

MARCH 2023

M	T	W	T	F	S	S
			1	2	3	4
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

SATURDAY  
FEBRUARY 2023

18

WEEK 07  
049-316

# Spring MVC Architecture

- 9 Spring MVC (Model View Controller) is a Java Framework for building web applications.
- 10 It's a part of larger Spring Framework and follows the Model View Controller design.
- 11 pattern.

## Components

- \* Model :- Represents the data and business logic of the application.
- \* View :- Responsible for rendering the user interface (UI) and displaying data to the user.
- \* Controller :- Handles user input, interacts with model, and updates the view

SUNDAY

19

## Flow of MVC

- 1) User sends request to the controller.
- 2) Controller processes the request, interacts with model and retrieves data.
- 3) Controller passes the data to the view
- 4) View renders the UI & displays data to user.

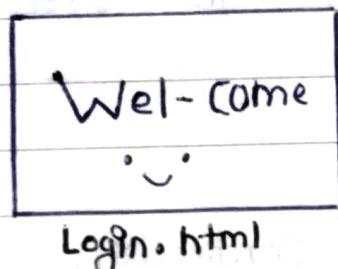
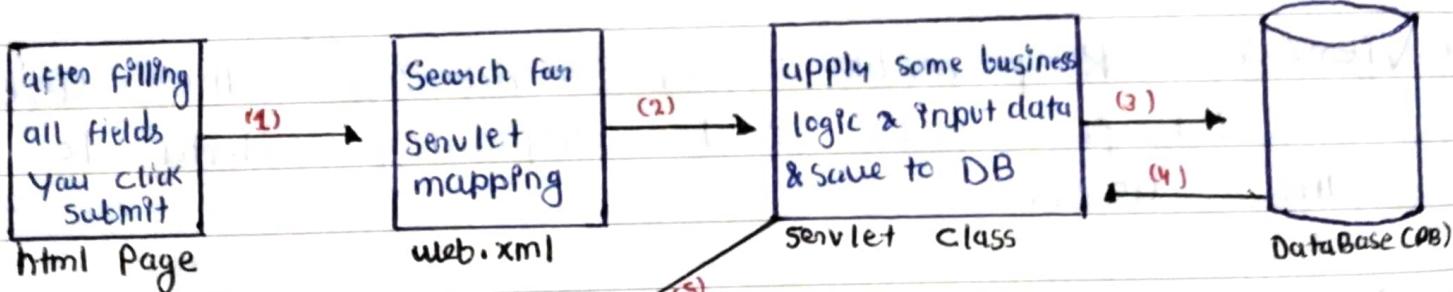
20

MONDAY  
2023 FEBRUARY

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

WEEK 08  
051-314To Create Web Application

- 8) 1) Create a maven project with architecture (maven-archetype-webapp)
- 9) 2). Download Apache tomcat Server
- 10) 3) Add Server to your project.  
 → right click on your project  
 → go to properties and choose Target Runtime  
 → add Server and close.

Basic Java Servlet Flow

after successful registration, servlet will send some response other page let it be login.html.

MARCH 2023

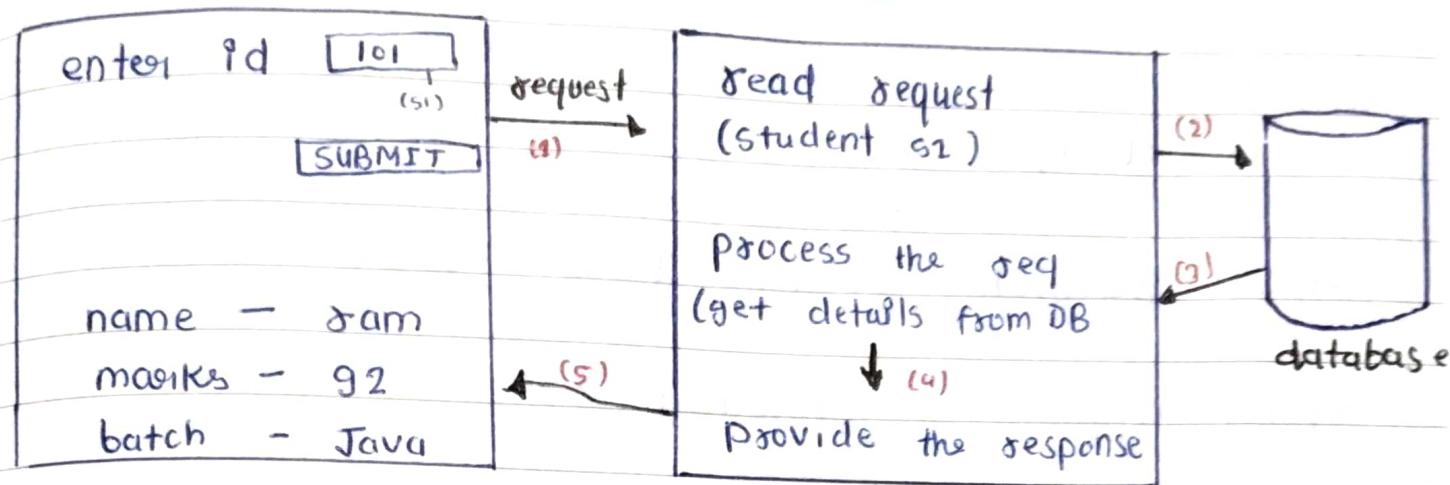
M	T	W	T	F	S	S
			1	2	3	4
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

TUESDAY  
FEBRUARY 2023

21

WEEK 08  
052-313

## Basic Servlet Project



## To Use Hibernate in Project

- 1) Need to add dependencies (mysql, hibernate, c3po)
- 2) Create a Entity class.
- 3) hibernate.cfg.xml configuration file for database.

### Hibernate Config File

```
<hibernate-configuration>
  <session-factory>
    <property name = "hibernate.connection.driver_class">
      com.cj.jdbc.mysql </property>
    <property name = "hibernate.connection.password"> root </property>
    <property name = "hibernate.connection.username"> root </property>
    <property name = "hibernate.connection.url"> jdbc:mysql://localhost:3306/bhanuclb </property>
    <mapping class = "com.entities.student" />
  </session-factory> </hibernate-configuration>
```

22

WEDNESDAY  
2023 FEBRUARY

WEEK 08  
053-312

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

2023 JANUARY

## HTML Page

8 <html>

9 <body>

10 <h2> Search Student here </h2>

11 <form action = "getStudent">

12 Enter Student NameId<input type="text" name="t1">

<input type = "submit" value = "submit" >

1 </form>

2 </body>

3 </html>

## Student Entity Class

4 @Entity

5 public class Student

6 {

7 @Id

8 private int id;

9 private String name;

10 private int marks;

11 → getter setter & Constructor ←

12 }

MARCH 2023

M	T	W	T	F	S	S
			1	2	3	4
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

THURSDAY  
FEBRUARY 2023

23

WEEK 08  
064-311

## Servlet Class

8 import com.entities.Student

9 public class Searchstu extends HttpServlet

{

10 public void doGet(HttpServletRequest req, HttpServletResponse res) throws Exception

11 {

12 PrintWriter out = response.getWriter();

13 String s1 = req.getParameter("t1");

14 int id = Integer.parseInt(s1);

### // hibernate Config.

15 Configuration config = new Configuration().configure();

16 Session session = config.buildSessionFactory().openSession();

17 Student s1 = session.get(Student.class, id);

18 out.println("Name = " + s1.getName());

19 out.println("Marks = " + s1.getMarks());

20 session.close();

}

}

Isme hamne MVC ka use nhi kia hai agar  
code ek hi jgh kia hai.

24

FRIDAY

2023 FEBRUARY

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

WEEK 08  
055-310Servlet

- 8 (1) Read the request
- 9 (2) Process the request (DB operations)
- 10 (3) Apply some business logic.
- 11 (4) Provide Response.

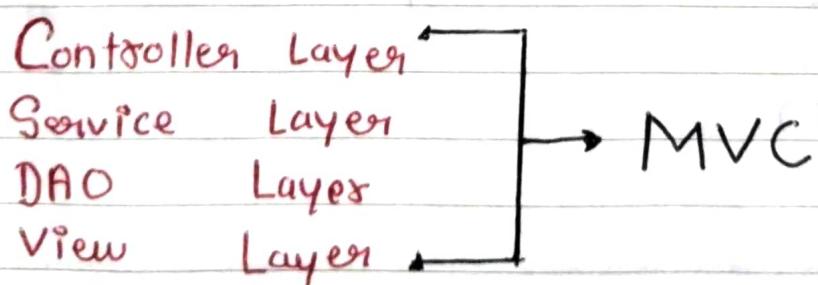
MVC

Model View Controller

In mvc architecture we used to separate each part, concern.

In servlet all the task like reading the request integration with database , applying business logic and providing response, All task done in same class It decrease the readability and increase the complexity of code.

To solve this problem we use MVC architecture.



MARCH 2023

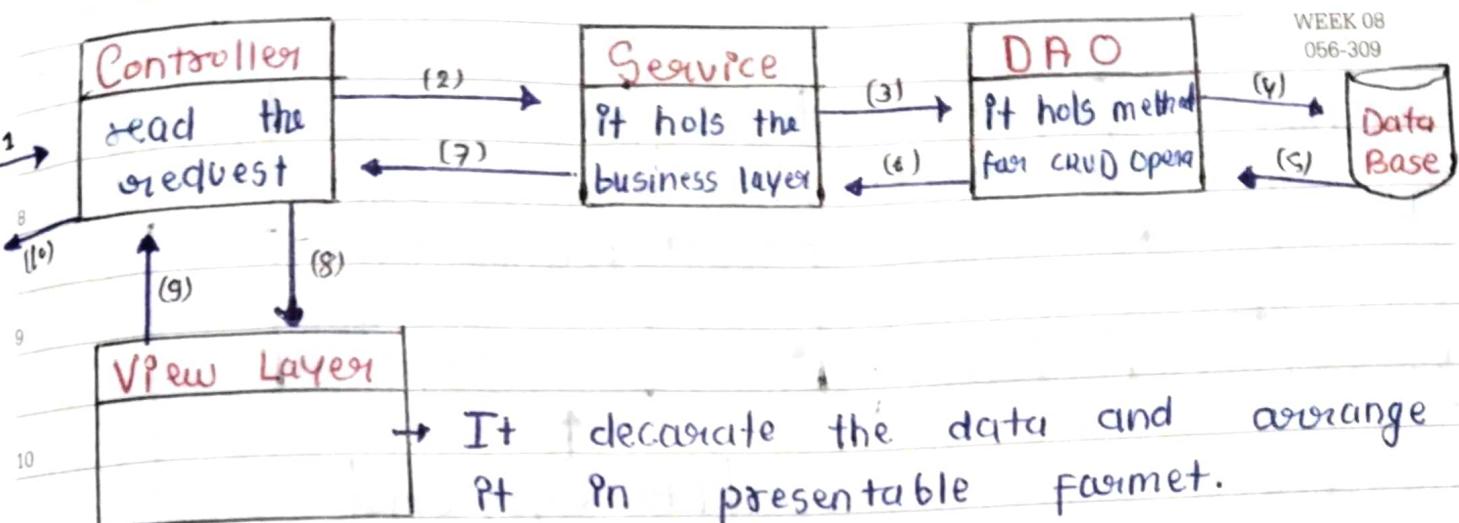
M	T	W	T	F	S	S
			1	2	3	4
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

SATURDAY  
FEBRUARY 2023

25

WEEK 08  
056-309

Data  
Base



## Model

Model is a Java object that contains the data which we need to transfer from one layer to another. A simple bean or entity class can be used as model.

## MVC Uses

MVC architecture is used for separation of various layers, business logic is separated from DB operation and Database operation is separated from view layer.

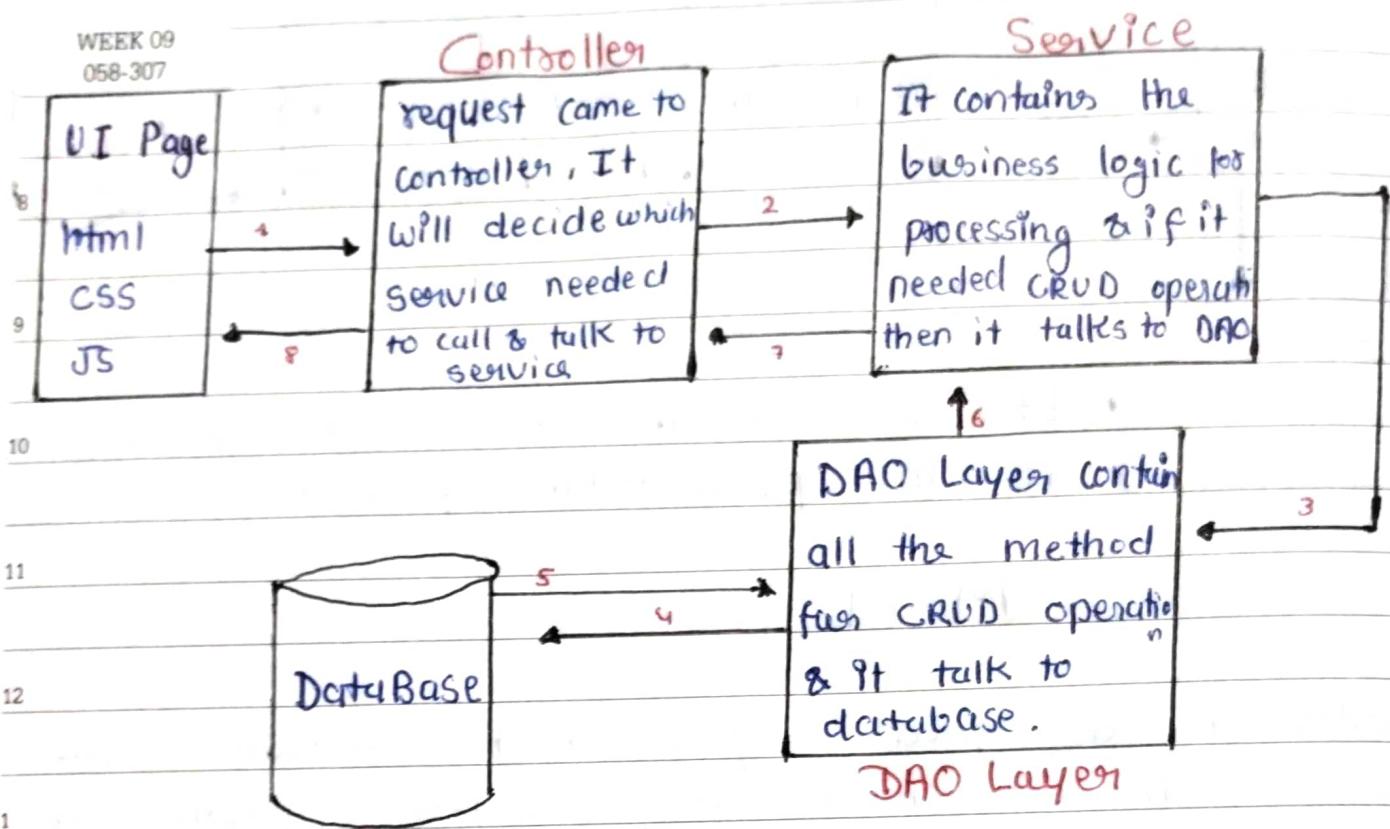
SUNDAY

26

27

MONDAY  
2023 FEBRUARY

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

WEEK 09  
058-307

## Simple MVC Project

### web.xml

&lt;web-app&gt;

&lt;servlet&gt;

&lt;servlet-name&gt; Student &lt;/servlet-name&gt;

&lt;servlet-class&gt; com.cb.servlet.SearchStudent &lt;/servlet-class&gt;

&lt;/servlet&gt;

&lt;servlet-mapping&gt;

&lt;servlet-name&gt; Student &lt;/servlet-name&gt;

&lt;url-pattern&gt; /Search &lt;/url-pattern&gt;

&lt;/servlet-mapping&gt;

&lt;servlet-name&gt;

&lt;/url-mapping&gt;

&lt;/web-app&gt;

MARCH 2023

M	T	W	T	F	S	S
			1	2	3	4
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

TUESDAY

FEBRUARY 2023

28

WEEK 09  
059-306

# Controller

## SearchStudent (Servlet)

public class SearchStudent extends HttpServlet

public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException

{ // read the request

int id = Integer.parseInt(req.getParameter("id"));

// forward Request to Service class

StudentService stserv = new StudentService();

Student student = stserv.findById(id);

// forward the response

req.setAttribute("student", student);

RequestDispatcher rd = req.getRequestDispatcher("details.jsp");

rd.forward(req, res);

}

}

1 WEDNESDAY  
2023 MARCH

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

WEEK 09  
060-305

## Service Layer

8 public class StudentService

9 private StudentDAO studentDAO;

10 public StudentService():

11 StudentDAO = new StudentDAO();

12 }

13 public Student findbyId (int id)

14 {

15 return StudentDAO.findById(id);

16 }

## DAO Layer

17 public class StudentDAO

18 {

19 private SessionFactory sf;

20 public StudentDAO()

21 {

22 Configuration con = new Configuration().configure();

23 sf = con.buildSessionFactory();

24 }

2

THURSDAY  
2023 MARCH

WEEK 09  
061-304

S	M	T	W	T	F	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

public Student findById (int id)

{

Session session = sf.openSession(); // Dao called  
Student st = session.get (Student.class, id); // getting data  
return st;

}

}

## Details.jsp

<%@ page isELIgnored = "false" %>

<html>

<head>

<title> Student Details </title>

</head>

<body>

<pre>

Student Name \$ {student.name}

Student Batch \$ {student.batch}

Student Marks \$ {student.marks}

</pre>

</body>

</html>

APRIL 2023

M	T	W	T	F	S	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

FRIDAY  
MARCH 2023

3

WEEK 09  
062-303

# Expression Language

8  
9     \$ (member) is called EL (Expression Lang.)

10     here EL is disable by JSP

11     \$ (member) is printed as a normal text

12     If you are using \$ as expression language, enable it by.

1     <%@page isELIgnored = "false" %>

2     we are saying not to ignore expression language.

## html Page

<html>

    <body>

        <form method = "get" action = "SearchStudent">

Enter Student Id <input type = "text" name = "t1"/>

    <input type = "submit" value = "Submit" />

    </form>

    </body>

    </html>

4

SATURDAY  
2023 MARCH

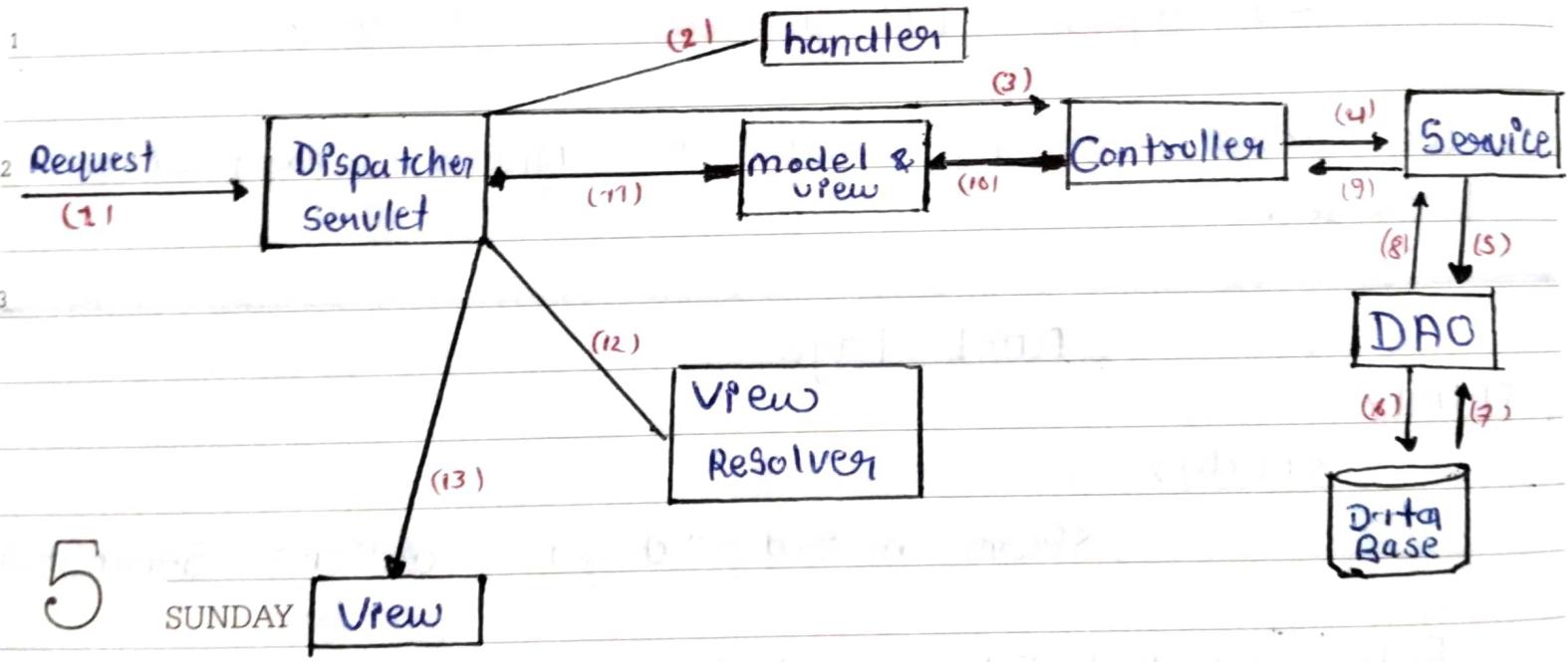
S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

WEEK 09  
063-302

# Spring Web MVC

A Spring MVC provides an elegant solution to use MVC in spring frame work by the help of Dispatcher Servlet

Here DispatcherServlet is a class that receives the incoming request and maps it to the right source such as Controllers, models and view.



5

SUNDAY

DispatcherServlet Request ko handler ke de dete hui and jo thi request aai thi use uske controller ke lejake corresponding mapping ke according method execute kt detu hui.

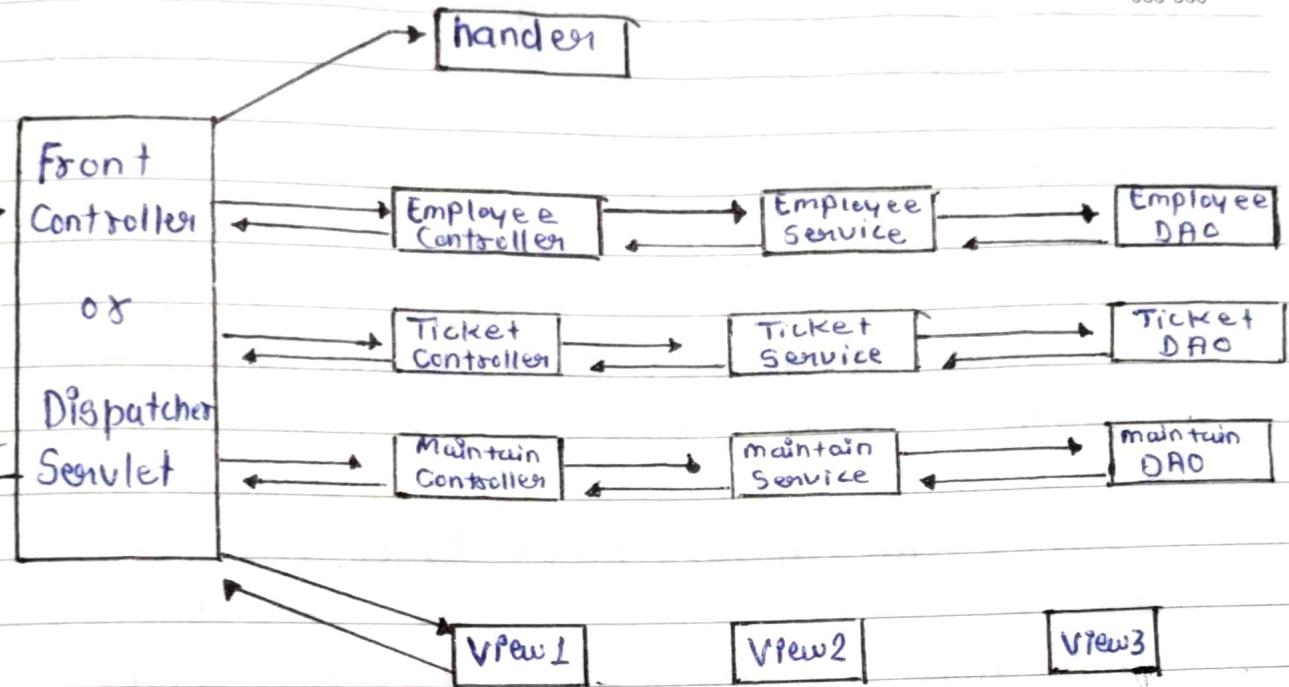
APRIL 2023

M	T	W	T	F	S	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

MONDAY  
MARCH 2023

6

WEEK 10  
065-300



For eg request is coming for (ticket) save DispatcherServlet will search for appropriate handler (Servlet) method and will transfer the request to that handler mapping. Handler consist all mapping for request.

## Flow of DispatcherServlet

- 1) Every Request goes to the DispatcherServlet (Front Controller)
- 2) Dispatcher Servlet scan for the handler mapping.

17

TUESDAY  
2023 MARCH

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

WEEK 10  
066-299

## 3) Handler Mapping

8

@Controller

9 class Ticket

10

@RequestMapping("/ticket/save")

11 public Ticket saveTicket()

12

System.out.println("Radhe Radhe");

13

14 //multiple method with different mapping

15

16 Your various Controller are having  
 17 different methods for handling different -  
 18 different request and each method should  
 19 have url-mapping (against which url, this  
 20 method should be called).

21 \*Note :- Home Mapping vali class ka  
 22 @Controller se mark karna compiler  
 23 -sayu hota hai taki DispatcherServlet  
 24 ko pta lag sake ki yahi controller  
 25 hui jiske ander se mapping methods  
 26 hai.

27 Abb Hamne Alg-Alg mapping ke liye alg  
 28 alg Servlet class nhi bnangi plegi. ek class  
 29 me se mapping ho Jayegi.

APRIL 2023

M	T	W	T	F	S	S
3	4	5	6	7	1	2
10	11	12	13	14	15	9
17	18	19	20	21	22	16
24	25	26	27	28	29	30

WEDNESDAY  
MARCH 2023

8

WEEK 10  
067-298

## Steps For Spring Web MVC

- 1) Create a project in maven with the archetype - webapp
- 2) Add Server to your project
- 3) Create Java folder inside src/main
- 4) Add required dependencies (bare minimum spring mvc)
  - // hibernate c3p0
  - // spring context
  - // spring core
- 5) Map all request to Front Controller aka Dispatcher Servlet (web.xml)
- 6) Create a configuration file for configure your beans, here we used xml based configuration (dispatcher-servlet.xml)
- Now, check your configuration file
- \* Create a controller
- \* Write a method with some handler-mapping (request mapping).

20

2023 MARCH

12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

WEEK 10  
068-297

# Spring Web MVC Code

8

## web.xml

9

```

<web-app>
    <servlet>
        <servlet-name> dispatcher </servlet-name>
        <servlet-class> org.springframework.web.servlet.DispatcherServlet
        </servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name> dispatcher </servlet-name>
        <url-pattern> / </url-pattern>
    </servlet-mapping>

</web-app>

```

4

## DispatcherServlet

```

<beans>
    <mvc:annotation-driven />
    <context:component-scan base-package = "com.db" />
</beans>

```

Note :- In this file we have to add the `.xml` context schema (`web.xml`) file ke pss locate hogi ye file.

APRIL 2023

M	T	W	T	F	S	S
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

FRIDAY  
MARCH 2023

10

WEEK 10  
069-296

## Java Class

```
8 package com.db;  
9  
10 @Controller  
11 public class TestController  
12 {  
13     @RequestMapping("/testing")  
14     public void test()  
15     {  
16         System.out.println("welcome ji");  
17     }  
18 }
```

```
19 @RequestMapping("/hello")  
20 public void say()  
21 {  
22     System.out.println("ram ram ji");  
23 }  
24 }
```

5 Ham Jab browser pr "/testing" likhenge to ye  
6 test() method call ho jayegi and output console  
7 pr aayega. Agar hamne browser pr "/hello" likha  
to say() method call ho jayegi.

Hame ab alg-alg mapping ke liye alg-alg  
servlet class nahi banani pdagi.

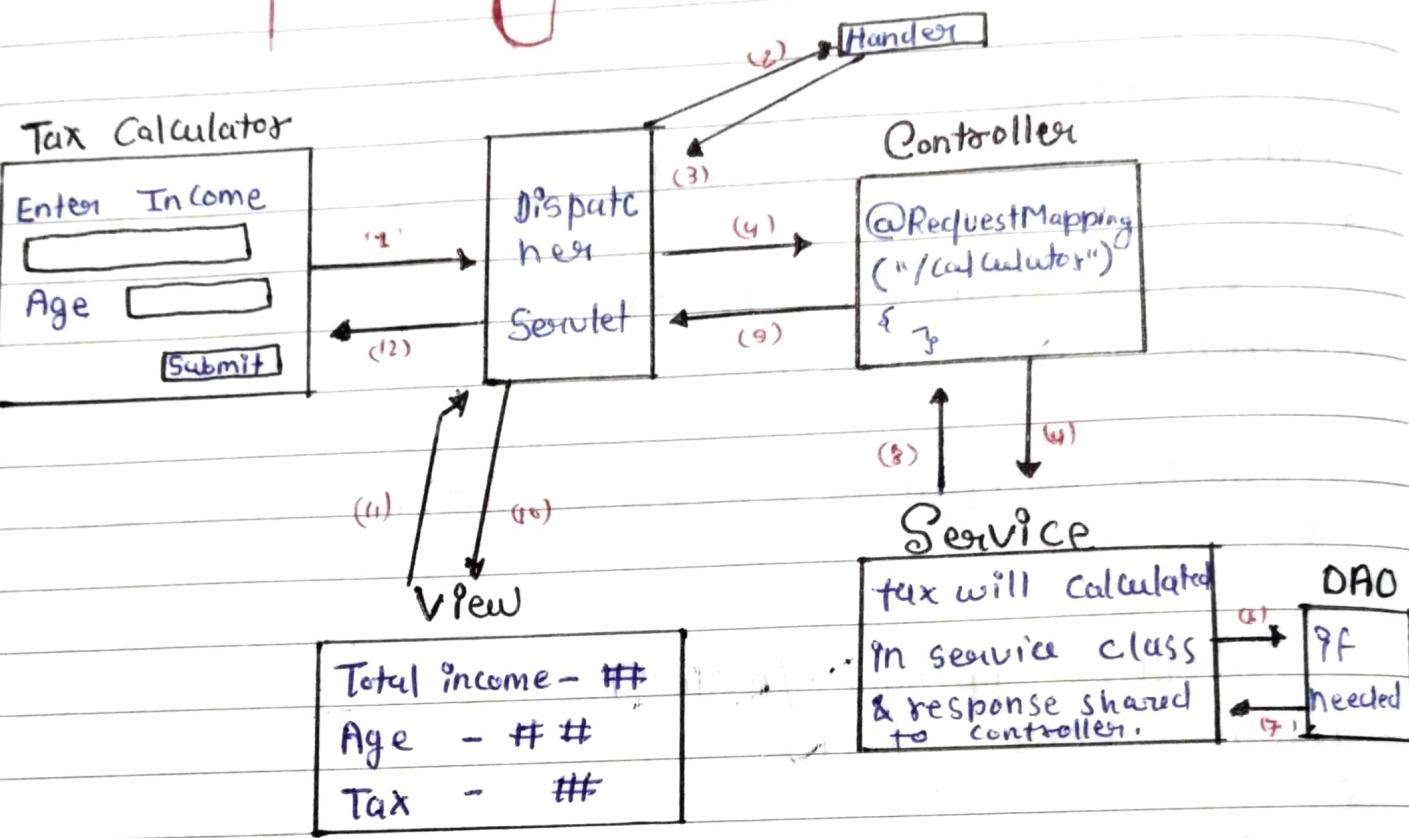
11

SATURDAY  
2023 MARCH

S	M	T	W	T	F	S
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

WEEK 10  
070-295

# Spring MVC Project



## Model Class

12

SUNDAY

package com.db.model;

public class TaxModel

{

private int income, age;

private int tax;

private int rebate, netTax;

→ getter Setters ←

J

3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

WEDNESDAY  
MARCH 2023

15

WEEK 11  
074-291

## Input.jsp

```
<html>
<body>
<form action = "compute">
<pre>
Enter Name → <input type = "text" name = "t1" />
Enter Age → <input type = "text" name = "t2" />
<input type = "submit" value = "calculate Tax" />
</pre>
</body>
</html>
```

## Controller

```
package com.db.controller
import com.db.service

@Controller
public class TaxController
{
    @Autowired
    TaxService taxService;
    @Autowired
    ModelAndView mv;
    @Autowired
    TaxModel taxModel;
```

16

THURSDAY  
2023 MARCH

2023 FEBRUARY

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

APRIL 2023

WEEK 11  
075-290

@RequestMapping (" /compute")

public ModelAndView computeTax (@RequestParam ("t1")  
int income, @RequestParam ("t2") int age)

taxModel.setIncome (income);  
taxModel.setAge (age);

// process

taxMode = taxService.computeTax (taxModel);

// Provide Response

mv.addObject ("taxModel", taxModel);

mv.setViewName ("result.jsp");

return mv;

3

3

## Service

package com.db.service;

public class TaxService

public TaxModel ComputeTax (TaxMode taxModel)

int income = taxModel.getIncome();

int age = taxModel.getAge();

int tax = 0;

int rebate = 0;

M	T	W	T	F	S	S
3	4	5	6	7	1	2
10	11	12	13	14	8	9
17	18	19	20	21	15	16
24	25	26	27	28	22	23
					29	30

FRIDAY  
MARCH 2023

17

WEEK 11  
076-289

if ( $\text{income} \geq 50000$ )

    tax =  $\text{income} * 20 / 100 ;$

else

    tax =  $\text{income} * 10 / 100 ;$

if ( $\text{age} \geq 60$ )

    rebate =  $\text{tax} * 2 / 100 ;$

}

int netTax = tax - rebate;

taxModel.setTax(tax);

taxModel.setRebate(rebate);

taxModel.setNetTax(netTax);

return TaxModel;

}

}

→ result.jsp

<%@ page isELIgnored="false" %>

<html>

    <body>

        <pre>

Income : \${taxModel.income}

Age : \${taxModel.age}

Tax : \${taxModel.tax}

Rebate : \${taxModel.rebate}

    </pre>

    </body> </html>