PROJECT 3 : ADVANCED JAVA PROJECT

- Project Name : DollarsBankJavaScriptAppV1
- Domain : Banking
- Application Type : Web App (Servlets/JSP)
- Architecture Used : MVC Architecture
- User Interface : Graphical User Interface

PROJECT 3 : Advanced JAVA PROJECT

<u>Objectives</u>
■

1. Use layered architecture (abstracted files under SOLID principles) to make an MVC banking application.

2. ***This a JSP project that expands, reworks the previous 2 projects so that the Front End is powered by JSP.

3. ***As always - use the Project Explorer Screenshots as a GUIDE to help brainstorm ideas, but you don't have to exactly implement in each file in the structure shown.

4. Create a user login system (no DB necessary, use in program memory).

5. Once logged in, have a user menu display in console.

6. Allow user to make :
   a. Deposit
   b. Withdrawl
   c. Funds Transfer
   d. 5 recent transaction history
   e. Display customer information
   f. Sign out

7. Apply business logic to handle illegal operations in :
   a. login system
   b. Transactions (withdrawls /depsosits, etc.)

8. See sample screenshots (next page).

9. After 1 week attempt - sample code will be released.

EXECUTION (SCREENSHOTS)

EXECUTION (SCREENSHOTS)
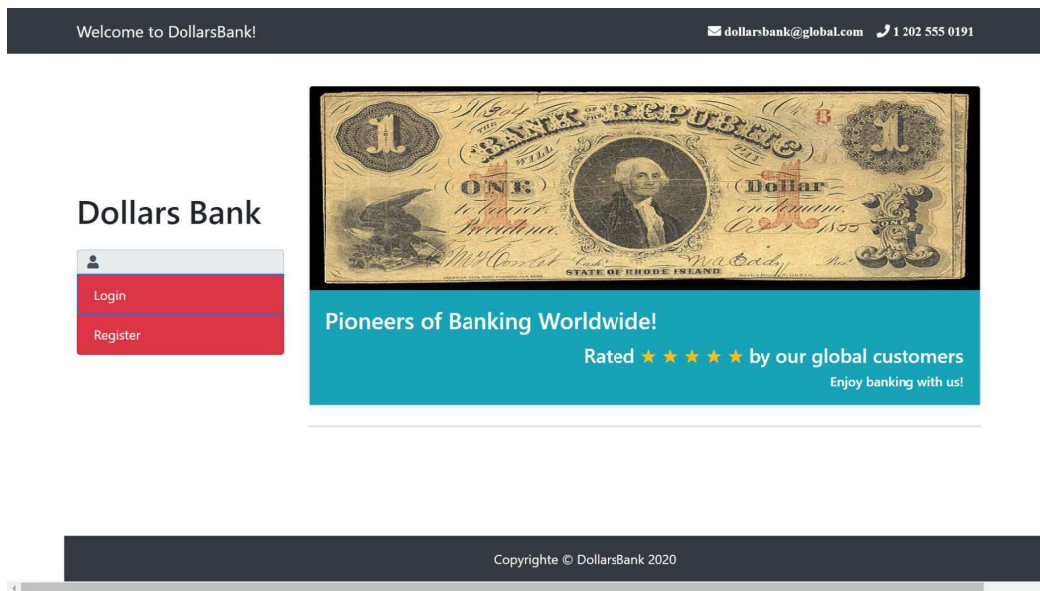
MVC ARCHITECTURE

```
v 🗐 DollarsBankWebAppV1
  >  ﷽ Deployment Descriptor: <web app>
  v  🗁 Java Resources
    v  🗁 src/main/java
      >  ⊞ com.dollarsbank.controller
      >  ⊞ com.dollarsbank.model
      >  ⊞ com.dollarsbank.repository.util
         ⊞ com.dollarsbank.service
      >  ⊞ com.dollarsbank.utility
      >  ⊞ com.dollarsbank.web
    >  🗁 src/test/java
    >  🗁 target/generated-sources/annotations
    >  🗁 target/generated-test-sources/test-annotations
    >  🗁 Libraries
  >  🗁 JavaScript Resources
  >  🗁 Deployed Resources
```

```
v  🗁 src
  v  🗁 main
    >  🗁 java
    v  🗁 webapp
      >  🗁 backup
      >  🗁 css
      >  🗁 dbscript
      >  🗁 images
         🗁 js
      >  🗁 vendor
      >  🗁 WEB-INF
         🗋 index.jsp
  >  🗁 test
  >  🗁 target
     🗋 pom.xml
```