        int Attempts = 0;
        int gameBoardLength = 6;
        char water = '-';
        char ship = 'S';
        char hit = 'X';
        char miss = 'O';
        int shipNumber = 6;

        System.out.println("\n___YOU SELECT THE 6x6 BOARD TO PLAY GAME___");
        System.out.println("-----------------RULES--------------------");
        System.out.println("> You have to enter row and column numbers to hit at the particular target in game board.");
        System.out.println("> If the ship found in that target it will hit otherwise its miss.");
        System.out.println("> You have to hit 6 ships in 6x6 game board to win!\n");

        char[][] gameBoard = createGameBoard(gameBoardLength, water, ship, shipNumber); //creating game board
        showGameBoard(gameBoard, water, ship); //displaying Game Board
```

```
        int undetectedShipNumber = shipNumber;
        while(undetectedShipNumber > 0){
            Attempts++;
            int[] guessCoordinates = getUserCoordinates(gameBoardLength);
            char[] locationViewUpdate = evaluateUserGuessAndGetTheTarget(guessCoordinates,
gameBoard, ship, water, hit, miss);
            if (locationViewUpdate[0] == hit && locationViewUpdate[1] == 'H') {
                undetectedShipNumber--;
            }
            gameBoard = updateGameBoard(gameBoard, guessCoordinates,
locationViewUpdate); //update method calling to update the game board after users guessed
place
            showGameBoard(gameBoard, water, ship); //displaying game board after users guess
        }
        winOutput(Attempts, boardName, shipNumber);//displaying won message after user
succesfully hit all the ships
    }

    //method to create a game for 8x8
    //all comments from 4x4 is same here
    public static void Game8x8(){
        String boardName = "Game8x8";
        int Attempts = 0;
        int gameBoardLength = 8;
        char water = '-';
        char ship = 'S';
        char hit = 'X';
        char miss = 'O';
        int shipNumber = 7;
        System.out.println("\n___YOU SELECT THE 8x8 BOARD TO PLAY GAME___");
        System.out.println("------------------RULES--------------------");
        System.out.println("> You have to enter row and column numbers to hit at the particular
target in game board.");
        System.out.println("> If the ship found in that target it will hit otherwise its miss.");
        System.out.println("> You have to hit 7 ships in 8x8 game board to win!\n");

        char[][] gameBoard = createGameBoard(gameBoardLength, water, ship, shipNumber);
        showGameBoard(gameBoard, water, ship);
        int undetectedShipNumber = shipNumber;
        while(undetectedShipNumber > 0){
            Attempts++;
            int[] guessCoordinates = getUserCoordinates(gameBoardLength);
            char[] locationViewUpdate = evaluateUserGuessAndGetTheTarget(guessCoordinates,
gameBoard, ship, water, hit, miss);
            if (locationViewUpdate[0] == hit && locationViewUpdate[1] == 'H') {
                undetectedShipNumber--;
            }
            gameBoard = updateGameBoard(gameBoard, guessCoordinates,
locationViewUpdate);
            showGameBoard(gameBoard, water, ship);
        }
        winOutput(Attempts, boardName, shipNumber);//displaying won message after user
succesfully hit all the ships
    }
```

```
    //method to create game board
    private static char[][] createGameBoard(int gameBoardLength, char water, char ship, int
shipNumber) {
        char[][] gameBoard = new char[gameBoardLength][gameBoardLength]; //board using
two-dimensional
        for(char[] row : gameBoard){ //iterating through every row in gameBoard
            Arrays.fill(row, water); //filling every row with water at first
        }
        return placeShip(gameBoard, shipNumber, water, ship); //calling a method directly to
place ships after creating game board with water
    }

    //method to update game board as per users guess to hit ship
    private static char[][] updateGameBoard(char[][] gameBoard, int[] guessCoordinates,
char[] locationViewUpdate) {
        int row = guessCoordinates[0]; //copying row number given by user in row element
from array
        int col = guessCoordinates[1]; //copying column number given by user in col element
from array
        gameBoard[row][col] = locationViewUpdate[0]; //changing the target place with users
guessed value
        return gameBoard;
    }

    //method to evaluate users guessed target and check what's at that target
    protected static int hittedShip = 0;
    private static char[] evaluateUserGuessAndGetTheTarget(int[] guessCoordinates, char[][]
gameBoard, char ship, char water,
            char hit, char miss) {
        String message;
        char mesg;
        int row = guessCoordinates[0]; //copying row number given by user in row element
from array
        int col = guessCoordinates[1]; //copying column number given by user in col element
from array
        char target = gameBoard[row][col]; //storing what is contains in the user's targated place

        if (target == ship) {
            hittedShip++;
            message = "Hit!"; //if the target contains ship, it will be hit
            mesg = 'H';
            target = hit; //storing hit char 'X' in target
        } else if (target == water) {
            message = "Miss!"; //if the target contains water and does not contains ship, it will
miss
            mesg = 'M';
            target = miss; //storing miss char 'O' in target
        } else {
            message = "Already Hit!"; //when when target does not contains either ship or water it
simply means it is already hitted by user and the 'X' hit char is stored at that place
            mesg = 'A';
        }
        System.out.println(message); //displaying message
```

```
        return new char[]{target,mesg};
    }

    //getting input number of row and column for game board
    private static int[] getUserCoordinates(int gameBoardLength) {
        int row, col;
        //inputing number for row
        do{
            System.out.print("Row: ");
            row = input.nextInt();
        } while(row < 1 || row > gameBoardLength); //if user input invalid number eg. a number
out of bound or range of the game board which is 4

        //inputing number for column
        do{
            System.out.print("Column: ");
            col = input.nextInt();
        } while(col < 1 || col > gameBoardLength);
        return new int[]{row - 1, col - 1}; //returning the decremented numbers to set that
number as index
    }

    //method to print Game board in output
    private static void showGameBoard(char[][] gameBoard, char water, char ship) {
        int gameBoardLength = gameBoard.length; //length of game board which will be 4
        System.out.print("  "); //this space is to display the numbers for column in format
        for(int i = 0; i < gameBoardLength; i++){ //printing the number of columns
            System.out.print(i + 1 + " ");
        } System.out.println(); //for formated output

        for(int row = 0; row < gameBoardLength; row++){
            System.out.print(row + 1 + " "); //printing the number of rows
            for(int col = 0; col < gameBoardLength; col++){ //printing the Game Board
                char position = gameBoard[row][col]; //storing what is in the position after every
iteration
                if(position == ship){
                    System.out.print(water + " "); //if the position contains ship then the water '-' will
be print to hide that from the user for no cheating
                } else {
                    System.out.print(position + " "); //if the position does not cointains ship
                }
            } System.out.println(); //for formated output
        } System.out.println("Hitted ship number: "+hittedShip+"\n"); //this will print how many
ship has been hit by user after each try
    }

    //method to place ship in game board at random places
    private static char[][] placeShip(char[][] gameBoard, int shipNumber, char water, char
ship) {
        int placedShip = 0; //to Attempts how many ships has been placed
        int gameBoardLength = gameBoard.length; //length of game board which will be 4
        while (placedShip < shipNumber) {
            //generateShipCoordinates method will return the random coordinates to place ship at
that coordinates in the game board
```

```
            int[] location = generateShipCoordinates(gameBoardLength); //calling
generateShipCoordinates method to find where the ship will be placed in gameBoard array
and storing that data in location[][] array
            char possiblePlacement = gameBoard[location[0]][location[1]]; //checking if the
placement is possible or not. it not possible when the location in gameBoard contains any
other than water
            if(possiblePlacement == water){
                gameBoard[location[0]][location[1]] = ship; //if the location only contains water
then ship will be placed
                placedShip++;
            }
        }
        return gameBoard;
    }

    //method to generate random coordinates for ship placing
    private static int[] generateShipCoordinates(int gameBoardLength) {
        int[] coordinates = new int[2]; //creating an array with one row and two column to store
x-axis and y-axis numbers
        for(int i=0; i < coordinates.length; i++) {
            coordinates[i] = new Random().nextInt(gameBoardLength); //storing random
numbers so the game will change the place of ship every time
        }
        return coordinates;
    }

    //method to print winner output after completion of game
    private static void winOutput(int Attempts, String boardName, int shipNumber){

        //if..else to first check what game board user have chose to play in and give output
accordingly
        if (boardName == "Game4x4") {
            if (Attempts >= 3 && Attempts <= 5) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take only "+Attempts+" Attempts to win in a 4x4 board,
Wonderful!");
            } else if (Attempts > 5 && Attempts <= 10) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take "+Attempts+" Attempts to win in a 4x4 board,
Good!");
            } else {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 4x4 board,
you can try again to perform more.");
            }
        } else if (boardName == "Game5x5") {
            if (Attempts >= 4 && Attempts <= 6) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take only "+Attempts+" Attempts to win in a 5x5 board,
Wonderful!");
            } else if (Attempts > 6 && Attempts <= 15) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take "+Attempts+" Attempts to win in a 5x5 board,
Good!");
```

```
            } else if (Attempts > 15 && Attempts <= 20) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 5x5 board,
good but you can perform more.");
            } else {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 5x5 board,
you can try again to perform more.");
            }
        } else if (boardName == "Game6x6") {
            if (Attempts >= 6 && Attempts <= 8) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take only "+Attempts+" Attempts to win in a 6x6 board,
Wonderful!");
            } else if (Attempts > 8 && Attempts <= 15) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take "+Attempts+" Attempts to win in a 6x6 board,
Good!");
            } else if (Attempts > 15 && Attempts <= 20) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 6x6 board,
good but you can perform more.");
            } else {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 6x6 board,
as the game board is complex it is obviously hard but you can try again to perform more.");
            }
        } else if (boardName == "Game8x8") {
            if (Attempts >= 7 && Attempts <= 10) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take only "+Attempts+" Attempts to win in a 8x8 board,
Wonderful!");
            } else if (Attempts > 10 && Attempts <= 17) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take "+Attempts+" Attempts to win in a 8x8 board,
Good!");
            } else if (Attempts > 17 && Attempts <= 22) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 8x8 board,
good but you can perform more.");
            } else if (Attempts > 22 && Attempts <= 30) {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 8x8 board,
it's a big game board so it is obvious to take more attempts.");
            } else {
                System.out.println("You hit "+shipNumber+" Ships successfully, You Won!");
                System.out.println("You take total "+Attempts+" Attempts to win in a 8x8 board,
as the game board is complex it is obviously hard but you can try again to perform more.");
            }
        }
    }
}
```

5.0 Outputs of the Micro-projects

5.1 Screenshot of complete output:

```
D:\Manthan's Doc\DIPLOMA SEM4\AOOP\Codes\Micro Project\Battleship Console Game\Main Code>javac BattleshipGame.java

D:\Manthan's Doc\DIPLOMA SEM4\AOOP\Codes\Micro Project\Battleship Console Game\Main Code>java BattleshipGame

<-- Welcome to Battleship console game -->

Enter '1' for 4x4 board
Enter '2' for 5x5 board
Enter '3' for 6x6 board
Enter '4' for 8x8 board
Note: The more size of matrix of game board, the more hard the game will.

Enter the number of Game board you want to play in: 1

___YOU SELECT THE 4x4 BOARD TO PLAY GAME___
------------------RULES--------------------
> You have to enter row and column numbers to hit at the particular target in game board.
> If the ship found in that target it will hit otherwise its miss.
> You have to hit 3 ships in 4x4 game board to win!

  1 2 3 4
1 - - - -
2 - - - -
3 - - - -
4 - - - -
Hitted ship number: 0

Row: 1
Column: 2
Miss!
  1 2 3 4
1 - O - -
2 - - - -
3 - - - -
4 - - - -
Hitted ship number: 0

Row: 3
Column: 3
Hit!
  1 2 3 4
1 - O - -
2 - - - -
3 - - X -
4 - - - -
Hitted ship number: 1

Row: 4
Column: 2
Hit!
  1 2 3 4
1 - O - O
2 O - O O
3 O - X -
4 O X O O
Hitted ship number: 2

Row: 1
Column: 1
Hit!
  1 2 3 4
1 X O - O
2 O - O O
3 O - X -
4 O X O O
Hitted ship number: 3

You hit 3 Ships successfully, You Won!
You take total 12 Attempts to win in a 4x4 board, you can try again to perform more.
Wanna play again? [Y]yes or [N]no:e
Wanna play again? [Y]yes or [N]no:u
Wanna play again? [Y]yes or [N]no:n
Thank You, will come again to play!
D:\Manthan's Doc\DIPLOMA SEM4\AOOP\Codes\Micro Project\Battleship Console Game\Main Code>
```

5.2 Output of validating only valid input from user:

5.2.1 Game board selecting:

```
<-- Welcome to Battleship console game -->

Enter '1' for 4x4 board
Enter '2' for 5x5 board
Enter '3' for 6x6 board
Enter '4' for 8x8 board
Note: The more size of matrix of game board, the more hard the game will.

Enter the number of Game board you want to play in: 5
Enter the number of Game board you want to play in: 6
Enter the number of Game board you want to play in: 7
Enter the number of Game board you want to play in: -1
Enter the number of Game board you want to play in: 1

___YOU SELECT THE 4x4 BOARD TO PLAY GAME___
------------------RULES--------------------
> You have to enter row and column numbers to hit at the particular target in game board.
> If the ship found in that target it will hit otherwise its miss.
> You have to hit 3 ships in 4x4 game board to win!

  1 2 3 4
1 - - - -
2 - - - -
3 - - - -
4 - - - -
Hitted ship number: 0

Row:
```

5.2.2 Input rows and columns:

```
Row: 5
Row: 6
Row: -1
Row: 2
Column: 5
Column: 3
Already Hit!
  1 2 3 4
1 - - - -
2 - - O -
3 - - - -
4 - - - -
Hitted ship number: 0

Row:
```

6.0 Actual Resources Used

Sr. No	Name of Resource/material	Remarks
1	Hardware: Computer System	Computer (AMD), RAM: 8 GB
2	Software: Operating System	Windows 10 Pro
3	Software	JDK 1.8.0 or above, Notepad++, VS Code or other text editor

7.0 Skill Developed/ learning out of this Micro-Project

Through this micro project, I gain hands-on experience with Java programming concepts such as object-oriented design, arrays, loops, conditional statements, and user input/output handling. I also learn problem-solving and algorithmic thinking while implementing the game logic and user interface. Overall, the "Battleship Console Game Using JAVA" micro project provides an engaging and practical way for Java learners to reinforce their programming skills while having fun with a classic game.

Here are few main skills I learnt in details:

1. Java Programming: Enhanced proficiency in Java programming language, including syntaxes and object-oriented principles.
2. Object-oriented design: Improved ability to design and implement object-oriented solutions to complex problems using class, objects and methods.
3. Algorithmic thinking: Strengthened problem-solving skills through implementing algorithms for game logic, such as ship placement and hit detection.
4. User Input Handling: Experience in handling user input validation and processing to ensure smooth interaction with the game by only inputting correct and reliable inputs from user.
5. Game State Management: Proficiency in managing the state of the game, including tracking ship positions, hits, and misses, as well as determining game outcomes.
6. Testing and Debugging: Practiced techniques for testing code functionality, identifying bugs, and debugging errors to ensure the game(or code) runs smoothly without any error.