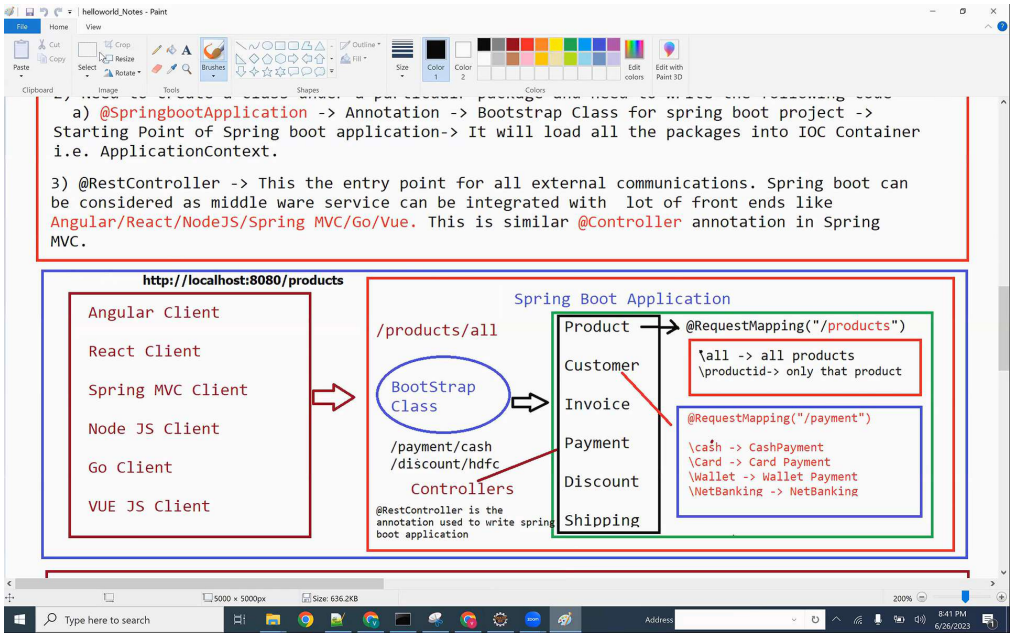
09

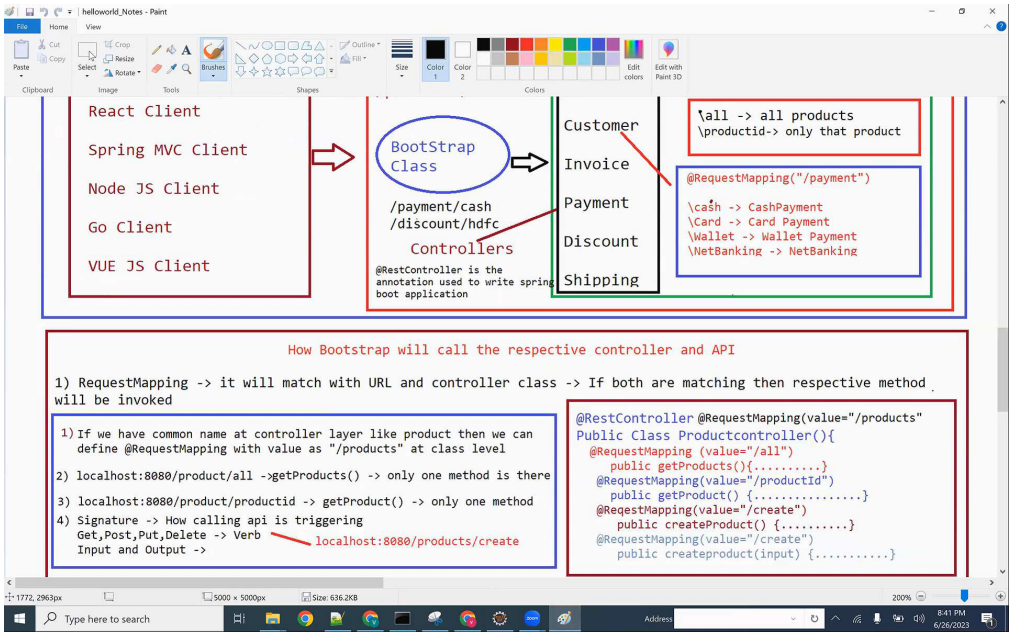
Wednesday, 25 June 2025 05:29

Nothing taught in this class.

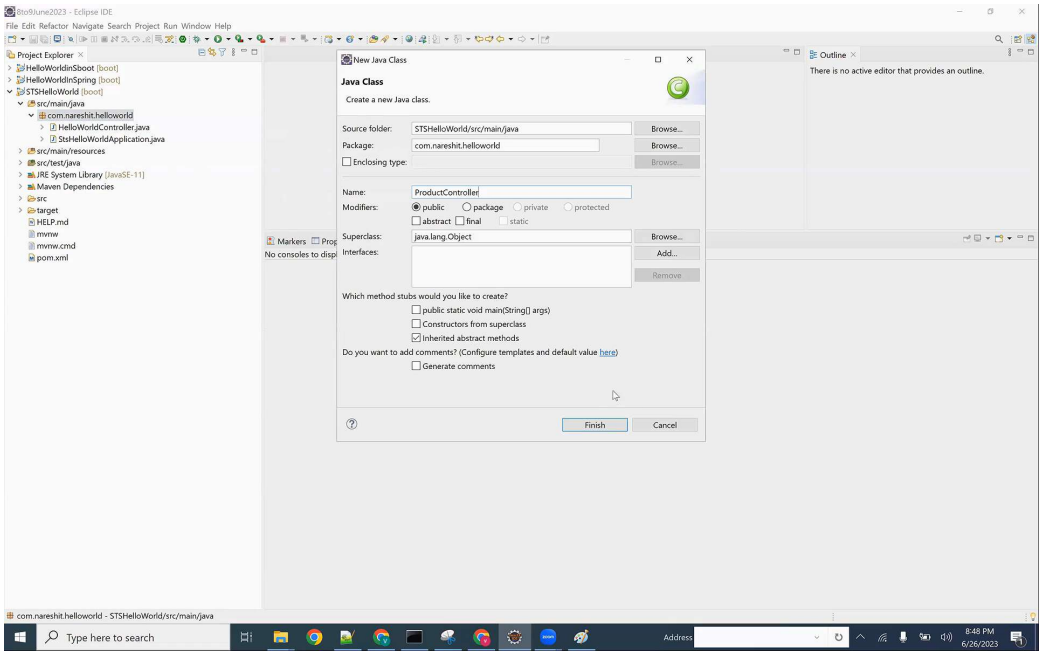
- told about how to download course material from google classroom and how to download project from the google classroom

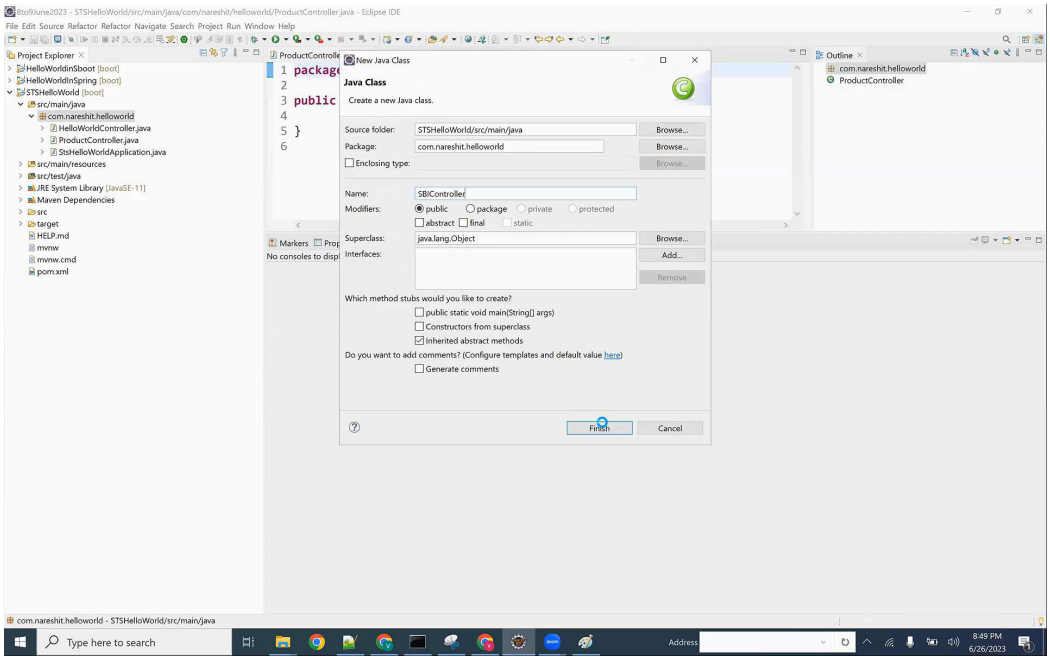
==========

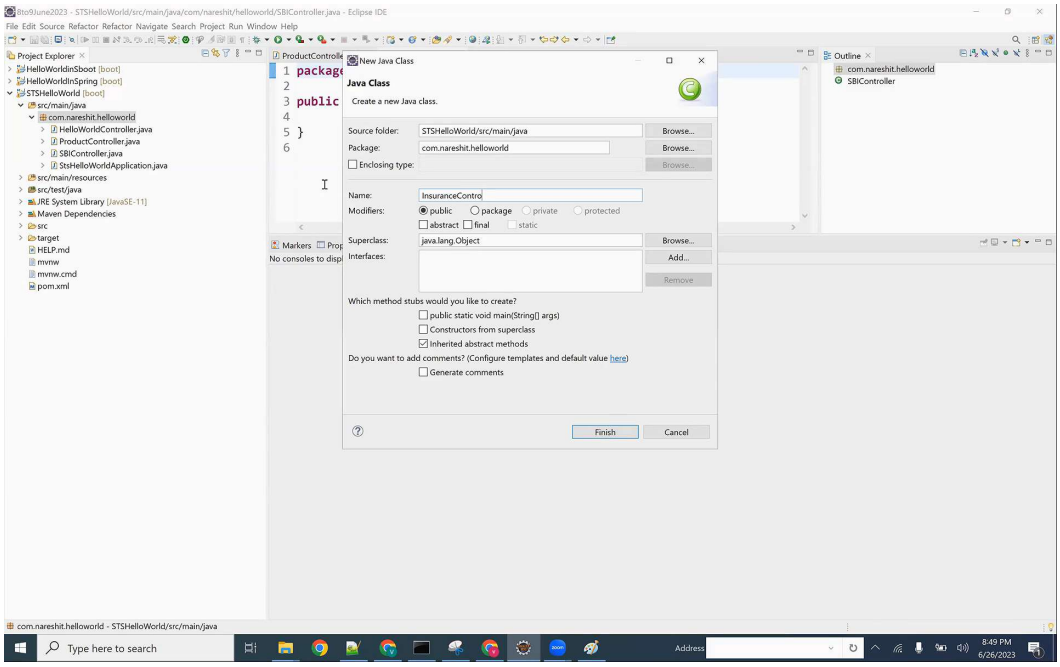


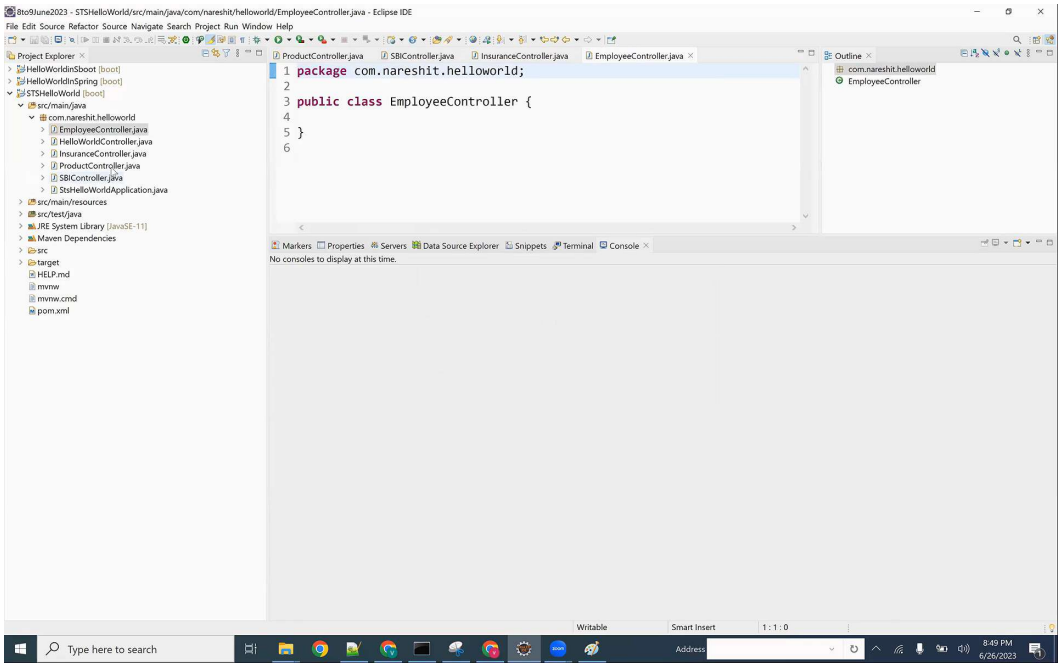


- REST + Controller
- Controller is the mediator between frontend and databaase
- controller is the entry point to access any functionality
- REST --> REST Webservices


8to9June2023 - Eclipse IDE
File Edit Refactor Navigate Search Project Run Window Help
Project Explorer
HelloWorldInSboot [boot]
HelloWorldInSpring [boot]
STSHelloWorld [boot]
src/main/java
com.nareshit.helloworld
HelloWorldController.java
StsHelloWorldApplication.java
src/main/resources
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
New Java Class
Java Class
Create a new Java class.
Source folder: STSHelloWorld/src/main/java
Package: com.nareshit.helloworld
Enclosing type:
Name: ProductController
Modifiers: public package private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish
Cancel
Outline
There is no active editor that provides an outline.
com.nareshit.helloworld - STSHelloWorld/src/main/java
Type here to search
8:48 PM
6/26/2023


8to9June2023 - STSHelloWorld/src/main/java/com/nareshit/helloworld/ProductController.java - Eclipse IDE
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Project Explorer
HelloWorldInSboot [boot]
HelloWorldInSpring [boot]
STSHelloWorld [boot]
src/main/java
com.nareshit.helloworld
HelloWorldController.java
ProductController.java
StsHelloWorldApplication.java
src/main/resources
src/test/java
JRE System Library [JavaSE-11]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml
1 package
2
3 public
4
5 }
6
New Java Class
Java Class
Create a new Java class.
Source folder: STSHelloWorld/src/main/java
Package: com.nareshit.helloworld
Enclosing type:
Name: SBIController
Modifiers: public package private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish
Cancel
Outline
com.nareshit.helloworld
ProductController
com.nareshit.helloworld - STSHelloWorld/src/main/java
Type here to search
8:49 PM
6/26/2023

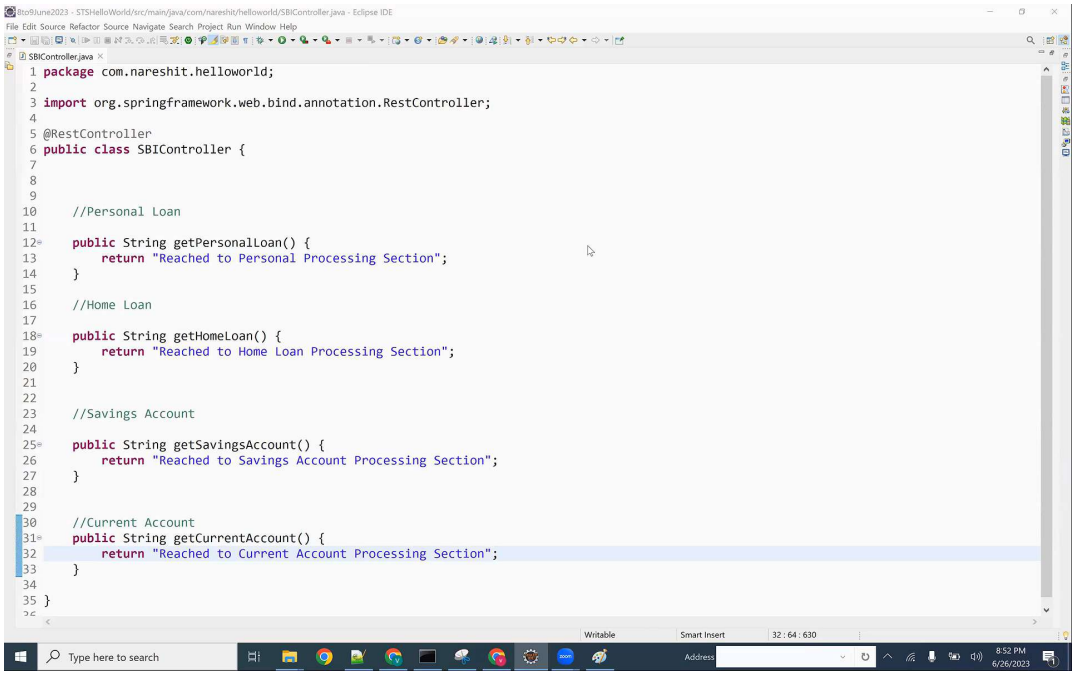




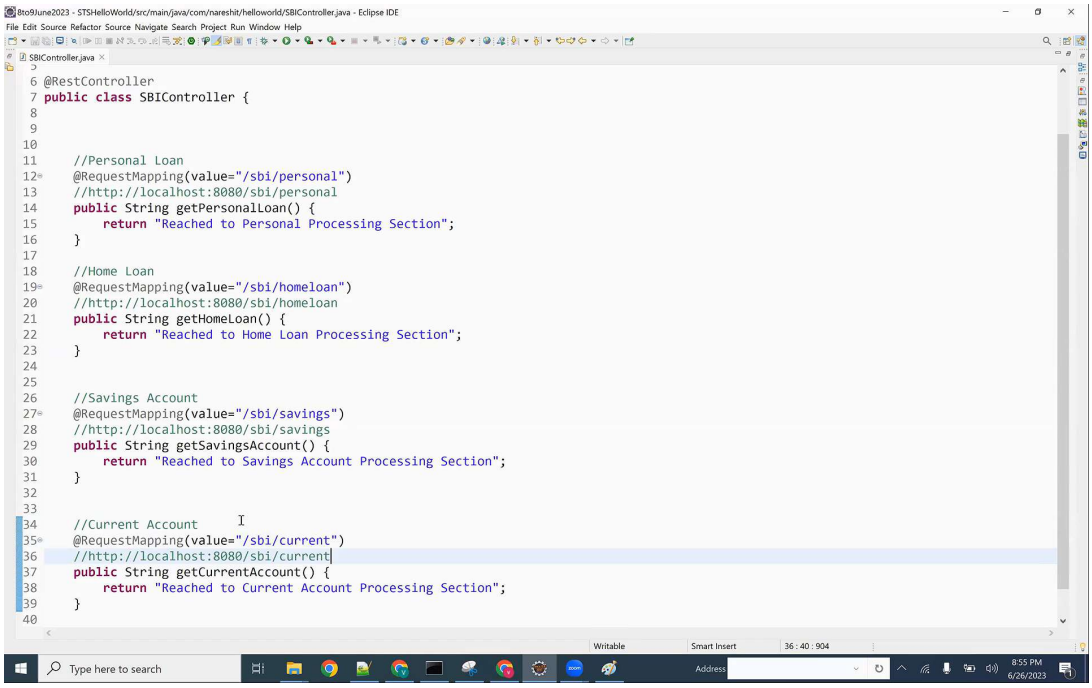
Writing @RestController and @RequestMapping

@RestContoller -->It is an entry point from outside to access any functionality. it is exposing the class to outside

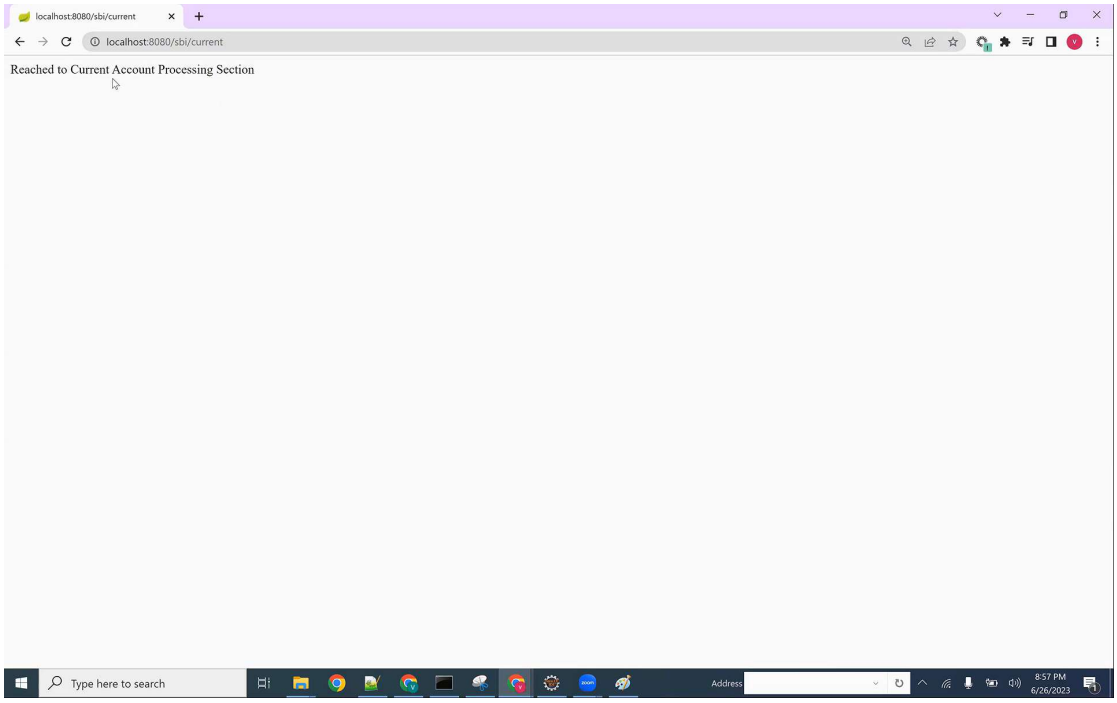
@RequestMapping --> it is going to compare the url and it is going to compare the method. which method is exactly matched, that method is called



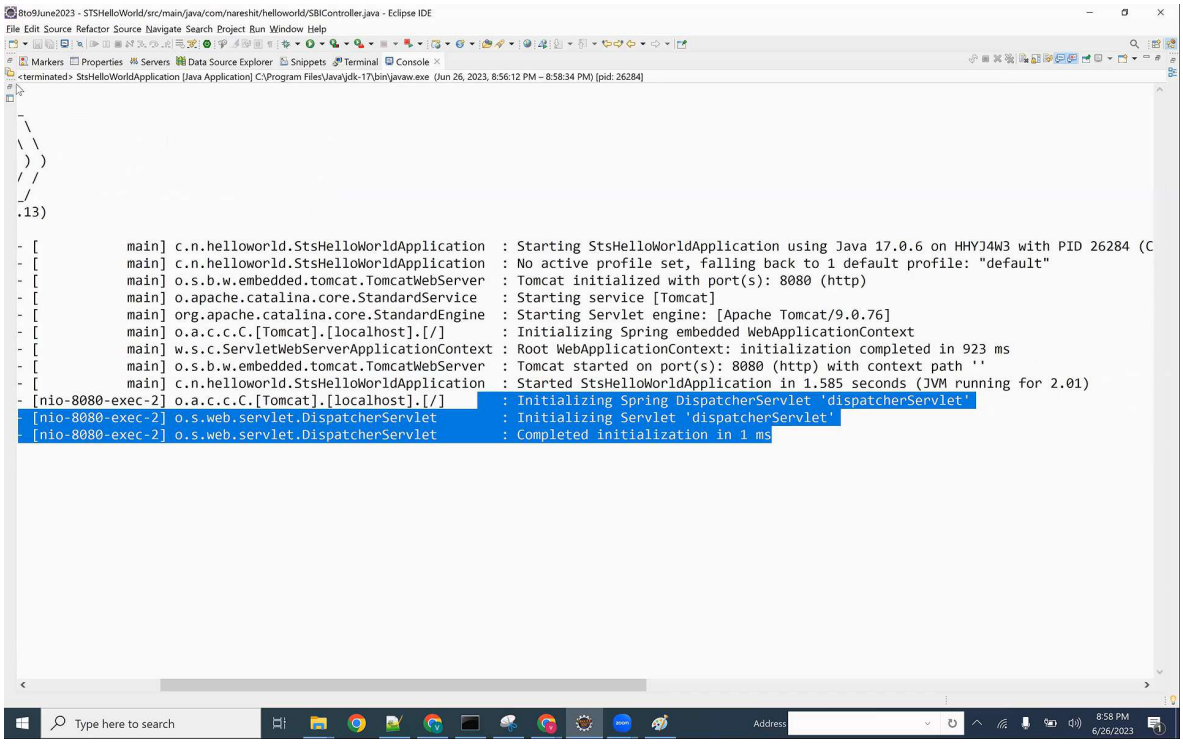
```
package com.nareshit.helloworld;

import org.springframework.web.bind.annotation.RestController;

@RestController
public class SBIController {



    //Personal Loan

    public String getPersonalLoan() {
        return "Reached to Personal Processing Section";
    }

    //Home Loan

    public String getHomeLoan() {
        return "Reached to Home Loan Processing Section";
    }


    //Savings Account

    public String getSavingsAccount() {
        return "Reached to Savings Account Processing Section";
    }


    //Current Account
    public String getCurrentAccount() {
        return "Reached to Current Account Processing Section";
    }

}
```



```
@RestController
public class SBIController {



    //Personal Loan
    @RequestMapping(value="/sbi/personal")
    //http://localhost:8080/sbi/personal
    public String getPersonalLoan() {
        return "Reached to Personal Processing Section";
    }

    //Home Loan
    @RequestMapping(value="/sbi/homeloan")
    //http://localhost:8080/sbi/homeloan
    public String getHomeLoan() {
        return "Reached to Home Loan Processing Section";
    }


    //Savings Account
    @RequestMapping(value="/sbi/savings")
    //http://localhost:8080/sbi/savings
    public String getSavingsAccount() {
        return "Reached to Savings Account Processing Section";
    }


    //Current Account
    @RequestMapping(value="/sbi/current")
    //http://localhost:8080/sbi/current
    public String getCurrentAccount() {
        return "Reached to Current Account Processing Section";
    }

```



==========



In tomorrow class we will see about dispatcherServlet.

==========

session completed.