****Java Developer Internship

Project Name: Expense Tracker And Binary To Decimal Converter

Name: Pawar Rajani Baliram

Intern ID: VN-JD-4W293

Email: [rajanipawar6229@gmail.com](mailto:rajanipawar6229@gmail.com)

Week 1 Task:

Goal: Develop a Java application that converts binary numbers (input as a string) into decimal numbers.

Structure:

-User Interface: Design a console-based or GUI interface using Java frameworks (e.g., Swing, JavaFX, or simple CLI for basic functionality).

-Expense Management: Implement features for adding, editing, and deleting expenses, and categorizing them.

-Reports: Generate expense reports, showing detailed breakdowns and summaries of expenses over time.

**1. Project Overview**  
Expense Tracker is a Java-based application designed to help users manage their expenses effectively. The application allows users to add, edit, delete, and view their expenses. Additionally, it can generate detailed reports summarizing expenses by category.

**2. Features**

* **Add Expense**: Users can input the category, amount, and date of the expense.
* **Edit Expense**: Users can modify the details (category, amount, date) of an existing expense.
* **Delete Expense**: Users can delete an expense from the system by providing its ID.
* **View Expenses**: Users can view all expenses stored in the system.
* **Generate Report**: The application can generate a report that summarizes expenses by category, displaying the total amount spent per category.

**3. Installation Instructions**

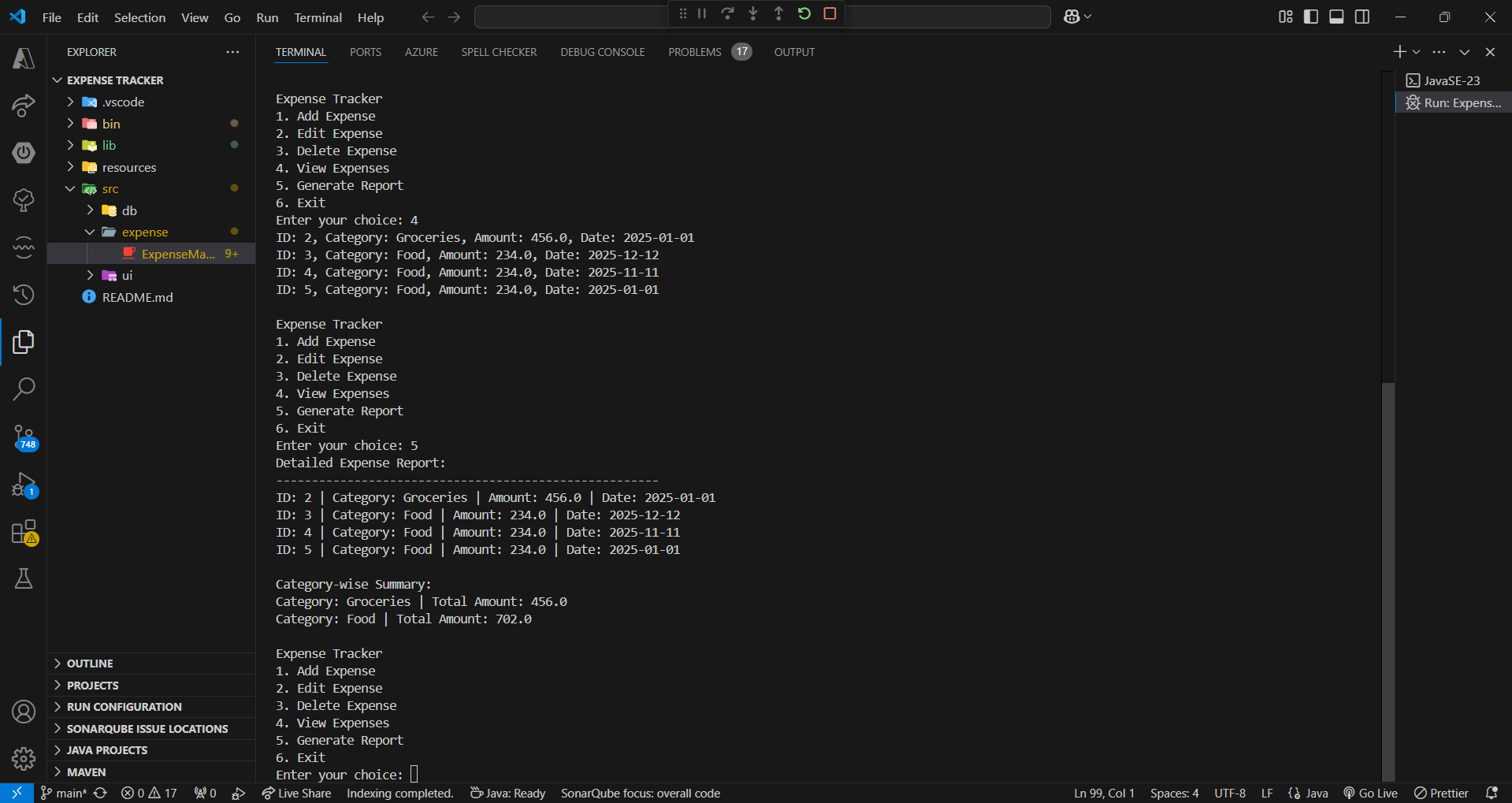
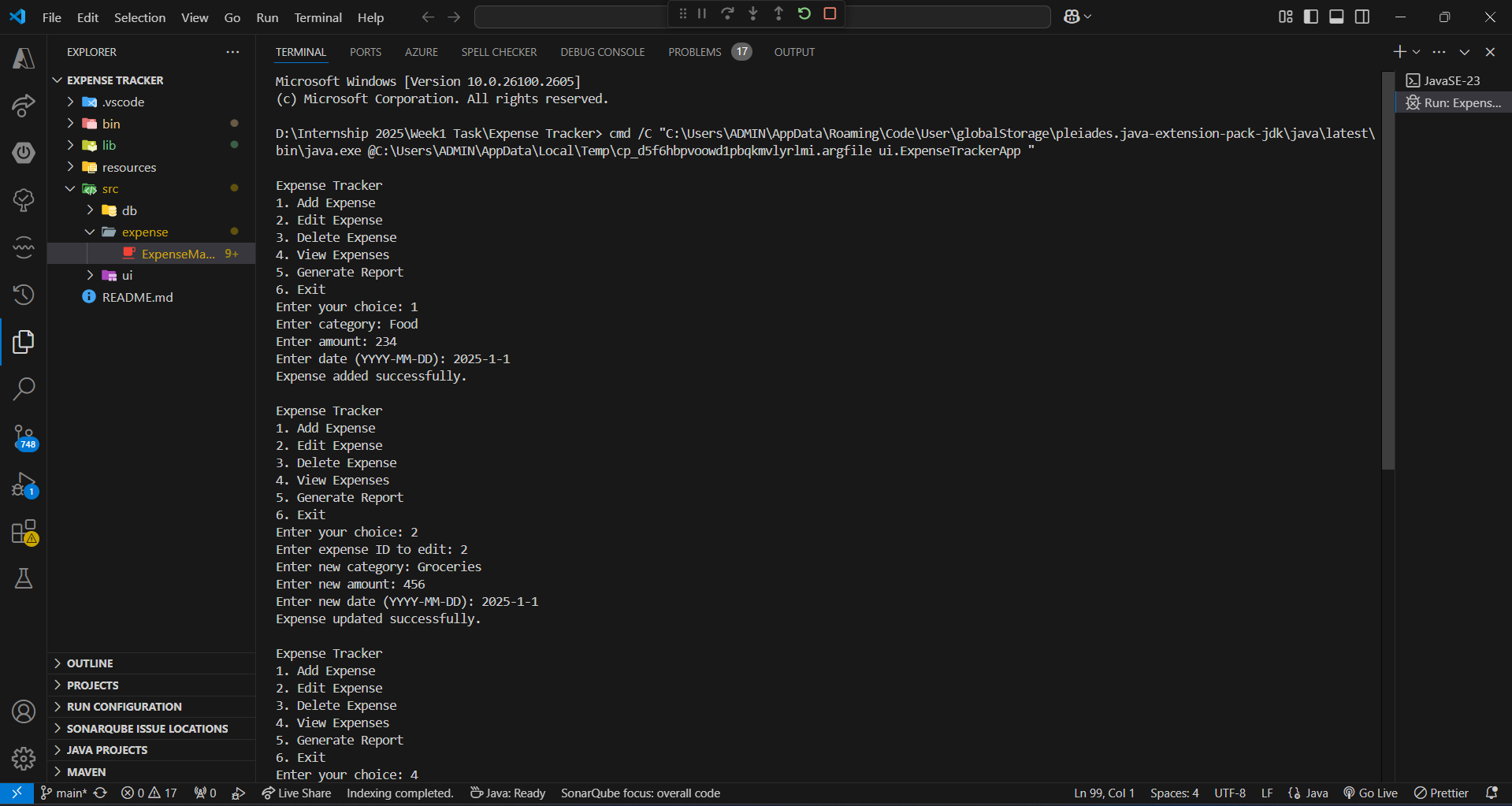
1. **Prerequisites**:
   * Java 8 or higher
   * MySQL database for storing expenses
2. **Database Setup**:
   * Create a MySQL table named expense with the following columns:
     + id INT PRIMARY KEY AUTO\_INCREMENT
     + category VARCHAR(255)
     + amount DOUBLE
     + date DATE
3. **Database Configuration**:
   * Update the DatabaseHelper class to provide the correct connection details for MySQL database.
4. **Running the Application**:
   * Compile and run the ExpenseTrackerApp class using an IDE like IntelliJ IDEA or Eclipse.

**4. Usage Instructions**  
Once the application is running:

* **Add Expense**: Select option 1 from the menu and enter the required details.
* **Edit Expense**: Select option 2, provide the expense ID, and update the necessary fields.
* **Delete Expense**: Select option 3 and input the expense ID you wish to delete.
* **View Expenses**: Select option 4 to view all stored expenses.
* **Generate Report**: Select option 5 to generate a report by category.

**5. Code Structure**

* **ExpenseManager Class**: Contains methods for adding, editing, deleting, viewing, and generating reports for expenses.
  + addExpense: Adds a new expense.
  + editExpense: Updates an existing expense.
  + deleteExpense: Deletes an expense by ID.
  + viewExpenses: Displays all expenses in the system.
  + generateReport: Generates and displays a report of expenses by category.

Output: 

**Binary to Decimal Converter**

**1. Project Overview**  
The Binary to Decimal Converter is a simple Java application that takes a binary number (input as a string) and converts it into its decimal equivalent. This tool is useful for understanding how binary numbers are represented in the decimal system.

**2. Features**

* **Binary to Decimal Conversion**: The application accepts a binary number as input and converts it to its decimal equivalent.

**3. Installation Instructions**

1. **Prerequisites**:
   * Java 8 or higher
2. **Running the Application**:
   * Compile and run the BinaryToDecimalConverter class using an IDE like IntelliJ IDEA or Eclipse.

**4. Usage Instructions**  
Once the application is running:

* Enter a binary number (e.g., 1010 or “1010” as String) when prompted.
* The application will output the decimal equivalent of the binary number.

**5. Code Structure**

* **BinaryToDecimalConverter Class**:
  + main: Accepts user input, validates the binary number, and converts it to decimal using the convertBinaryToDecimal method.
  + convertBinaryToDecimal: Converts a valid binary string into its decimal equivalent.

Output: 