

# Mini SpringBoot/Web MVC/SpringDataJPA Applications

## 1) Develop a SpringBoot/Web MVC application to calculate Personal Income Tax

**Step-I :** Create a jsp by the name `incomedetails.jsp`

Keep fields to accept Name, Annual Salary ,Gender & a Submit button by the label **"ShowTax"**

**Step-II:** Create a Controller Class `TaxController.java`

Retrieve all the `incomedetails` fields data .

Call the `TaxLogic` Class & pass above data from `TaxController`.

Accept return value of calculated tax amount from `TaxLogic` Class

Redirect Calculated Tax Amount to `taxdisplay.jsp`

**Step-III :** Create a service class `TaxService.java` Use `@Service` stereotype above the class .

Write a method `getTax(int annualSalary ,String gender)`

Compute the tax amount using the following tax structure.

Write Methods for below calculations & return calculated Tax Amount to the `TaxController`

If the Gender is Male, calculate the tax amount to be paid as follows:

- Salary upto 2.5 lacs - Tax is 0
- Salary from 2.5lacs upto 5 lacs - Tax is 5% of Annual Salary
- Salary from 5lacs upto 10 lacs - Tax is 10% of Annual Salary
- Salary from 10lacs & above - Tax is 20% of Annual Salary

If the Gender is Female, calculate the tax amount to be paid as follows:

- Salary upto 5 lacs - Tax is 0
- Salary from 5lacs upto 10 lacs - Tax is 5% of Annual Salary

- Salary from 10lacs & above - Tax is 10% of Annual Salary

**Step-IV :** Create taxdisplay.jsp

Display the Tax Amount with Name ,Annual Salary

Output eg: **Johns Annual Salary is 500000 Rs & Tax amount to be paid is 25000 Rs.**

## **2) Create a Products Display Application using Spring Boot/Spring Data JPA**

**Task1 :** Create productform.jsp

Keep fields like Id,Name , Price, Qty

Create ProductController.java

Create Product.java Bean class Mapped to Table Product

Write multiple methods to Save Product data in Product table

**Task-2:**

Step-1 : Create productsearch.jsp

Keep 1 text box Product Name & a submit button "Search"

On Clicking Search Button it should call ProductController.java

Step-3: Create productSearch method in the controller class

Retrieve Product Name from productsearch.jsp

Write logic to Search Product data from Product table

Set the productlist on SessionAttributes & ModelAndView Object

Redirect this to productdisplay.jsp

Step-4 Create productdisplay.jsp

Display productlist in HTML Table Rows & Cols .

## **3. Develop an Insurance Policy Application**

**Step-I :**

Create index.jsp page with following links :

[Login](#) [SignUp](#)

**Step-II :**

If the User clicks Login Link , display login.jsp with the fields

i) Login ID

ii) Password & a Login Submit Button

**Action :** On clicking Login Button User should be able to Login & redirected to dashboard.jsp

**Validation :** If the User has not Signed Up he cannot Login

**Step-III :**

If the User clicks SignUp Link , display signup.jsp with the fields

i) Login ID

ii) Password

iii) Name

iv) Address

v) Email

vi) Mobile No & a Signup Submit Button

**Action :** On clicking SignUp Button Users data should be stored in the User table. Then redirect to index page again.

**Validation :** Login ID & Password cannot be blank .Redirect again to Login Page if blank

**Step-IV :** Once Login is successful display dashboard.jsp with following page links :

[Buy Policy](#)

[View Portfolio](#)

[Delete Policy](#) [Logout](#)

**Step-V :**

Create Buy Policy Page with following fields :

i) Policy Name (DropDown with preloaded policy names)

ii) Insurance Tenure in Yrs. (Integer value only)

iii) Premium Amount

iv) Sum Insured (Only integers)

v) Buy Submit Button

Above data should be added to Policy Table . Then redirect to dashboard.jsp again .

**Step-VI:**

On clicking [View Portfolio](#) link , all the policies of the User should be displayed on portfolio.jsp . Display should be in HTML Rows & Cols .

**Step-VII:**

Create Delete Policy Page with following fields :

i) Policy Name (DropDown with preloaded policy names)

ii) Delete Submit Button

Then redirect to dashboard.jsp again .

**Step-VIII:**

On clicking Logout link , User should be redirected to index Page