

ZUMBA CLASS MANAGEMENT APP

Objective:

To build an application to handle data on students learning Zumba, which will manage the enrollment of students in the correct batch and provide appropriate timing for their classes.

Prerequisites:

- Java
- Servlets
- JSP
- Maven
- JDBC
- MySQL

Task:

1. Create a Dynamic Web Project in Eclipse and configure Maven
2. Create Java Classes for participants and batches
3. Create Servlets for participants and batches related to communication
4. Create HTML Pages for CRUD operations related to batches and participants
5. Perform CRUD operations using database methods using JDBC
6. Create JSP Pages for query parameters
7. Build and Run the Project on the Apache Tomcat web server
8. Package the project as a Jar file using Maven Package Goal

Modules:

1. **Admin:** The admin has full control over the system, including managing participants and batches.
2. **User:** The user (participant) can log in to view available batches and register for classes after the admin has registered them.

Login Credentials:

Admin Login:

- Authentication Method: Hard-coded
- Email: admin@zumba.com
- Password: admin@123

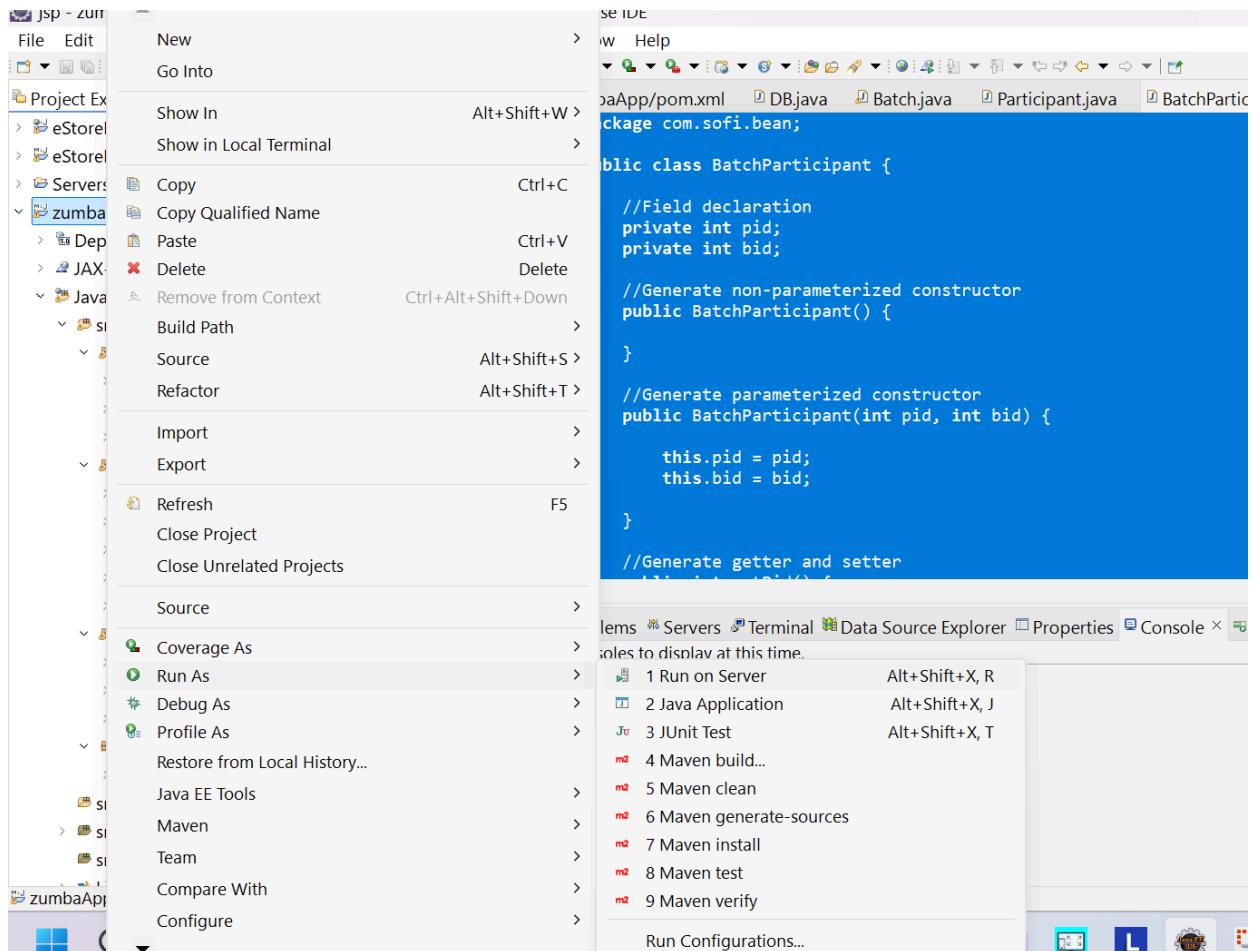
User Login:

- After the admin registers a participant, the user can log in using their Email and Password to access their personalized dashboard and enroll in available batches.

Steps to Run the Project:

1. Clone the Repository:
2. `git clone https://github.com/nayaksofia/zumbaApp_MavenJsp.git`
3. Set up the Database:
 - Ensure you have MySQL installed.
 - Create a database and import the provided schema file to set up tables for participants, batches, and enrollments.
4. Configure Database Connection:
 - Update the database connection details in the configuration file (DB.java) to match your local MySQL setup.
5. Run the Application:
 - Use a Java IDE (like IntelliJ IDEA or Eclipse) or run the project using Maven
 - Access the application through your web browser.
6. Log in as Admin:
 - Use the admin credentials to log in and manage participants and batches.
7. Log in as User:

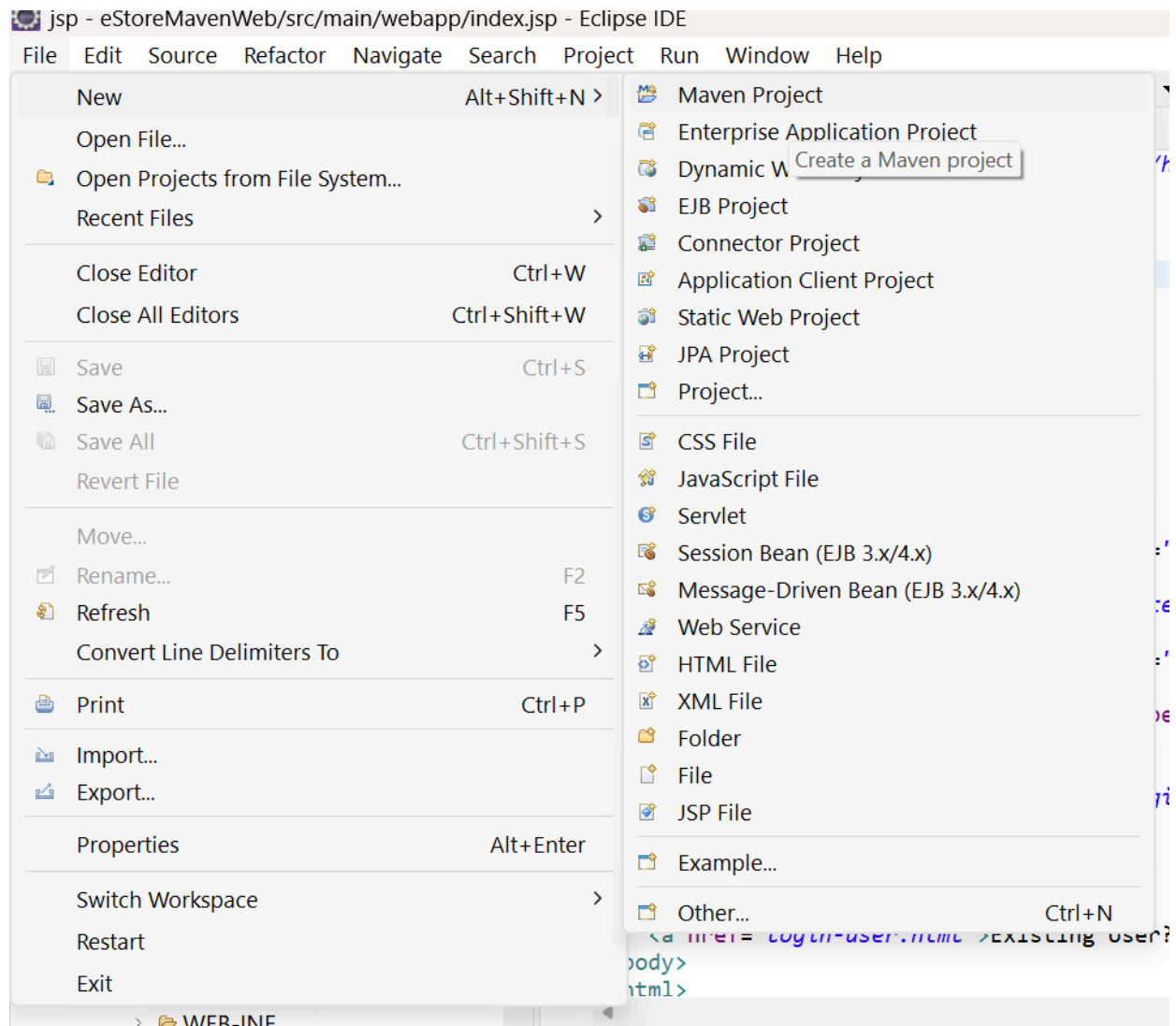
- After admin registration, users can log in with their email and password to register for batches.

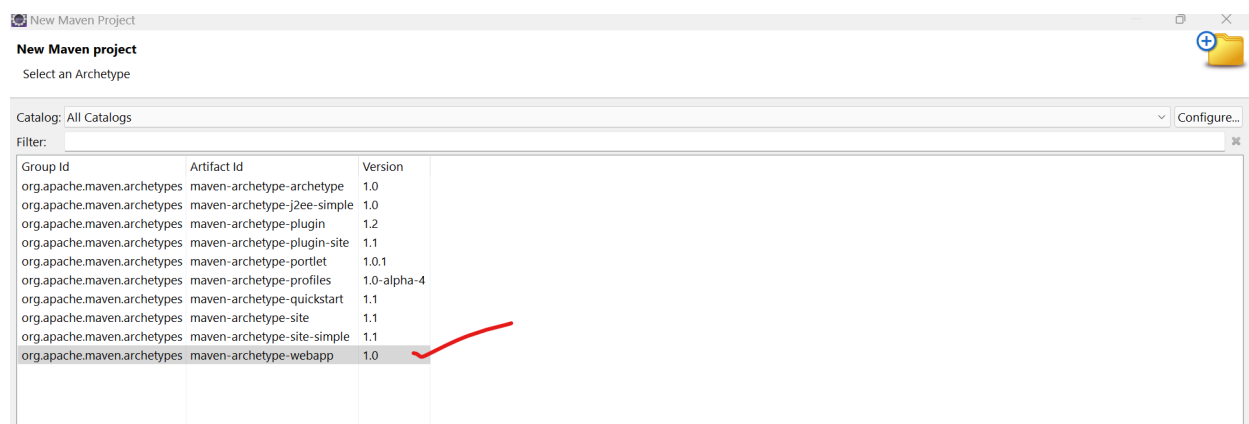
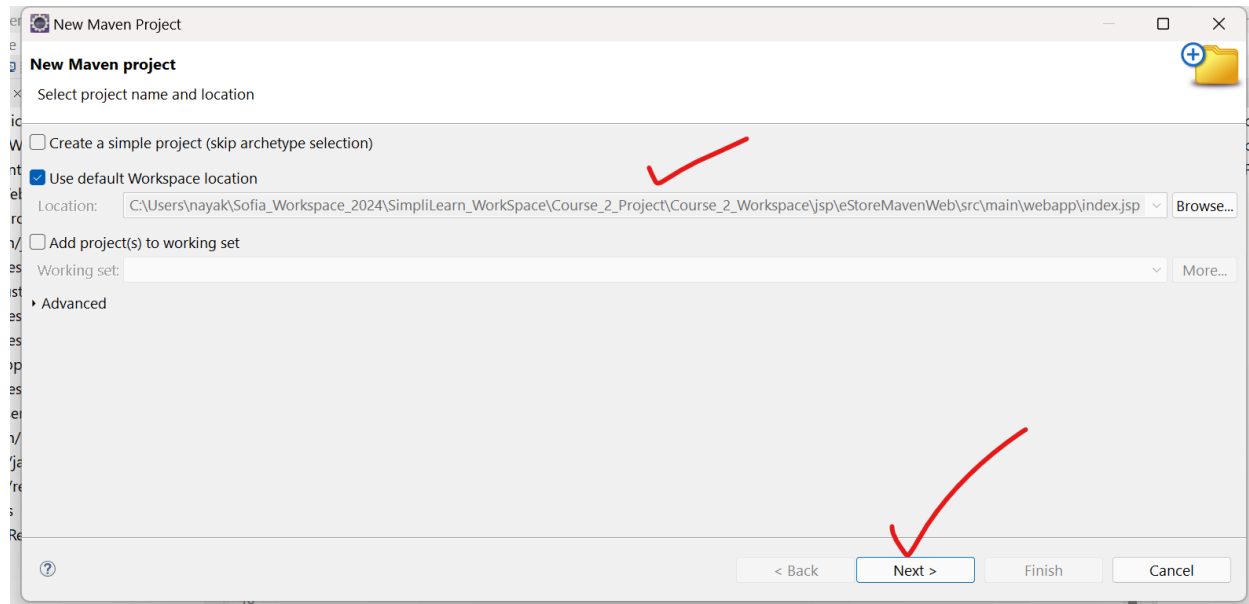


Backend Part:

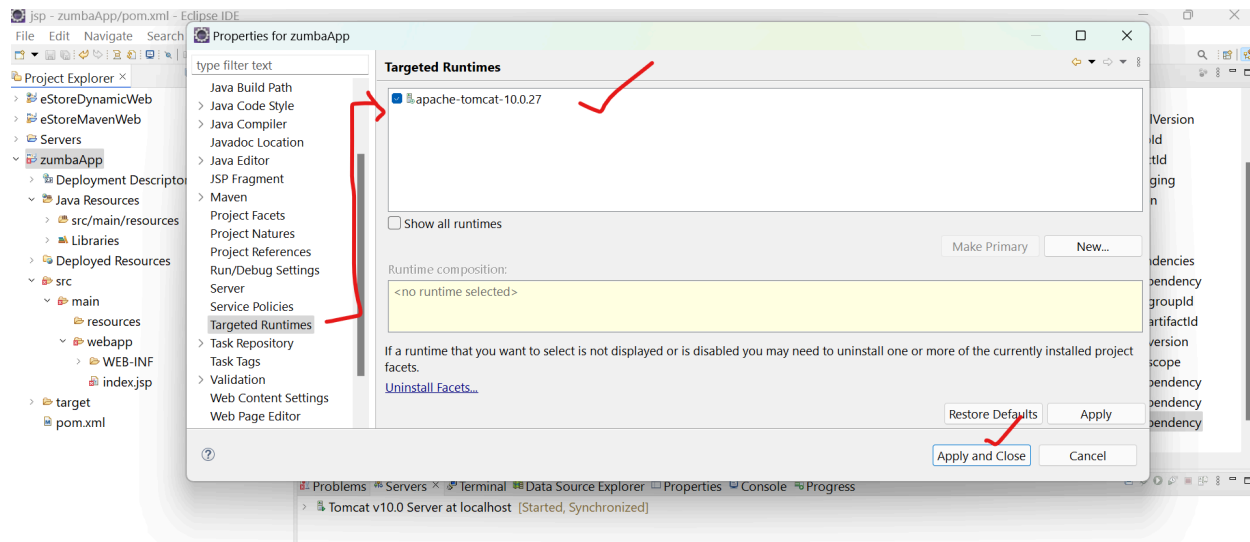
Step 1: Create a Maven project

Name of Project: zumbaApp

























Step-3: Set Targeted Runtime By Going through Properties



Step-4: File Structure of zumbaApp

Choose JavaSE [1.8] and update the project by Right Click on projectName→ Maven→ Update Project

- ▼  zumbaApp
 - >  Deployment Descriptor: Archetype Created Web Application
 - ▼  Java Resources
 - >  src/main/java
 - ▼  src/main/resources
 - >  Deployment Descriptor: Archetype Created Web Application
 - >  src/test/java
 - >  src/test/resources
 - ▼  Libraries
 - >  JRE System Library [JavaSE-1.8]
 - >  Maven Dependencies
 - >  Server Runtime [apache-tomcat-10.0.27]
 - ▼  Deployed Resources
 - >  WEB-INF
 -  index.jsp
 - ▼  src
 - >  main
 - >  test
 - >  target
 -  pom.xml

Step-5: Coding Part

BackendOperation:

Step-5.1: Create DB Class file for database connection

```
package com.sofi.db;
```

```
import java.sql.Connection;  
import java.sql.DriverManager;
```



```

import java.sql.SQLException;

public class DB {

    // Singleton instance
    private static DB db = new DB();

    private DB() {
        try {

            Class.forName("com.mysql.cj.jdbc.Driver");
            System.out.println("[DB] Driver Loaded!!!");

        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    // Lazy initialization of singleton
    public static DB getDB() {
        return db;
    }

    // Method to get a new connection for each call
    public Connection getConnection() throws SQLException {
        String url = "jdbc:mysql://localhost:3306/zumba";
        String user = "root";
        String password = "root";
        return DriverManager.getConnection(url, user, password);
    }
}

```

Step-5.3: Create Bean Package to represent tables in the databases

Batch.java

```

package com.sofi.bean;

import java.sql.Date;
import java.sql.Time;
import java.time.LocalDateTime;

```

```
public class Batch {

    // Field Declarations
    private int bid;
    private String instructor;
    private Date date;
    private String schedule;
    private Time time;

    // Non-parameterized constructor
    public Batch() {
    }

    // Parameterized constructor
    public Batch(int bid, String instructor, Date date, String schedule, Time time) {
        this.bid = bid;
        this.instructor = instructor;
        this.date = date;
        this.schedule = schedule;
        this.time = time;
    }

    // Getters and Setters
    public int getBid() {
        return bid;
    }

    public void setBid(int bid) {
        this.bid = bid;
    }

    public String getInstructor() {
        return instructor;
    }

    public void setInstructor(String instructor) {
        this.instructor = instructor;
    }

    public Date getDate() {
        return date;
    }
}
```

```

    public void setDate(Date date) {
        this.date = date;
    }

    public String getSchedule() {
        return schedule;
    }

    public void setSchedule(String schedule) {
        this.schedule = schedule;
    }

    public Time getTime() {
        return time;
    }

    public void setTime(Time time) {
        this.time = time;
    }

    // toString method for debugging
    @Override
    public String toString() {
        return "Batch [bid=" + bid + ", instructor=" + instructor + ", date=" + date + ", schedule=" +
        schedule
            + ", time=" + time + "];"
    }
}

```

Participant.java

```

package com.sofi.bean;
public class Participant {
    //Field declaration :: pid, name, age, email, password
    private int pid;
    private String name;
    private int age;
    private String email;
    private String password;

    //Non-parameterized constructor
    public Participant() {

    }
}

```

//Parameterized Constructor

```
public Participant(int pid, String name, int age, String email, String password) {  
    super();  
    this.pid = pid;  
    this.name = name;  
    this.age = age;  
    this.email = email;  
    this.password = password;  
}
```

//Getter and Setter

```
public int getPid() {  
    return pid;  
}  
public void setPid(int pid) {  
    this.pid = pid;  
}  
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public int getAge() {  
    return age;  
}  
public void setAge(int age) {  
    this.age = age;  
}  
public String getEmail() {  
    return email;  
}  
public void setEmail(String email) {  
    this.email = email;  
}  
public String getPassword() {  
    return password;  
}  
public void setPassword(String password) {  
    this.password = password;  
}  
//toString Method , exclude password for security  
@Override  
public String toString() {  
    return "Participant [pid=" + pid + ", name=" + name + ", age=" + age + ", email=" +  
    + email +  
    " ]";  
}
```

```
}
```

BatchParticipant.java

```
package com.sofi.bean;
public class BatchParticipant {
    //Field declaration
    private int pid;
    private int bid;
    //Generate non-parameterized constructor
    public BatchParticipant() {

    }

    //Generate parameterized constructor
    public BatchParticipant(int pid, int bid) {

        this.pid = pid;
        this.bid = bid;

    }

    //Generate getter and setter
    public int getPid() {
        return pid;
    }
    public void setPid(int pid) {
        this.pid = pid;
    }
    public int getBid() {
        return bid;
    }
    public void setBid(int bid) {
        this.bid = bid;
    }

    //Generate toString method
    @Override
    public String toString() {
        return "ParticipantBatch [pid=" + pid + ", bid=" + bid + ", pname=" + "]";
    }
}
```

```
}
```

Step-5.4: MySQLWorkbench: Create Database and tables [Database name: zumba]

```
create database zumba;
```

```
use zumba;
```

```
show tables;
```

```
CREATE TABLE Batch (  
    bid INT AUTO_INCREMENT PRIMARY KEY,  
    instructor VARCHAR(100) NOT NULL,  
    date DATE NOT NULL,  
    schedule VARCHAR(100) NOT NULL,  
    time TIME NOT NULL  
);
```

```
CREATE TABLE Participant (  
    pid INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    age INT NOT NULL,  
    email VARCHAR(100) NOT NULL,  
    password VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE ParticipantBatch (  
    pid INT NOT NULL,  
    bid INT NOT NULL,  
    PRIMARY KEY (pid, bid),  
    FOREIGN KEY (pid) REFERENCES Participant(pid) ON DELETE CASCADE,  
    FOREIGN KEY (bid) REFERENCES Batch(bid) ON DELETE CASCADE  
);
```

```
select * from Batch;
```

```
select * from Participant;
```





```
select * from ParticipantBatch;
```

```
describe Batch;
```







```
describe Participant;
```

```
describe ParticipantBatch;
```

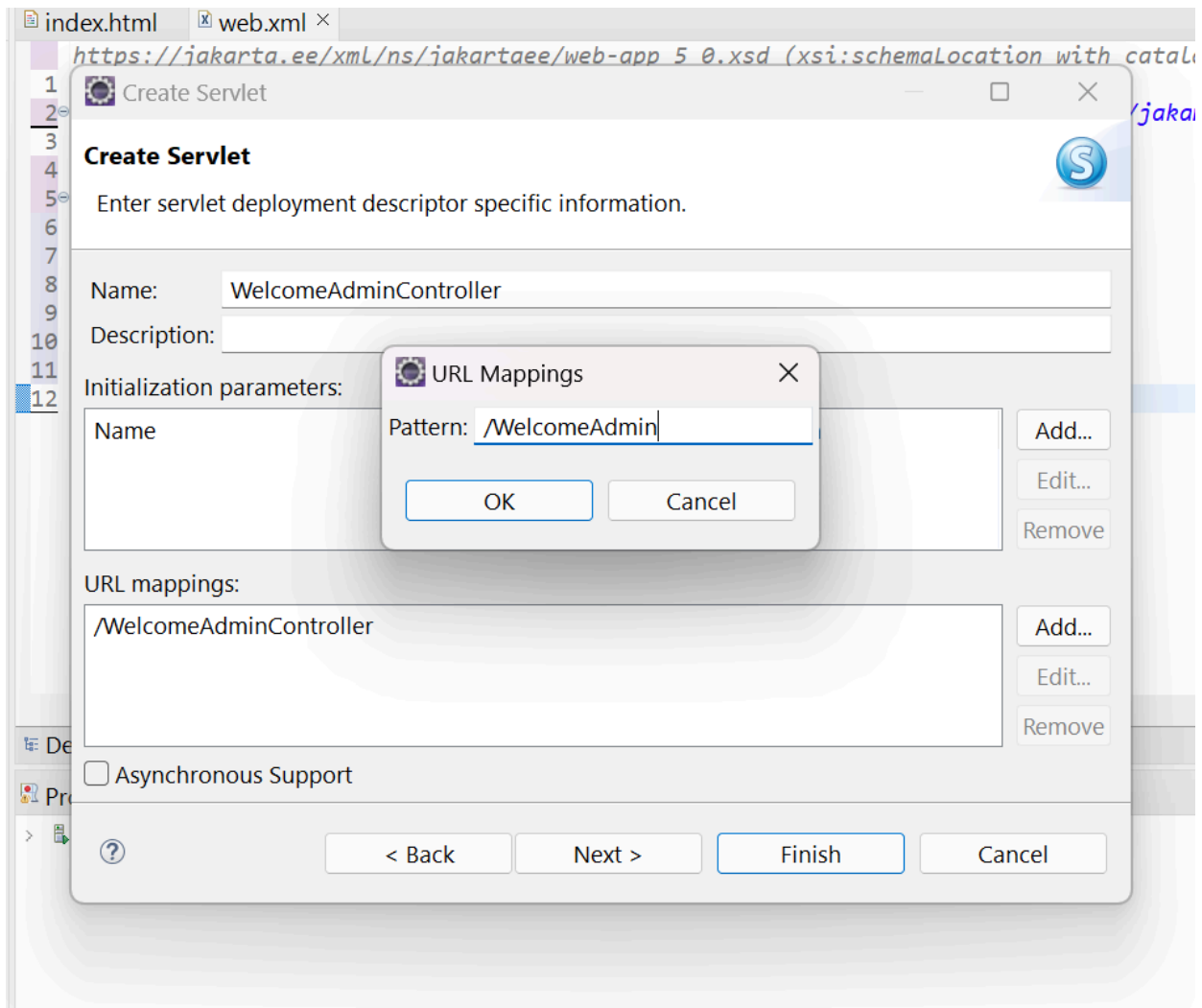
Step-5.5: Create DAO Classes for CRUD operation

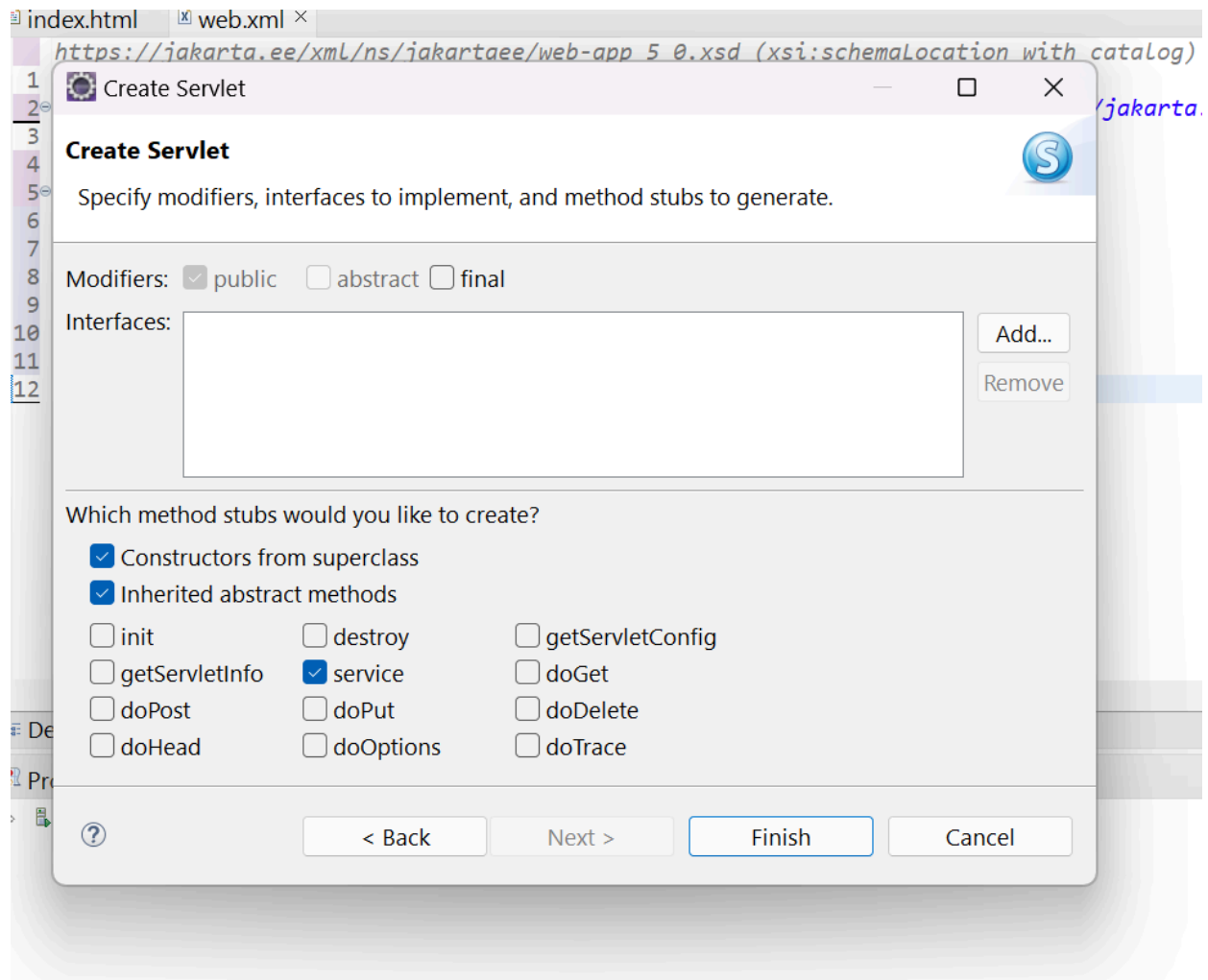
- ▼  com.sofi.dao
 - >  BatchDAO.java
 - >  ParticipantBatchDAO.java
 - >  ParticipantDAO.java

Step-5.6: Create Controller Package to manage servlet operation

- ▼  com.sofi.controller
 - >  AddBatchController.java
 - >  AddParticipantController.java
 - >  LogoutServlet.java
 - >  RegisterBatchController.java
 - >  WelcomeAdminController.java

Few eg:





1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Create Servlet

Create Servlet

Specify class file destination.

Project: zumbaApp

Source folder: /zumbaApp/src/main/java

Java package: com.sofi.controller

Class name: AddParticipantController

Superclass: jakarta.servlet.http.HttpServlet

☐ Use an existing Servlet class or JSP

Class name: AddParticipantController

< Back

Next >

Finish

Cancel

<%@ page language="java" contentType="text/html; charset=UTF-8"%>

Create Servlet

Enter servlet deployment descriptor specific information.

Name: AddParticipantController

Description:

Initialization parameters:

Name

URL mappings:

/AddParticipantController

☐ Asynchronous Support

< Back Next > Finish Cancel

URL Mappings

Pattern: /AddParticipant ✓

OK Cancel

Logout Servlet:

web.xml welcomeAdmin.jsp headerAdmin.jsp welcomeAdmin.jsp headerUser.jsp

Create Servlet

Specify class file destination.

Project: zumbaApp

Source folder: /zumbaApp/src/main/java Browse...

Java package: com.sofi.controller Browse...

Class name: LogoutServlet

Superclass: jakarta.servlet.http.HttpServlet Browse...

☐ Use an existing Servlet class or JSP

Class name: LogoutServlet Browse...

? < Back Next > Finish Cancel

```
request.getRequestDispatcher("/headerAdmin.jsp").include(request, response);
```

web.xml welcomeAdmin.jsp headerAdmin.jsp welcomeAdmin.jsp headerUser.jsp

Create Servlet

Enter servlet deployment descriptor specific information.

Name: LogoutServlet

Description:

Initialization parameters:

Name

URL mappings:

/LogoutServlet

☐ Asynchronous Support

? < Back Next > Finish Cancel

URL Mappings


Pattern: /Logout

OK Cancel

Add... Edit... Remove

```
);  
);  
response);  
response);  
}
```

```
ge language="java" contentType="text/html; charset=UTF-8"
```

ge  Create Servlet

Create Servlet

Specify class file destination.

Project: zumbaApp

Source folder: /zumbaApp/src/main/java

Browse...

Java package: com.sofi.dao

Browse...

Class name: ViewParticipantController

Superclass: jakarta.servlet.http.HttpServlet

Browse...

☐ Use an existing Servlet class or JSP

Class name: ViewParticipantController

Browse...



< Back

Next >

Finish

Cancel

language= java contentType= text/html; charset=UTF-8

Create Servlet

Create Servlet

Enter servlet deployment descriptor specific information.

Name: ViewParticipantController

Description:

Initialization parameters:

Name

URL mappings:

/ViewParticipantController

☐ Asynchronous Support

?

< Back

Next >

Finish

Cancel

URL Mappings

Pattern: /ViewParticipant

OK

Cancel

Add...

Edit...


Remove


Add...

Edit...

Remove


language="java" contentType="text/html; charset=UTF-8"

 Create Servlet



Specify class file destination.

Project:	<input type="text" value="zumbaApp"/>	▼
Source folder:	<input type="text" value="/zumbaApp/src/main/java"/>	<input type="button" value="Browse..."/>
Java package:	<input type="text" value="com.sofi.controller"/>	<input type="button" value="Browse..."/>
Class name:	<input type="text" value="AddBatchController"/>	
Superclass:	<input type="text" value="jakarta.servlet.http.HttpServlet"/>	<input type="button" value="Browse..."/>
<input type="checkbox"/> Use an existing Servlet class or JSP		
Class name:	<input type="text" value="AddBatchController"/>	<input type="button" value="Browse..."/>



language="java" contentType="text/html; charset=UTF-8"

Create Servlet

Enter servlet deployment descriptor specific information.

Name: AddBatchController

Description:

Initialization parameters:

URL mappings:

Asynchronous Support

< Back Next > Finish Cancel

URL Mappings

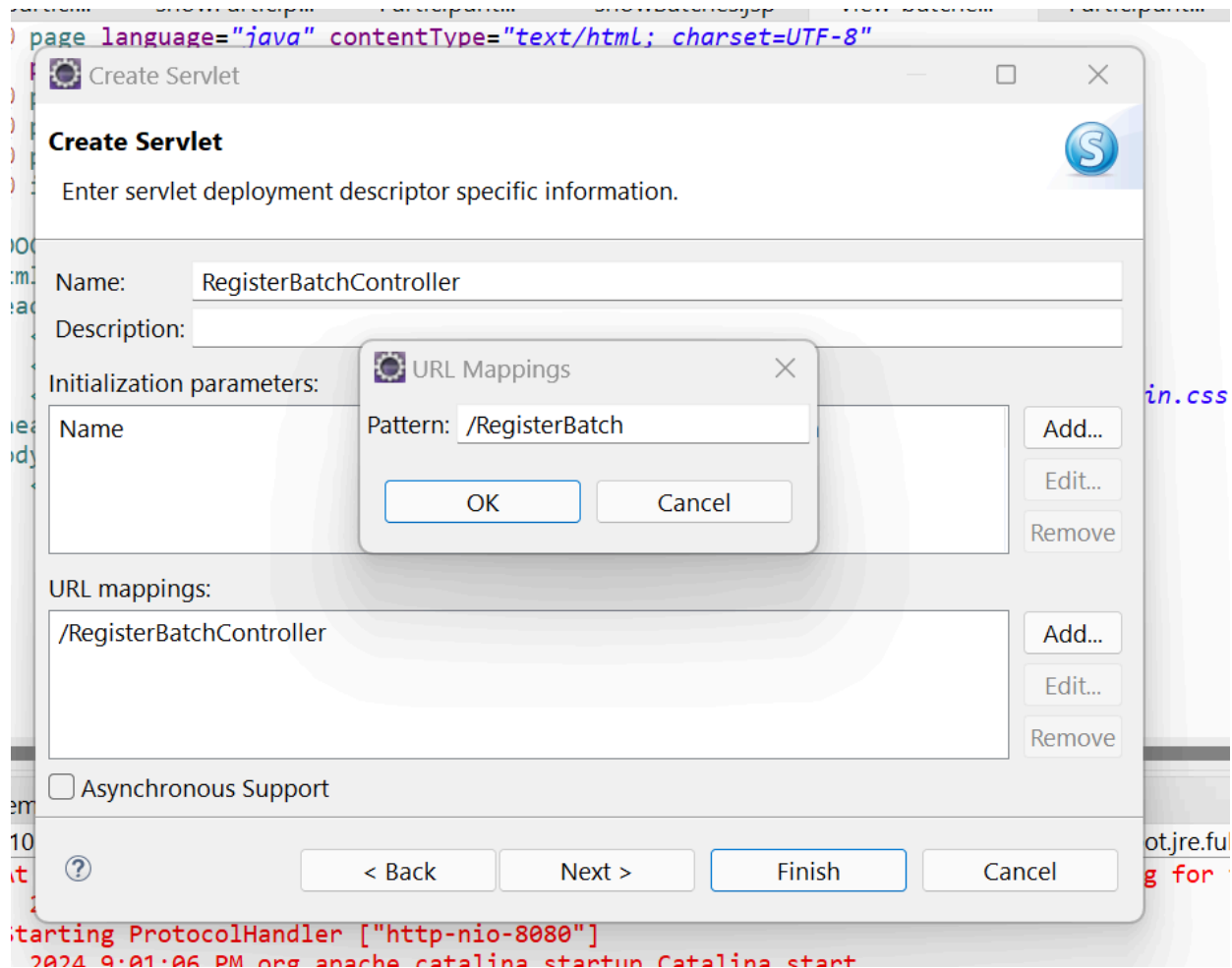
Pattern: /AddBatch

OK Cancel

Add... Edit... Remove

Add... Edit... Remove

?





Front-end Operation:


Create Required JSP and HTML file


▼  main


>  java


 resources


▼  webapp


>  images


>  WEB-INF


 addBatch.jsp


 addParticipant.jsp


 adminLogin.jsp


 deleteBatchController.jsp


 deleteParticipantController.jsp


 editBatch.jsp


 editBatchController.jsp


 editParticipant.jsp

 editParticipantController.jsp


 footer.jsp


 headerAdmin.jsp

 headerUser.jsp

 index.html

 registeredBatchParticipantList.jsp


 showBatches.jsp


 showParticipants.jsp

 successDeleteBatch.jsp

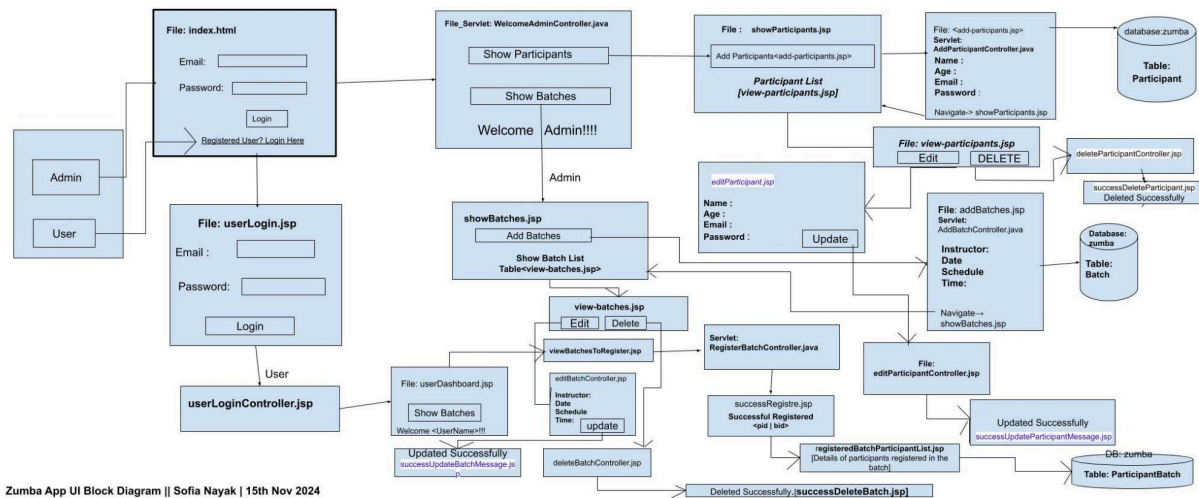
 successDeleteParticipant.jsp

 successRegister.jsp

 successUpdateBatchMessage.jsp

 successUpdateParticipantMessage.jsp

Block Diagram: Flow of Zumba app

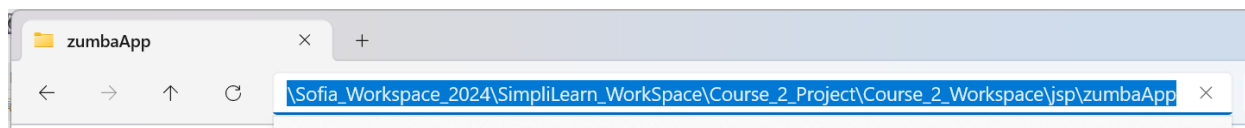


Step [Final]: Git and GitHub

Github Link: [Zumba App]

<https://github.com/nayaksofia/zumbaApp> MavenJsp.git

1. Select the **path**



2. Write **cmd** and the **command prompt** will open

```
C:\Windows\System32\cmd
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>
```

3. Initiate git command

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git init
Initialized empty Git repository in C:/Users/nayak/Sofia_Workspace_2024/SimpliLearn_WorkSpace/Course_2_Project/Course_2_Workspace/jsp/zumbaApp/.git/

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .classpath
        .project
        .settings/
        pom.xml
        src/
        target/

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git add .

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git commit -m "initial commit"
[master (root-commit) 95f4561] initial commit
```

4. github

nayaksofia / zumbaApp_MavenJsp

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

zumbaApp_MavenJsp

Public

Pin Unwatch 1 Fork 0 Star 0

Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Get started with GitHub Copilot

Add collaborators to this repository

Search for people using their GitHub username or email address.

Invite collaborators

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH

https://github.com/nayaksofia/zumbaApp_MavenJsp.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# zumbaApp_MavenJsp" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/nayaksofia/zumbaApp_MavenJsp.git
git push -u origin master
```

...or push an existing repository from the command line

...or push an existing repository from the command line

```
git remote add origin https://github.com/nayaksofia/zumbaApp_MavenJsp.git
git branch -M master
git push -u origin master
```

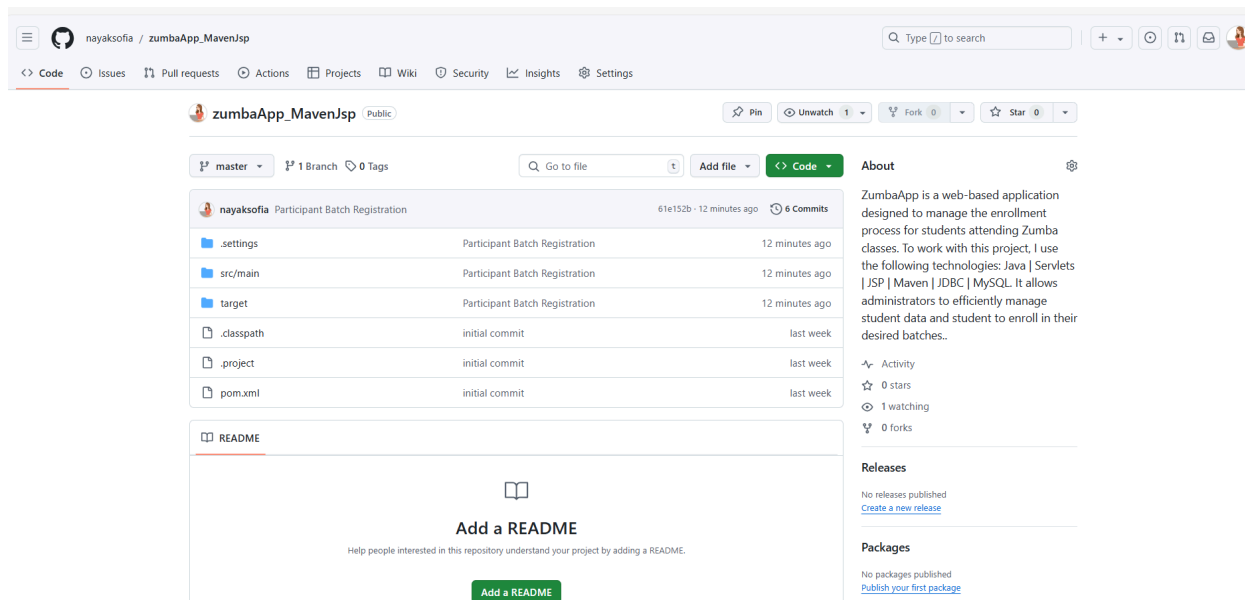
💡 ProTip! Use the URL for this page when adding GitHub as a remote.

```
C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git log --oneline
61e152b (HEAD -> master) Participant Batch Registration
cbbfc15 ParticipantBatchDAO : Insert Operation
a2bfc72 ParticipantDAO: implement try-with-resources
b5661c9 BatchDAO : implement try-with-resource'
5764ea4 CRUD Operation BatchDAO and ParticipantDAO
95f4561 initial commit

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git remote add origin http
s://github.com/nayaksofia/zumbaApp_MavenJsp.git

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git push -u origin master
Enumerating objects: 217, done.
Counting objects: 100% (217/217), done.
Delta compression using up to 4 threads
Compressing objects: 100% (163/163), done.
Writing objects: 100% (217/217), 1.16 MiB | 1.22 MiB/s, done.
Total 217 (delta 45), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (45/45), done.
To https://github.com/nayaksofia/zumbaApp_MavenJsp.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>
```



The screenshot displays the GitHub repository page for `nayaksofia/zumbaApp_MavenJsp`. The repository is public and has 6 commits. The commit history shows a series of commits related to Participant Batch Registration. The file structure includes `.settings`, `src/main`, `target`, `.classpath`, `.project`, and `pom.xml`. The README section is currently empty, with a prompt to "Add a README".

Latest changes on github repository to local repository:

```

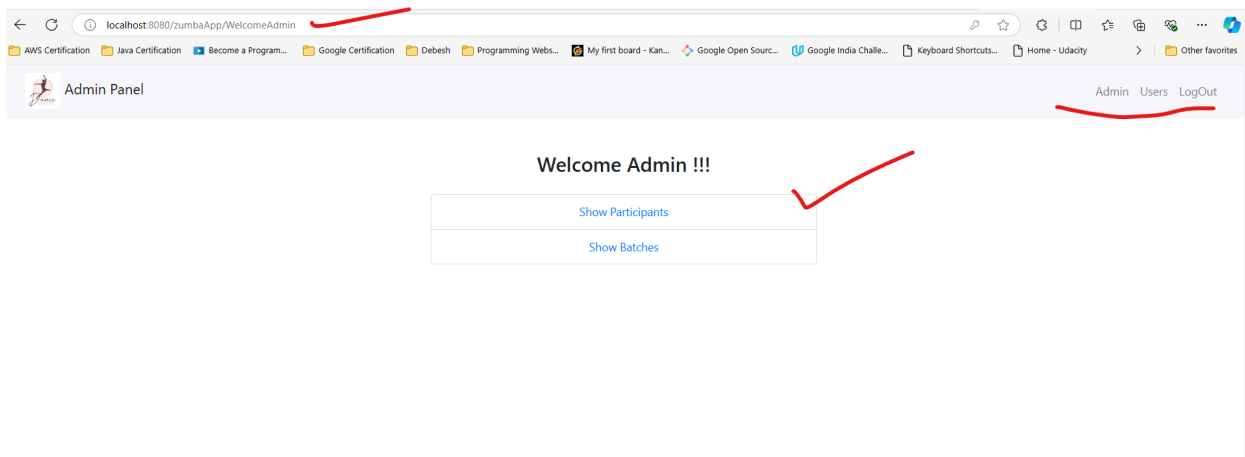
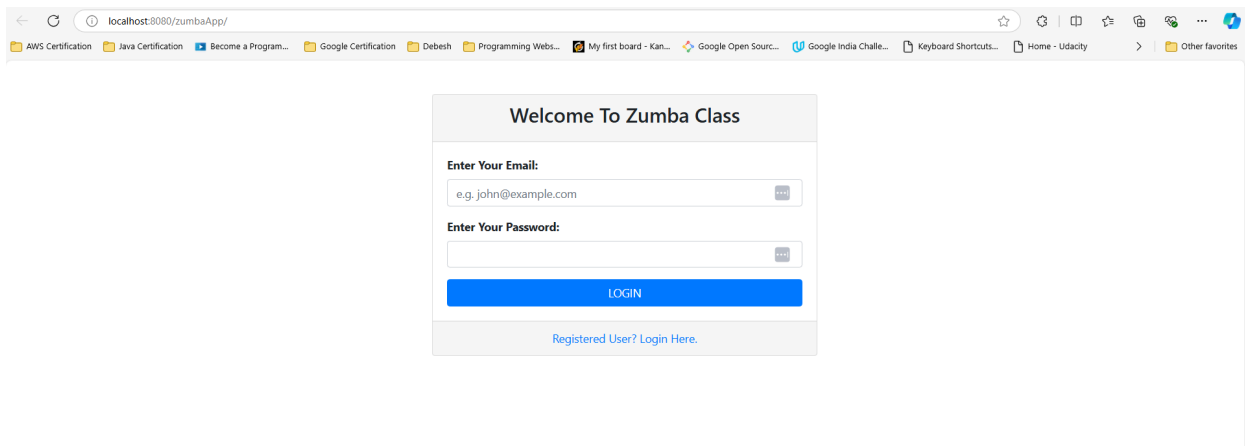
C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.65 KiB | 11.00 KiB/s, done.
From https://github.com/nayaksofia/zumbaApp_MavenJsp
* branch          master       -> FETCH_HEAD
   61e152b..f927d6a master     -> origin/master
Updating 61e152b..f927d6a
Fast-forward
 README.md | 43 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 43 insertions(+)
 create mode 100644 README.md

C:\Users\nayak\Sofia_Workspace_2024\SimpliLearn_WorkSpace\Course_2_Project\Course_2_Workspace\jsp\zumbaApp>git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

```

Result / OutPut:



Add Participant

View Participants

Back To Admin Dashboard

Add Participant

Participant Name

Participant Age

Participant Email

Participant Password

Add Participant

Participants Dashboard

Add Participant

Participant Name

Participant Age

Participant Email

Participant Password

Add Participant

Participants Dashboard

Silli Jha added to the database successfully!

Back To Participant Directory

List of Participants!!!

Participant ID	Participant Name	Participant Age	Participant Email	Participant Password	Edit	Delete
1	Sofia Nayak	34	worldofsimpleprogrammers@gmail.com	ps@555	Edit	Delete
2	Shagun Patel	25	sahgun25@gamil.com	ps@123	Edit	Delete
3	Sona Sha	34	shasona@gmail.com	ps@234	Edit	Delete
6	Prangya Paramita Muduli	38	pmuduli@gmail.com	ps@muduli	Edit	Delete
7	Adf adf	78	adf@gmail.com	ps@adfadf	Edit	Delete
8	Anna Patel	56	panna@gmail.com	ps@anna	Edit	Delete
13	Alli Aga	23	agalli@gmail.com	ps@agal	Edit	Delete
14	Silli Jha	54	sillishine@gmail.com	ps@silli	Edit	Delete

Back To Participants Dashboard

Admin: View Batch

localhost:8080/zumbaApp/view-batches.jsp

AWS Certification Java Certification Become a Program... Google Certification Debash Programming Webs... My first board - Kan... Google Open Sourc... Google India Challe... Keyboard Shortcuts... Home - Udacity Other favorites

List of Batches!!!

Batch ID	Instructor Name	Date	Schedule	Time	Edit	Delete
1	Padmaja Patel	2024-11-15	Morning	07:35:00	Edit	Delete
2	Swagat Patel	2024-11-22	Evening	10:38:00	Edit	Delete

Back To BatchDashboard

localhost:8080/zumbaApp/editBatch.jsp?id=1

Java Certification Become a Program... Google Certification Debash Programming Webs... My first board - Kan... Google Open Sourc... Google India Challe... Keyboard Shortcuts... Home - Udacity

Edit Batch

Instructor Name

Date

Schedule

Time

Update Batch

Batch List

localhost:8080/zumbaApp/successUpdateBatchMessage.jsp

Java Certification Become a Program... Google Certification Debash Programming Webs... My first board - Kan... Google Open Sourc... Google India Challe... Keyboard Shortcuts... Home - Udacity

Batch updated successfully!

View Batch

Now:

localhost:8080/zumbaApp/view-batches.jsp

Java Certification Become a Program... Google Certification Debash Programming Webs... My first board - Kan... Google Open Sourc... Google India Challe... Keyboard Shortcuts... Home - Udacity

List of Batches!!!

Batch ID	Instructor Name	Date	Schedule	Time	Edit	Delete
1	Deepti Patel	2024-11-27	Morning	12:40:00	Edit	Delete
2	Swagat Patel	2024-11-22	Evening	10:38:00	Edit	Delete

Back To BatchDashboard

localhost:8080/zumbaApp/addBatch.jsp?

AWS Certification Java Certification Become a Program... Google Certification Debash Programming Webs... My first board - Kan... Google Open Sourc... Google India Challe... Keyboard Shortcuts... Home - Udacity Other favorites

Admin Panel Admin Users LogOut

Add Batch

Instructor Name
Govinda Agrawal

Date
29-11-2024

Schedule
Afternoon

Time
08:30

Add Batch

Batch Dashboard

localhost:8080/zumbaApp/AddBatchController?instructor=Govinda+Agrawal&date=2024-11-29&schedule=Afternoon&time=08%3A30

AWS Certification Java Certification Become a Program... Google Certification Debash Programming Webs... My first board - Kan... Google Open Sourc... Google India Challe... Keyboard Shortcuts... Home - Udacity Other favorite

Admin Panel Admin Users LogOut

Govinda Agrawal added to the database successfully!

Back To Batch Directory

List of Batches!!!

Batch ID	Instructor Name	Date	Schedule	Time	Edit	Delete
1	Deepti Patel	2024-11-27	Morning	12:40:00	Edit	Delete
2	Swagat Patel	2024-11-22	Evening	10:38:00	Edit	Delete
4	Govinda Agrawal	2024-11-29	Afternoon	08:30:00	Edit	Delete

Back To BatchDashboard

Add Batch

View Batches

Back To Admin Dashboard

User Login

Enter Your Email:

e.g. john@example.com

Enter Your Password:

LOGIN

Admin ? Login Here.

User Login

Enter Your Email:

Enter Your Password:

LOGIN

[Admin ? Login Here.](#)

Welcome To Zumba Class!!!

[Show Batches](#)

List of Batches!!!

Batch ID	Instructor Name	Date	Schedule	Time	Register
1	Padmaja Patel	2024-11-15	Morning	07:35:00	Register
2	Swagat Patel	2024-11-22	Evening	10:38:00	Register

[Back To UserDashboard](#)

[View Registered Participant- Batch List](#)

localhost:8080/zumbaApp/successRegister.jsp?pid=14&bid=2

AWS Certification Java Certification Become a Program... Google Certification Debesh Programming Webs... My first board - Kan... Google Open Sour... Google India Challe... Keyboard Shortcuts... Home - Udacity Other favorites

User Panel Users LogOut

Registration Successful!

Your registration was successful for the following:

Participant ID: 14

Batch ID: 2

[View Registered Participant- Batch List](#)

[Batch List](#)