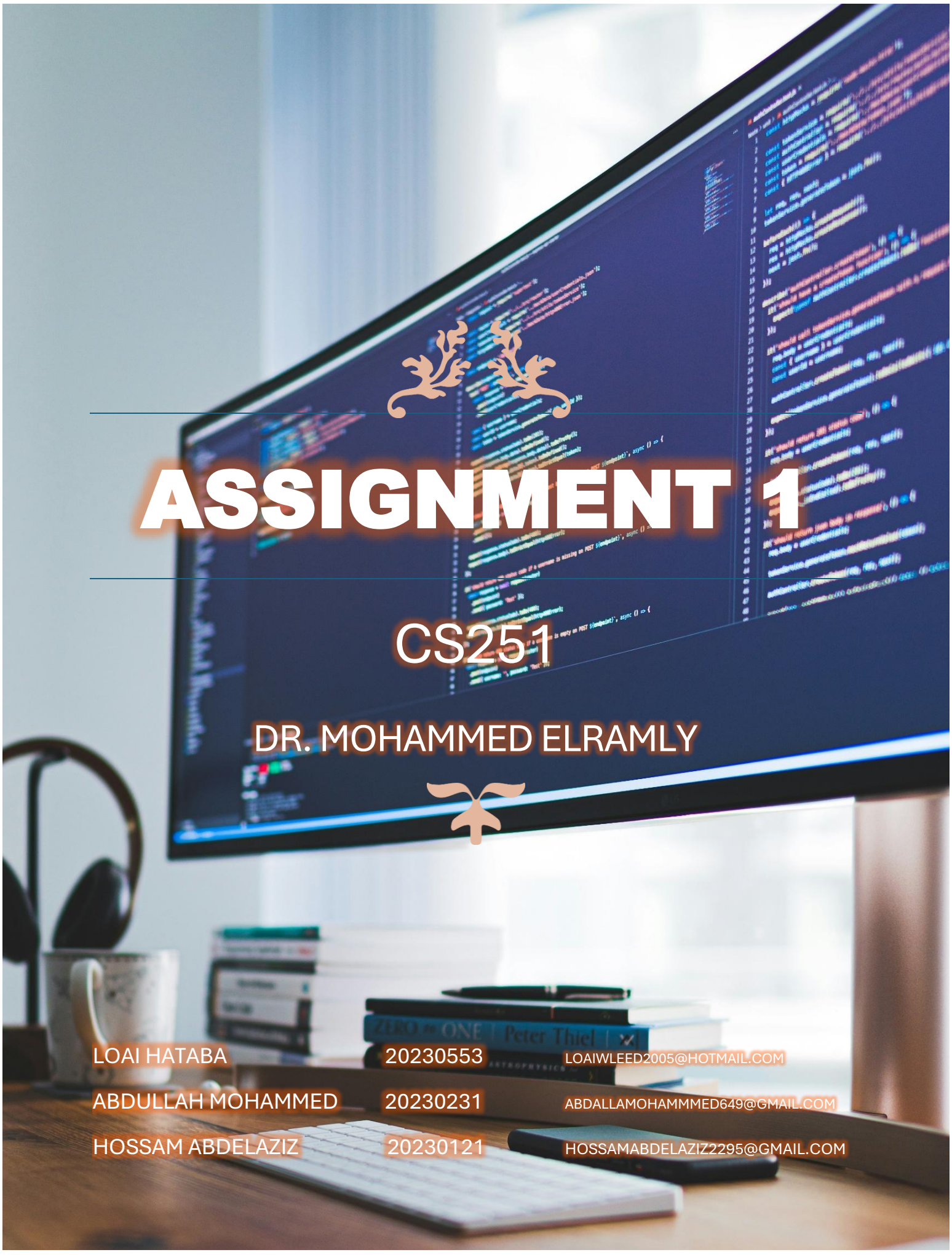


ASSIGNMENT 1

CS251

DR. MOHAMMED ELRAMLY

LOAI HATABA	20230553	LOAIWLEED2005@HOTMAIL.COM
ABDULLAH MOHAMMED	20230231	ABDALLAMOHAMMMED649@GMAIL.COM
HOSSAM ABDELAZIZ	20230121	HOSSAMABDELAZIZ2295@GMAIL.COM

Languages:

Loai → Java 

Abdullah → Java 

Hossam → Java 

Learning:

Name	Duration	Sources
Loai	8 Hours	https://youtube.com/playlist?list=PLJhTWoCm8I6DXaq7XECfyGKtsq4Z6fWZr&si=w_PmOmUhKEH6EyCl https://youtu.be/drQK8ciCAjY?si=cmx5cv4of_BQn4k5
Abdullah	4 Days	https://www.tutorialspoint.com/java/index.htm https://www.youtube.com/watch?v=mNvJipMTKSM&list=PLClnYL3l2AajYlZGzU_LVrHdoouf8W6ZN
Hossam	1 Day	https://www.youtube.com/watch?v=mNvJipMTKSM&list=PLClnYL3l2AajYlZGzU_LVrHdoouf8W6ZN https://www.youtube.com/playlist?list=PLJhTWoCm8I6DXaq7XECfyGKtsq4Z6fWZr

Food Alternative (App 1 Loai):

Main Function:

```
Scanner scanner = new Scanner(System.in);
// Load food from JSON
String jsonPath = "food/foodDictionary.json";
List<FoodItem> foodList = GsonTool.loadFood(jsonPath);
while (true){
        printBanner();
        int menu = optionsMenu(scanner);
        switch(menu)
        {
          // Alternative Food
          case 1:
            foodMenu(scanner, foodList);
            int ans = continueApp(scanner);
            if (ans == 0){
              scanner.close();
              System.exit(0);
            }
            break;
          //Add new Food
          case 2:
            addFood(scanner, foodList, jsonPath);
            int ans2 = continueApp(scanner);
            if (ans2 == 0){
              scanner.close();
              System.exit(0);
            }
```

```
            break;

        // Delete Food
        case 3:
            deleteFood(scanner, foodList, jsonPath);
            int ans3 = continueApp(scanner);
            if (ans3 == 0){
                scanner.close();
                System.exit(0);
            }
            break;

        case 4:
            prinInfoBanner();
            int ans4 = continueApp(scanner);
            if (ans4 == 0){
                scanner.close();
                System.exit(0);
            }
            break;

        case 5:
            System.out.println("\nGoodbye!!");
            scanner.close();
            System.exit(0);
            break;
      }
   }
}
```

Screenshots:


Welcome choose an option:
1)Find Alternative Food
2)Add new Food
3)Delete Food
4)Info
5)Exit
Choice: 2
Add Food: ma7shy wara2 3enab
Calories per 100g: 350
Alternatives:
How many Alternatives: 2
Food Name: ma7shy kromb
Conversion Factor: 1.0
Food Name: ma7shy kosa
Conversion Factor: 1.2
Food Added!
Do you want to continue? (Y/N)
{
"name": "Avocados",
"unit": "grams",
"calories": 160,
"alternatives": {
"Walnuts": 1.2,
"Almonds": 1.2,
"Olives": 1.1
}
},
{
"name": "Walnuts",
"unit": "grams",
"calories": 654,
"alternatives": {
"Cashews": 1.1,
"Almonds": 1.0,
"Pistachios": 1.1
}
},
{
"name": "Pizzaya",
"unit": "gram",
"calories": 350,
"alternatives": {
"koki": 1.6,
"cake": 1.5
}
},
{
"name": "Tuna",
"unit": "grams",
"calories": 150,
"alternatives": {
"salmon": 1.1
}
},
{
"name": "ma7shy wara2 3enab",
"unit": "grams",
"calories": 350,
"alternatives": {
"ma7shy kosa": 1.2,
"ma7shy kromb": 1.0
}
}

Welcome choose an option:

1)Find Alternative Food
2)Add new Food
3)Delete Food
4)Info
5)Exit

Choice: 3

Delete Food:
Food Name: fera5
Couldn't find fera5 anywhere :(

Do you want to continue? (Y/N)

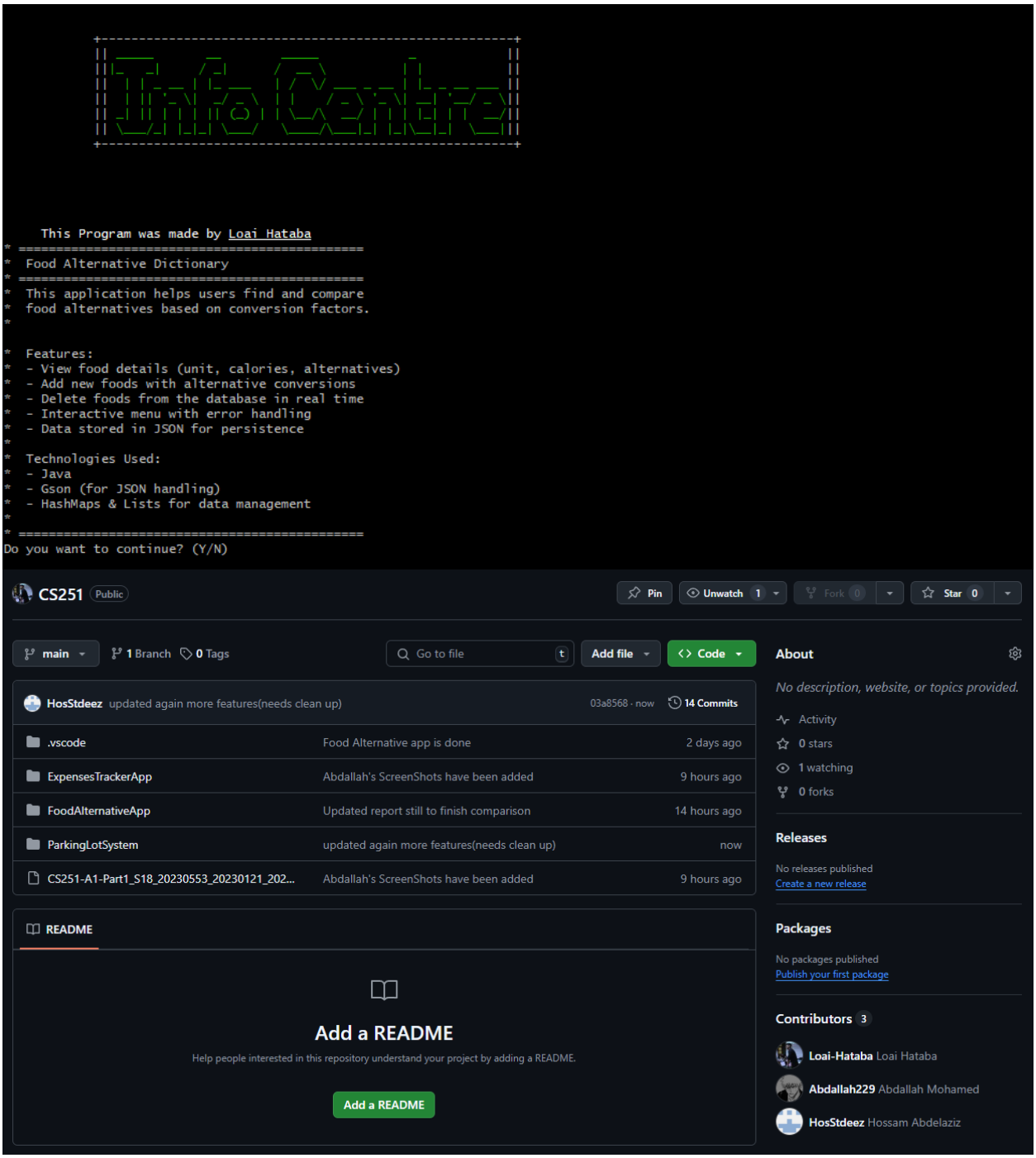
Welcome choose an option:

1)Find Alternative Food
2)Add new Food
3)Delete Food
4)Info
5)Exit

Choice: 3

Delete Food:
Food Name: ma7shy wara2 3enab
ma7shy wara2 3enab was deleted successfully!

Do you want to continue? (Y/N)



Video Link:

https://drive.google.com/file/d/17hZjf0I-YBnwHZ371pdMD7uhwbIYaQP3/view?usp=sharing

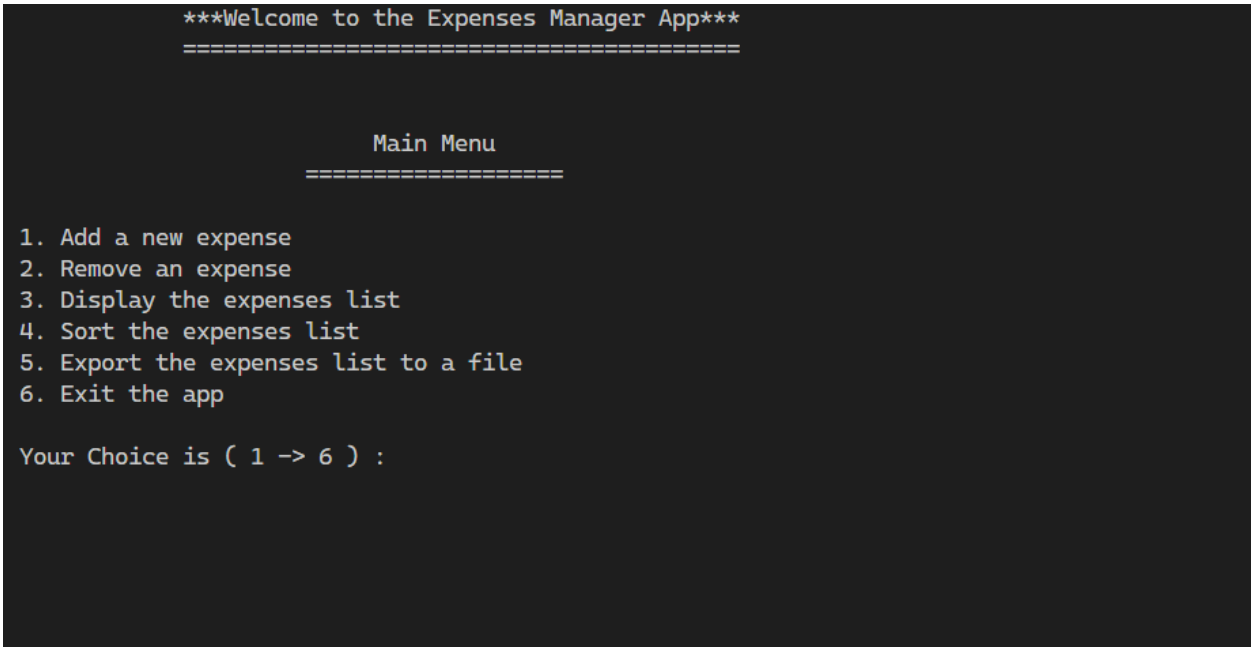
Budget Tracker (App 2 Abdullah):

Main Function:

```
   System.out.println(
"            ***Welcome to the Expenses Manager App***");
    System.out.println(
"            ========================================\n");
    // Create an instance of the expenses list
    final ExpensesList myExpenses = new ExpensesList();
    //The app menu :
    while (true) {
        System.out.println("\n                          Main Menu  ");
        System.out.println("                     ==================\n");
        // Main menu options :
        // 1 Adding an expense :
        System.out.println("1. Add a new expense ");
        // 2 removing an expense :
        System.out.println("2. Remove an expense ");
        // 3 Display the expenses list :
        System.out.println("3. Display the expenses list ");
        // 4 Sort the expenses list :
        System.out.println("4. Sort the expenses list ");
        // 5 Export the expenses list to a file :
        System.out.println("5. Export the expenses list to a file ");
        // 6 Exit the app :
        System.out.println("6. Exit the app ");
        //read the user choice  :

        final int choice = validInput.getValidInt("\nYour Choice is ( 1 -> 6
            ) : ", "Error : Invalid Choice !!", 1, 6);
        switch (choice) {
            case 1 ->
                myExpenses.addExpense();
            case 2 ->
                myExpenses.removeExpense();
            case 3 ->
                myExpenses.displayExpenses();
            case 4 ->
                myExpenses.sortExpenses();
            case 5 ->
                myExpenses.exportExpenses();
            case 6 -> {
final int ch = validInput.getValidInt("Do you want to saving before closing
```

```
?\n1)Yes\n2)No ", "Error : Invalid Choice !!", 1, 2);
  if (ch == 1) {
       myExpenses.exportExpenses();
       }
  System.out.println("Terminating the program :(");
           return;
             }
    default ->
        throw new AssertionError();
        }
    }
```

Screenshots:

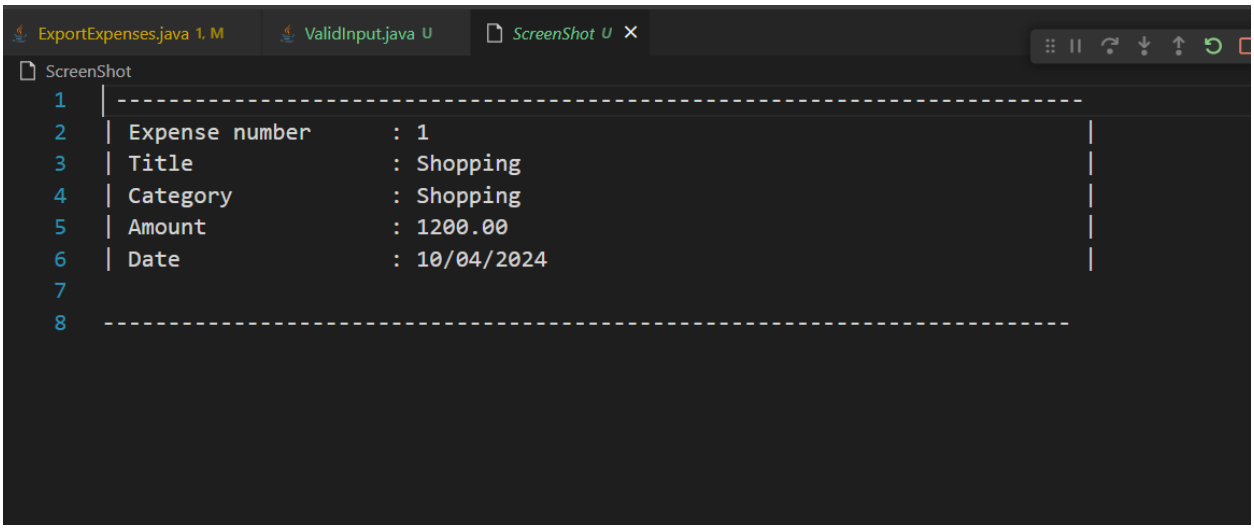


```
Enter the title of the expense: Cinema

Enter the amount of the expense : 80
Current Categories :
1- Food
2- Transport
3- Entertainment
4- Shopping
5- Bills
6- Others
Choose a category ( Write the index ) :3
Do you want to add a specific date or use the current date  ?
1 - current date
2 - specific date 1
The expense has been added successfully !

                        Main Menu
                    ====================

1. Add a new expense
2. Remove an expense
3. Display the expenses list
4. Sort the expenses list
5. Export the expenses list to a file
6. Exit the app

Your Choice is ( 1 -> 6 ) :
```

```
                        Main Menu
                    ====================

1. Add a new expense
2. Remove an expense
3. Display the expenses list
4. Sort the expenses list
5. Export the expenses list to a file
6. Exit the app

Your Choice is ( 1 -> 6 ) : 3
================================================================================
 ------------------------------------------------------------------------
| Expense number :    : 1                                                |
| Title               :                                                  |
| Category            : Entertainment                                    |
| Amount              : 80.00                                            |
| Date                : 27/02/2025                                       |
 ------------------------------------------------------------------------

================================================================================
```

```
                      Main Menu
                    ==================

1. Add a new expense
2. Remove an expense
3. Display the expenses list
4. Sort the expenses list
5. Export the expenses list to a file
6. Exit the app

Your Choice is ( 1 -> 6 ) : 2
              ( Removing an expense )
================================================================================
 ------------------------------------------------------------------------
| Expense number :    : 1                                                |
| Title               : Cinema                                           |
| Category            : Entertainment                                    |
| Amount              : 80.00                                            |
| Date                : 27/02/2025                                       |
 ------------------------------------------------------------------------

================================================================================
Choose an expense to remove ( Write the index ) : 1
The expense has been removed successfully !
```

```
                        Main Menu
                    ====================

 1. Add a new expense
 2. Remove an expense
 3. Display the expenses list
 4. Sort the expenses list
 5. Export the expenses list to a file
 6. Exit the app

 Your Choice is ( 1 -> 6 ) : 5
               ( Saving an expense )
 Do you want the file be in the current directory or a specific one ?
 1)Yes
 2)No1
 Enter the file name without any file format ScreenShot
 Attempting to create file at: ScreenShot
 Export successful: ScreenShot
```



Video Link:

https://drive.google.com/file/d/1_kMyuxn59d8yfKYCnpGyDzSnsc1SPGdx/view?usp=drive_link

Parking System (App 3 Hossam):

Main Function:

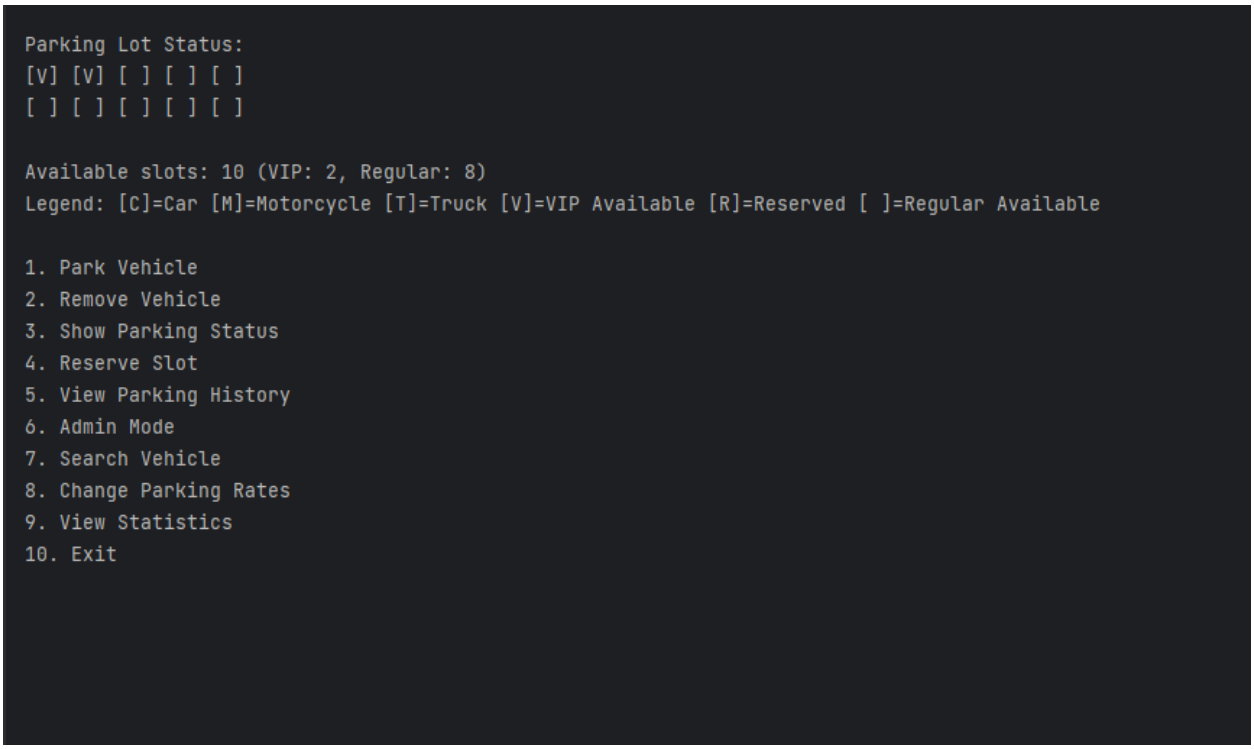
```
ParkingLot parkingLot = new ParkingLot(10);
Scanner scanner = new Scanner(System.in);

parkingLot.displayGrid();

// Track the last time we checked for expired reservations
long lastReservationCheck = System.currentTimeMillis();

while (true) {
  // Check for expired reservations every 5 seconds
  long currentTime = System.currentTimeMillis();
  if (currentTime - lastReservationCheck > 5000) { // 5 seconds
    parkingLot.checkReservations();
    lastReservationCheck = currentTime;
  }

  // Main menu options
  System.out.println("" +
      "\n1. Park Vehicle" +
      "\n2. Remove Vehicle" +
      "\n3. Show Parking Status" +
      "\n4. Reserve Slot" +
      "\n5. View Parking History" +
      "\n6. Admin Mode" +
      "\n7. Search Vehicle" +
      "\n8. Change Parking Rates" +
```

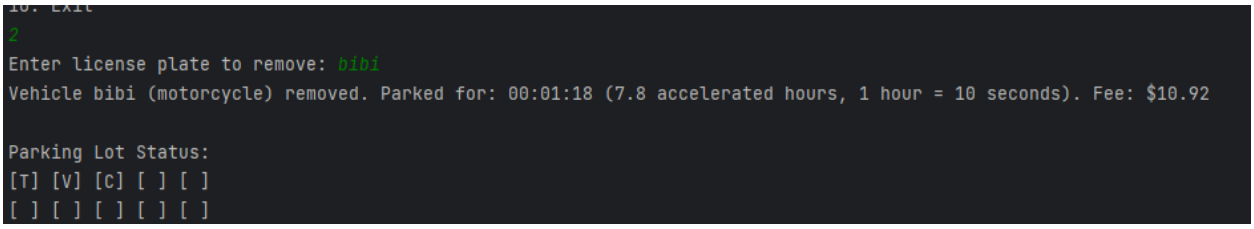
```
        "\n9. View Statistics" +
        "\n10. Exit");
int choice = scanner.nextInt();
scanner.nextLine();

// Check for expired reservations after any user action
parkingLot.checkReservations();

switch (choice) {
    case 1:
        System.out.print("Enter license plate: ");
        String plate = scanner.nextLine();
        System.out.print("VIP Slot? (yes/no): ");
        boolean isVIP = scanner.nextLine().equalsIgnoreCase("yes");
        System.out.print("Vehicle type (car/motorcycle/truck): ");
        String vehicleType = scanner.nextLine().toLowerCase();
        parkingLot.parkVehicle(plate, isVIP, vehicleType);
        parkingLot.displayGrid();
        break;
    case 2:
        System.out.print("Enter license plate to remove: ");
        plate = scanner.nextLine();
        parkingLot.removeVehicle(plate);
        parkingLot.displayGrid();
        break;
    case 3:
        parkingLot.displayGrid();
        break;
    case 4:
```

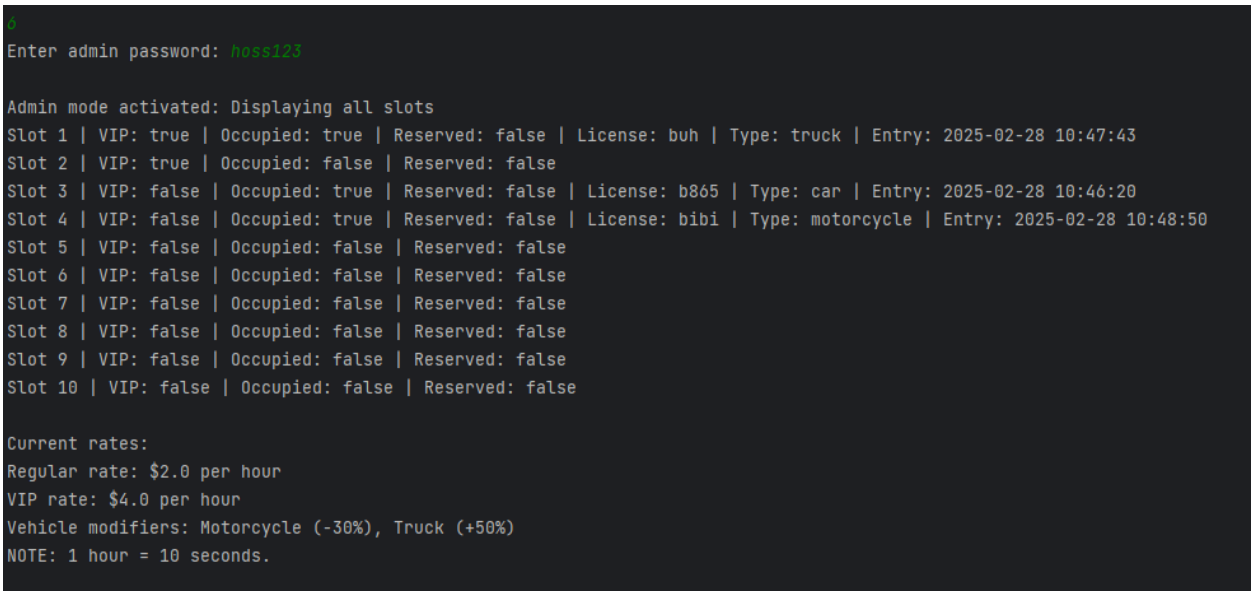
```
            System.out.print("Enter slot number to reserve: ");
            int slotNum = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Enter reservation duration (hours): ");
            int hours = scanner.nextInt();
            scanner.nextLine();
            parkingLot.reserveSlot(slotNum, hours);
            break;
        case 5:
            parkingLot.displayParkingHistory();
            break;
        case 6:
            // Simple password protection for admin mode
            System.out.print("Enter admin password: ");
            String password = scanner.nextLine();
            if (password.equals("hoss123")) {
                parkingLot.adminMode();
            } else {
                System.out.println("Incorrect password!");
            }
            break;
        case 7:
            System.out.print("Enter license plate to search: ");
            plate = scanner.nextLine();
            parkingLot.searchVehicle(plate);
            break;
        case 8:
            // Password protection for changing rates
            System.out.print("Enter admin password: ");
```

```
            password = scanner.nextLine();
            if (password.equals("hoss123")) {
                System.out.print("Enter new regular rate: ");
                double regularRate = scanner.nextDouble();
                System.out.print("Enter new VIP rate: ");
                double vipRate = scanner.nextDouble();
                scanner.nextLine();
                ParkingFeeCalc.updateRates(regularRate, vipRate);
                System.out.println("Rates updated successfully!");
            } else {
                System.out.println("Incorrect password!");
            }
            break;
        case 9:
            parkingLot.displayStatistics();
            break;
        case 10:
            System.out.println("Exiting...");
            return;
        default:
            System.out.println("Invalid option!");
    }
}
```

Screenshots:







```
Parking Lot Status:
[V] [V] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ]

Available slots: 10 (VIP: 2, Regular: 8)
Legend: [C]=Car [M]=Motorcycle [T]=Truck [V]=VIP Available [R]=Reserved [ ]=Regular Available

1. Park Vehicle
2. Remove Vehicle
3. Show Parking Status
4. Reserve Slot
5. View Parking History
6. Admin Mode
7. Search Vehicle
8. Change Parking Rates
9. View Statistics
10. Exit
1
Enter license plate: b865
VIP Slot? (yes/no): no
Vehicle type (car/motorcycle/truck): car
Vehicle parked in slot 3

Parking Lot Status:
[V] [V] [C] [ ] [ ]
[ ] [ ] [ ] [ ] [ ]
```

```
Enter license plate to search: buh

Vehicle found:
License plate: buh
Vehicle type: truck
Parked in slot: 1 (VIP)
Entry time: 2025-02-28 10:47:43
Parked for: 00:03:31 (21.1 accelerated hours, 1 hour = 10 seconds)
Current fee: $126.60
```

```
9. View Statistics
10. Exit
9

Parking Lot Statistics:
Total vehicles parked: 3
Total revenue: $222.92

Vehicles by type:
Cars: 1
Motorcycles: 1
Trucks: 1

Current status:
Occupied slots: 0 (0.0%)
Reserved slots: 0
Available slots: 10
```

Video Link:

Link:

https://drive.google.com/file/d/1E3eVMdre3xwt6Xg1BuPAEF-Zgqf1_IS_/view?usp=sharing

LCNC Analysis: (AppGyver vs Glide)

1. Introduction

Low-code and no-code development platforms have gained popularity for enabling non-developers and businesses to create applications with minimal coding.

Among these platforms, **AppGyver** and **Glide** stand out as powerful tools for building apps efficiently. This report analyzes AppGyver's and Glide's usability, benefits, and system quality while comparing them to determine their strengths and suitability for different use cases.

---

2. Evaluation of AppGyver

Overview

AppGyver is a professional-grade no-code development platform that allows users to create web and mobile applications without writing traditional code. It is particularly known for its flexibility and extensive customization options.

Usability and Features

- **Drag-and-Drop Interface**: Offers an intuitive builder for designing app layouts and functionalities.
- **Extensive Components**: Provides pre-built UI elements and logic modules.
- **Data Integration**: Supports REST APIs, databases, and third-party services.
- **Multi-Platform Deployment**: Applications can be deployed on web, iOS, and Android.
- **Logic and Automation**: Allows users to define workflows and dynamic logic visually.

Benefits and System Quality

- **Customization**: Unlike many no-code tools, AppGyver enables deep customization, making it ideal for complex applications.
- **Performance**: Apps built with AppGyver are optimized for high performance, especially on mobile devices.

- **Scalability**: Supports scalable applications, making it suitable for startups and enterprises.
- **Security**: Provides robust authentication and data security features.

Impact on Developers' Roles

While no-code platforms like AppGyver simplify application development, they are unlikely to replace traditional developers entirely. Instead, they serve as **enhancement tools** that allow developers to prototype faster and focus on more complex backend logic. Additionally, organizations can leverage no-code platforms for internal tools without needing a dedicated development team.

---

3. Evaluation of Glide

Overview

Glide is a no-code development platform that specializes in creating simple, data-driven applications using Google Sheets as a backend. It is widely used for lightweight business applications, internal tools, and prototypes.

Usability and Features

- **Google Sheets Integration**: Data is dynamically synced with Google Sheets, making data management straightforward.
- **Pre-Built Templates**: Provides templates to accelerate app creation.
- **Mobile-First Design**: Optimized for mobile applications with responsive design.
- **Drag-and-Drop Builder**: Users can easily add and arrange elements without coding.
- **Limited Logic & Automation**: Basic workflows can be set up, but advanced logic is limited.

Benefits and System Quality

- **Ease of Use**: Designed for non-technical users, making app creation accessible to a wide audience.
- **Speed of Development**: Apps can be created and deployed in minutes with minimal effort.
- **Cloud-Based**: Eliminates the need for complex hosting or deployment.

- **Data Management**: Real-time updates with Google Sheets ensure seamless synchronization.

Impact on Developers' Roles

Glide significantly lowers the barrier for app creation, allowing businesses to create internal tools without needing a development team. However, its **limited customization and scalability** mean that traditional developers are still essential for building feature-rich, enterprise-level applications. Glide is best suited for rapid prototyping and simple, data-driven applications rather than complex business solutions.

---

4. Comparison of AppGyver and Glide

Feature	AppGyver	Glide
Ease of Use	Moderate learning curve	Very beginner-friendly
Customization	High (supports complex logic and UI design)	Limited (focuses on simplicity)
Scalability	Suitable for enterprise applications	Best for small projects and internal tools
Pricing	Free for solo developers; enterprise pricing available	Free tier with premium plans for business features
Integrations	REST API, third-party services, external databases	Google Sheets, Airtable, Zapier, and limited API support

---

5. Sample To-Do List App

As part of this comparison, we created a **To-Do List App** using Glide. The application enables users to:

- Add new tasks with descriptions.
- Mark tasks as completed.
- Store and retrieve data using Google Sheets.
- Access the app from both web and mobile devices.

Glide's simplicity allowed for quick development, making it an excellent choice for basic applications.

Video link:

https://drive.google.com/file/d/1vqT3N6ripz5jJ3y9BYiqkThsBJVXNKQ7/view?usp=drive_link

App link:

https://habit-tracker-app-ills.glide.page

---

6. Conclusion

AppGyver and Glide both offer unique advantages in the no-code development space. Glide is ideal for quick prototyping and simple applications, while AppGyver provides greater flexibility and customization for more complex projects.