**Software Needed**

**Java Development Kit (JDK)**

* **Purpose**: The main software required to write, compile, and run Java programs.
* **Version**: Java 8 or higher (preferably Java 11 for long-term support).
* **Installation**:
  + You can download it from Oracle JDK or use OpenJDK.

**Steps to Execute the Code**

1. **Install Java**:
   * Ensure Java JDK is installed by checking with the command: java -version.
   * If not installed, download and install it.
2. **Recommended Version**:
   * **Java 8**: Minimum for solid Swing support and modern features.
   * **Java 11 or higher**: Better performance, long-term support (LTS), and updated features.
3. **Save the Code**:
   * Save the code in a file named FinanceTrackerApp.java.
4. **Compile the Code**:
   * Open the terminal, navigate to the folder where the file is saved, and run:

javac FinanceTrackerApp.java

**Run the Application**:

* + Execute the program by running:

java FinanceTrackerApp

1. **Login Details**:
   * When the login window opens, use the following credentials:
     + **Username**: admin
     + **Password**: password123

**Libraries Used**

**javax.swing.\***

* **Purpose**: Helps to create graphical interfaces for the app.
* **Important Parts**:
  + JFrame: The main window of the app.
  + JTextArea: A box to show the list of transactions.
  + JButton: Buttons for user actions (e.g., Add Income, Add Expense).
  + JScrollPane: Adds a scrollbar to components like the transaction area.
  + JOptionPane: Shows pop-ups for messages or inputs.
  + JPasswordField: A field to securely enter passwords.

**java.awt.\***

* **Purpose**: Provides tools to arrange and style GUI components.
* **Important Parts**:
  + Color: Used to set the colors of buttons and panels.
  + GridBagLayout and GridBagConstraints: Help arrange components neatly in the login screen.
  + Insets: Adds space around components for better spacing.

**java.awt.event.\***

* **Purpose**: Handles user actions like button clicks.
* **Important Parts**:
  + ActionEvent: Represents a button click or similar action.
  + ActionListener: Allows defining what happens when a button is clicked.

**java.util.\***

* **Purpose**: Provides tools to manage collections of data.
* **Important Parts**:
  + ArrayList: Used to keep track of all the transactions (in TransactionManager).

**java.lang. (Automatically Included)**

* **Purpose**: Includes basic classes that every Java program can use without importing.
* **Important Parts**:
  + String: For text like transaction details and profile information.
  + Double: To work with numbers like income, expenses, and budget.