

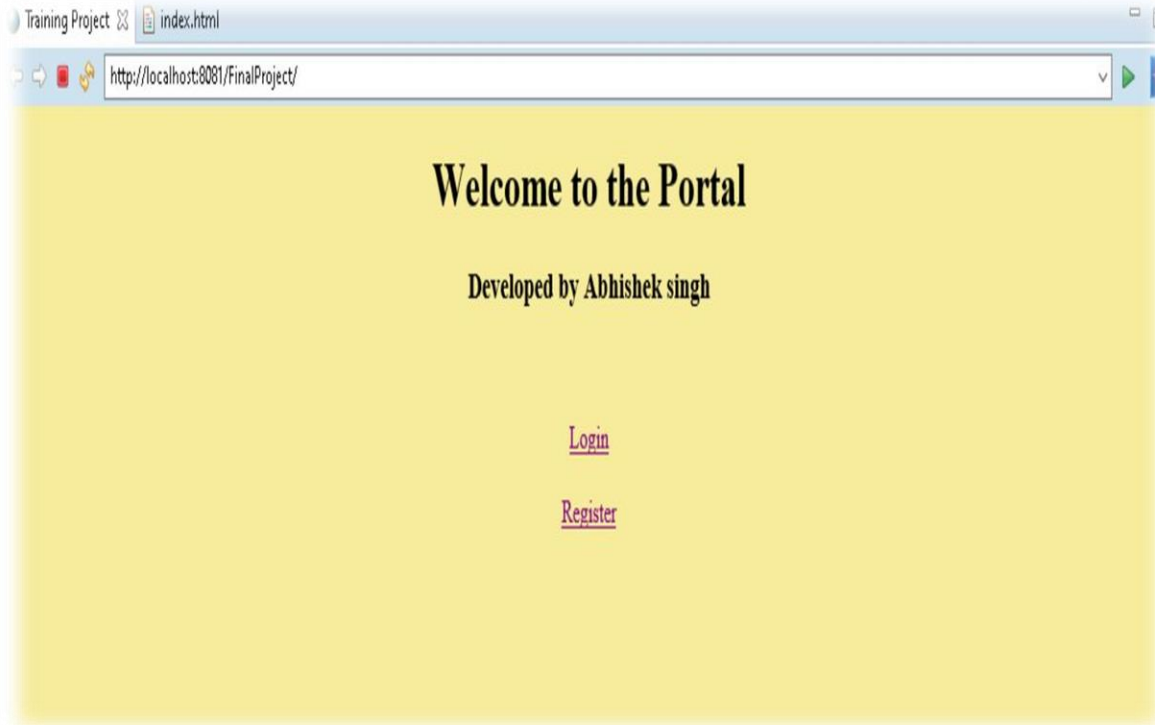
AMDOCS TRAINING PROJECT

ABHISHEK SINGH

180949



PROJECT FEATURES



New User Registration

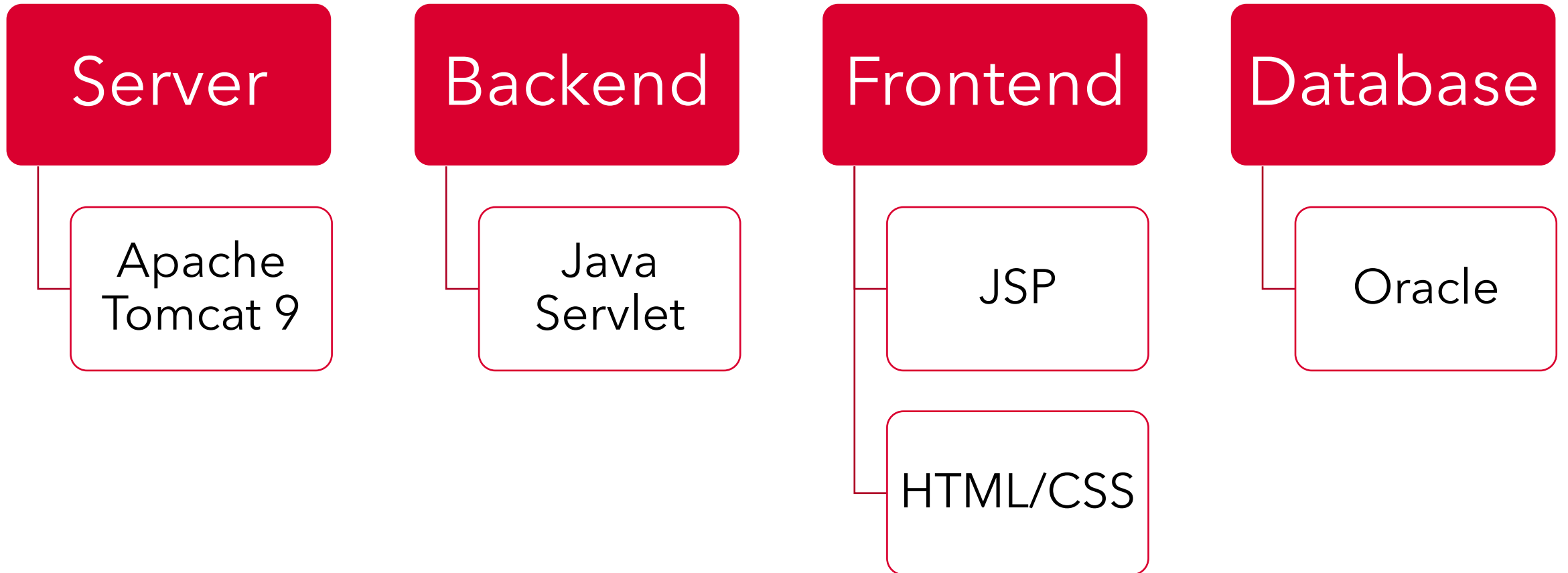
User

- View and Enroll Course
- Give Feedback
- View Admin

Admin

- Add and View Course
- View Feedback
- View Admin and User
- Add New Admin

TECHNOLOGY USED



PROJECT STRUCTURE

com.Amdocs.main

- LoginServlet
- Login
- LogOut Servlet
- DBConnect
- Course
- Feedback

com.Amdocs.user

- UserServlet
- User
- UserRegisterServlet
- MyCourseServlet
- EnrollCourseServlet
- FeedbackServleet

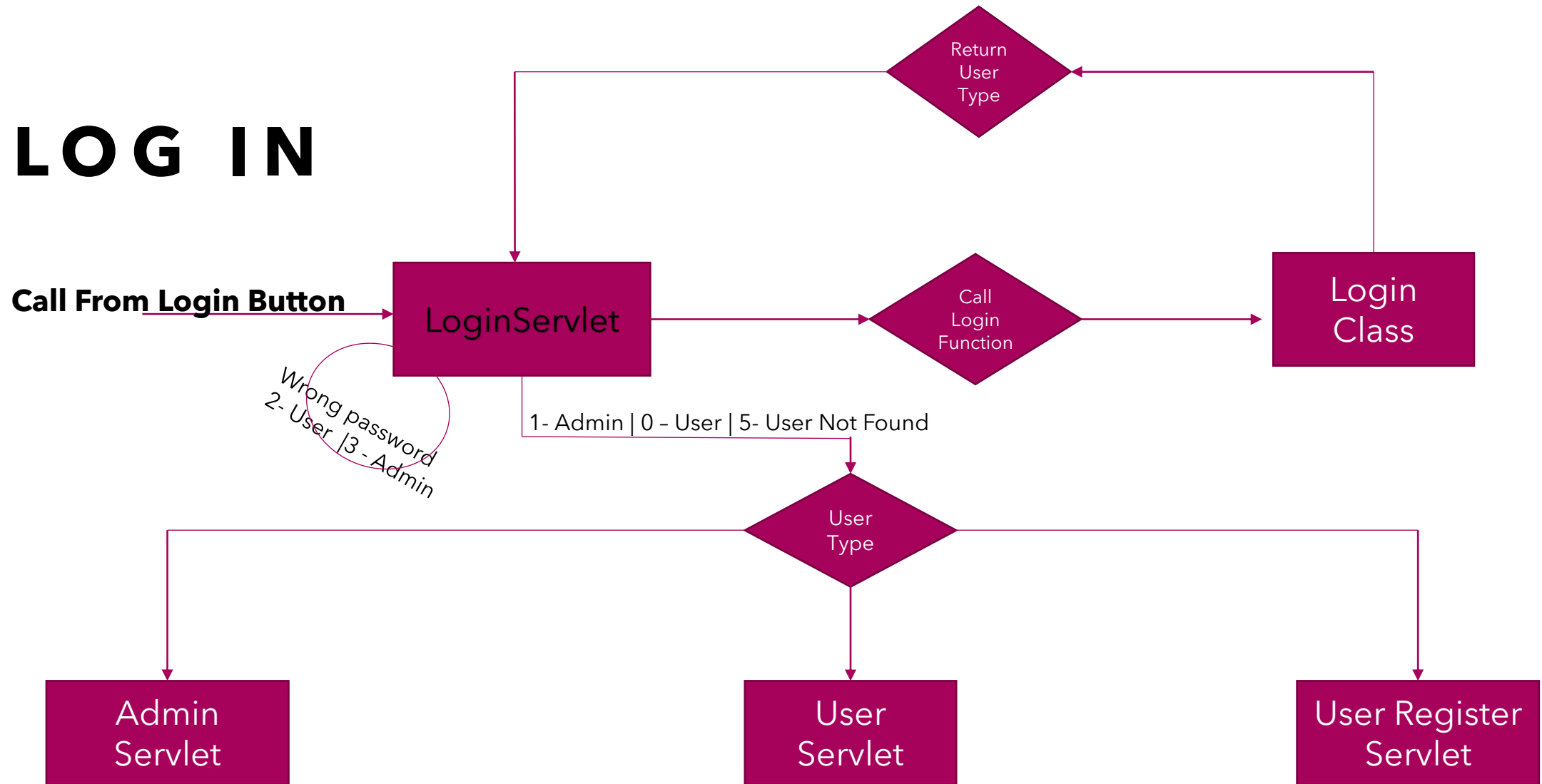
com.Amdocs.admin

- AdminServlet
- Admin
- AdminRegisterServlet
- AddCourseServlet
- ViewCourseServlet
- ViewAdminServlet
- ViewFeedbackServlet
- ViewUserServlet

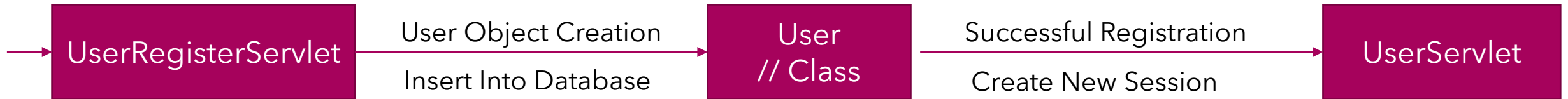
com.Amdocs.Test

- UserTypeDetectionTest

LOG IN



ADD USER

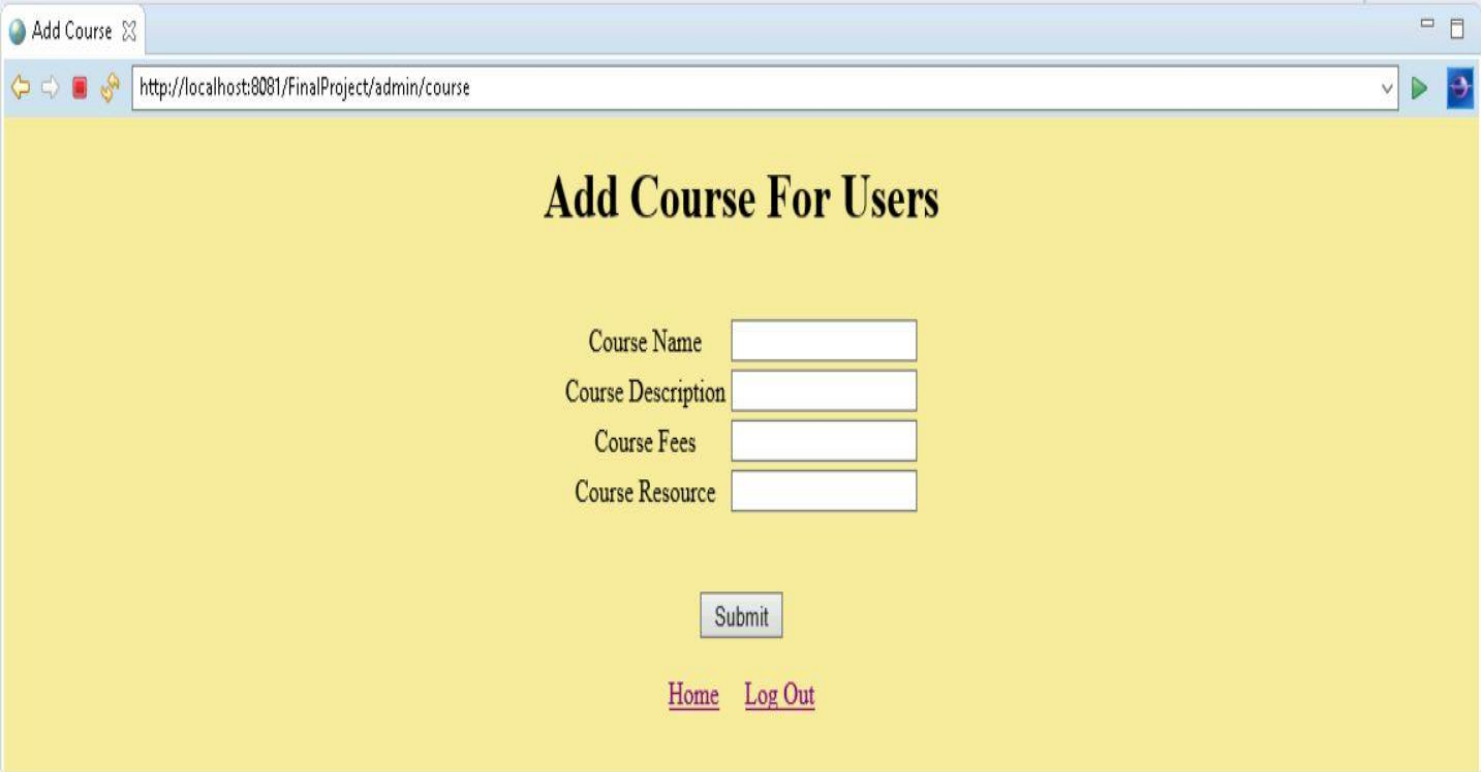


A screenshot of a web browser window titled "Register" showing the "User Registration Form". The browser's address bar displays the URL `http://localhost:8081/FinalProject/user-register`. The form is set against a light yellow background and includes the following fields:

- Name
- Phone No
- Email
- Password
- Address
- Upload Photo

A "Submit" button is located at the bottom center of the form.

ADD COURSE



The screenshot shows a web browser window with the title 'Add Course' and the URL 'http://localhost:8081/FinalProject/admin/course'. The page has a yellow background and is titled 'Add Course For Users'. It contains four input fields for 'Course Name', 'Course Description', 'Course Fees', and 'Course Resource'. Below these fields is a 'Submit' button. At the bottom of the page, there are two links: 'Home' and 'Log Out'.

Course Name

Course Description

Course Fees

Course Resource

[Home](#) [Log Out](#)

AddCourseServlet

Call Course
Class

ADD
course in
Database

Course (Class)

Success
Addition

View
Course
After
Addition

ViewCourseServlet

ENROLL COURSE



Enroll Course

COURSE ID	COURSE NAME	COURSE DESCRIPTION	COURSE FEES	ENROLL
101	JAVA	Programing language	299	<input type="button" value="Enroll"/>
102	C Programming	Programming language	199	<input type="button" value="Enroll"/>
120	python	Very easy	4000	<input type="button" value="Enroll"/>

[Home](#) [Log Out](#)

My Course

ENROLL ID	COURSE ID	COURSE NAME	COURSE DESCRIPTION	COURSE FEES	COURSE RESOURCE
1021	102	C Programming	Programming language	199	cprogram.mp4
1022	120	python	Very easy	4000	techadmey

[Home](#) [Log Out](#)

ADD FEEDBACK



A screenshot of a web browser window titled "Feedback". The address bar shows the URL "http://localhost:8081/FinalProject/user/feedback". The page has a yellow background and is titled "User Feedback Form". In the center, there is a text input field with the placeholder text "Enter Your Feedback Here...". Below the input field is a "Submit" button. At the bottom of the page, there are two links: "[Home](#)" and "[Log Out](#)".

ADD ADMIN

Admin Add
new Admin

AdminRegisterServlet

User Object Creation
Insert Into Database

Admin
// Class

Successful Registration
Create New Session

AdminServlet

The screenshot shows a web browser window titled "Admin Registration" with the URL "http://localhost:8081/FinalProject/admin/add". The page has a yellow background and is titled "Admin Registration Form". It contains three input fields for "Name", "Email", and "Password", each with a label to its left. Below these fields is a "Submit" button. At the bottom of the form, there are two links: "Home" and "Log Out".

Admin Registration Form

Name

Email

Password

[Home](#) [Log Out](#)

OTHER FUNCTIONS

Log Out

- `request.getSession().invalidate();`

View FeedBack

- ViewFeedback Servlet
 - `Select * from feedback;`

View User

- ViewUserServlet
 - `Select * from user1;`

View Admin

- ViewAdminServlet
 - `Select * from admin1;`

View Course

- ViewCourseServlet
 - `Select * from course;`

MISCELLANEOUS

ENVIRONMENT

- IDE - Eclipse
- Git - For Version Control
- Github - Source Code Repo
<https://github.com/abhishek-singh-IN/Amdocs-Training-Project>

SOURCE CODE

- Session - For Storing logged in User Details
- Junit 5 - Testing

PL/SQL FUNCTIONS

SUBMIT FEEDBACK

```
CREATE OR REPLACE FUNCTION submitFeedback(  
    p_user_id IN feedback.USER_ID%TYPE,  
    p_feedback_message IN feedback.FEEDBACK_MESSAGE%TYPE)  
RETURN FEEDBACK.F_ID%TYPE IS  
    p_f_id FEEDBACK.F_ID%TYPE;  
BEGIN  
    p_f_id := f_id.NEXTVAL;  
    INSERT INTO feedback("USER_ID","F_ID","FEEDBACK_MESSAGE")  
    VALUES (p_user_id,p_f_id,p_feedback_message);  
    COMMIT;  
    RETURN p_f_id;  
END;
```

ADD COURSE

```
CREATE OR REPLACE FUNCTION addCOURSE(  
    p_course_name IN COURSE.C_NAME%TYPE,  
    p_course_desp IN COURSE.C_DESP%TYPE,  
    p_course_fees IN COURSE.C_FEES%TYPE,  
    p_course_resource IN COURSE.C_RESOURCE%TYPE)  
RETURN COURSE.course_id%TYPE IS  
    p_course_id COURSE.course_id%TYPE;  
BEGIN  
    p_course_id := course_id.nextval;  
    INSERT INTO COURSE ("COURSE_ID","C_NAME","C_DESP","C_FEES","C_RESOURCE")  
    VALUES (p_course_id,p_course_name,p_course_desp,p_course_fees,p_course_resource);  
    COMMIT;  
    RETURN p_course_id;  
END;
```

ENROLL COURSE

```
CREATE OR REPLACE FUNCTION enrollCOURSE(  
    p_user_id IN ENROLL.enroll_id%TYPE,  
    p_course_id IN ENROLL.course_id%TYPE)  
RETURN ENROLL.ENROLL_ID%TYPE IS  
    p_enroll_id ENROLL.ENROLL_ID%TYPE;  
BEGIN  
    p_enroll_id := enroll_id.nextval;  
    INSERT INTO ENROLL ("ENROLL_ID", "USER_ID", "COURSE_ID")  
    VALUES (p_enroll_id, p_user_id, p_course_id);  
    COMMIT;  
    RETURN p_enroll_id;  
END;
```

INSERT USER

```
CREATE OR REPLACE FUNCTION submitFeedback(  
  p_user_id IN feedback.USER_ID%TYPE,  
  p_feedback_message IN  
    feedback.FEEDBACK_MESSAGE%TYPE)  
RETURN FEEDBACK.F_ID%TYPE IS  
  p_f_id FEEDBACK.F_ID%TYPE;  
BEGIN  
  p_f_id := f_id.NEXTVAL;  
  INSERT INTO  
    feedback("USER_ID","F_ID","FEEDBACK_MESSAGE")  
VALUES (p_user_id,p_f_id,p_feedback_message);  
COMMIT;  
RETURN p_f_id;  
END;
```

INSERT ADMIN

```
CREATE OR REPLACE FUNCTION insertADMIN1(  
  p_name IN ADMIN1.name%TYPE,  
  p_email IN ADMIN1.email%TYPE,  
  p_password IN ADMIN1.password%TYPE)  
RETURN ADMIN1.ADMIN_ID%TYPE IS  
  p_userid ADMIN1.ADMIN_ID%TYPE;  
BEGIN  
  p_userid := userid.nextval;  
  INSERT INTO ADMIN1 ("ADMIN_ID",  
    "NAME","EMAIL", "PASSWORD")  
VALUES (p_userid, p_name, p_email,p_password);  
COMMIT;  
RETURN p_userid;  
END;
```

USER AND ADMIN LIST

View Feedback

http://localhost:8081/FinalProject/user/admin

List of Admins

ADMIN ID	ADMIN NAME	E-MAIL
107	Himanshu	himanshu@google.com
108	Aastha	aasthap@amdocs.com

[Home](#) [Log Out](#)

View Feedback

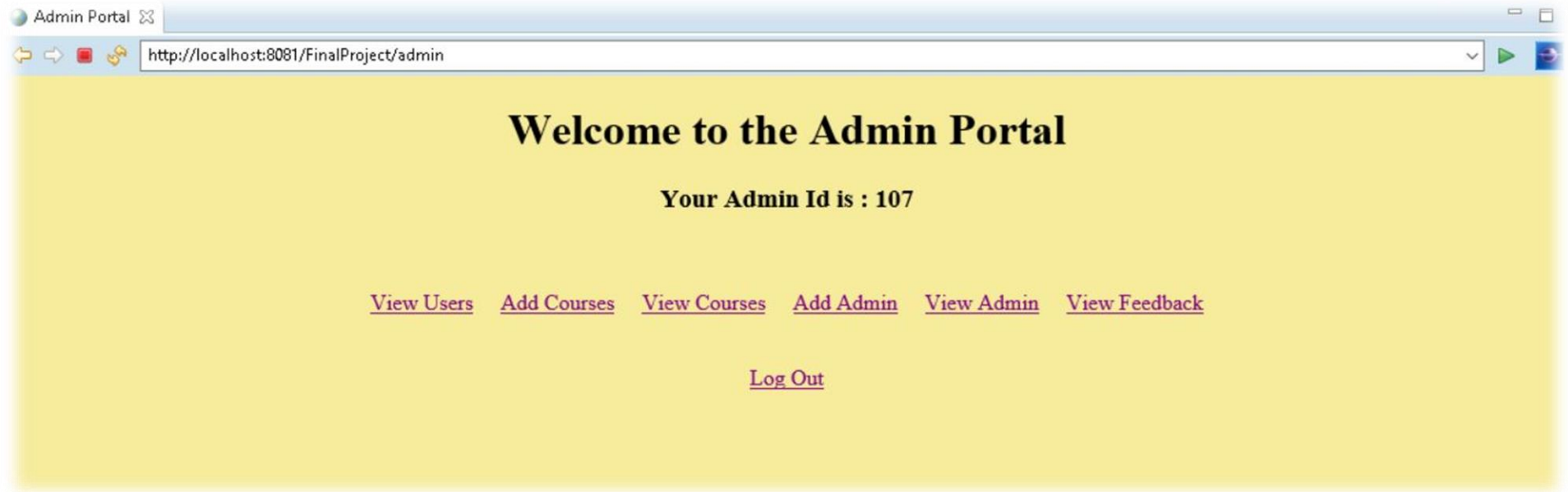
http://localhost:8081/FinalProject/admin/user

List of Users

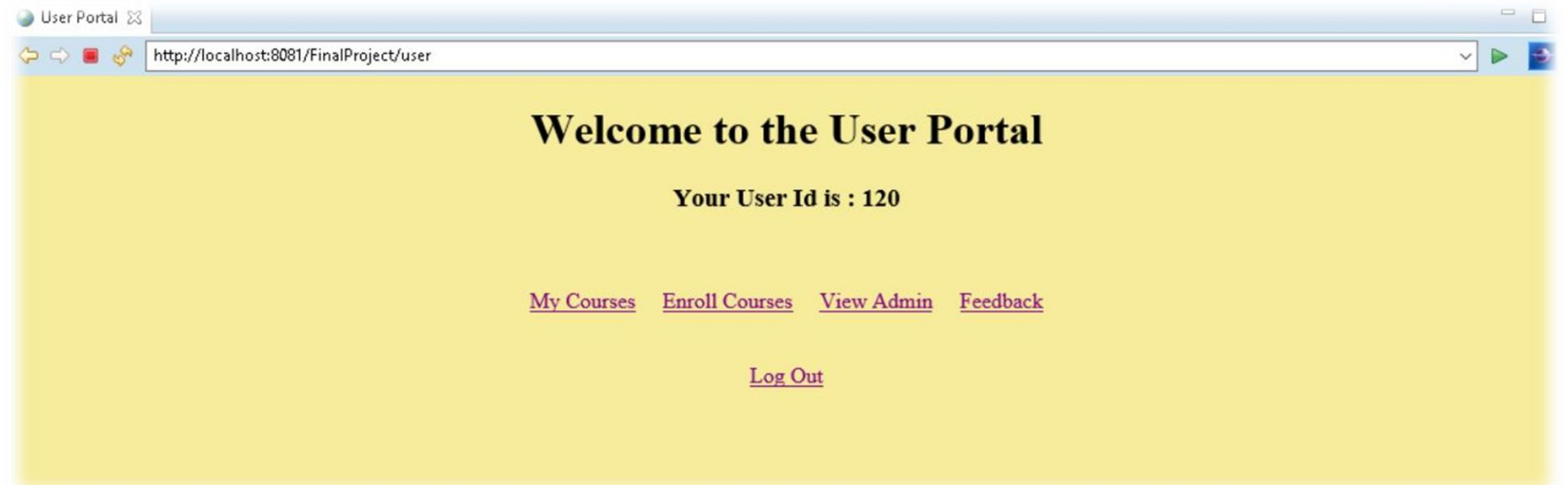
USER ID	NAME	PHONE NO	E-MAIL	ADDRESS	REGISTRATION DATE	PHOTO
100	Ankit	4578008234	ankit@gmail.com	noida	2021-01-03 00:00:00.0	img.jpg
104	Rahul	9586234712	rahul@gmail.com	banglore	2021-03-03 00:00:00.0	img1.jpg
105	gouthami	8524379162	gouthami@gmail.com	anantapur	2021-02-03 00:00:00.0	img2.jpg
106	Abhishek	8860110307	abhishek@singhabhishek.tech	Delhi	2021-07-25 00:00:00.0	abhishek.jpg
120	Deshna	9855599431	deshna@microsoft.com	Pakistan	2021-07-26 00:00:00.0	idiot.jpg

[Home](#) [Log Out](#)

ADMIN



USER



FEEDBACK

View Feedback

http://localhost:8081/FinalProject/admin/feedback

FEEDBACK FROM USER

USER ID	FEEDBACK ID	FEEDBACK MESSAGE
100	102	good
100	104	good
105	120	null
105	121	null
105	122	Best Work
120	123	Awesome
120	124	Amazing
106	140	Good Work

[Home](#) [Log Out](#)

COURSE

Enroll Course

http://localhost:8081/FinalProject/admin/view-course

List of Courses for Users

COURSE ID	COURSE NAME	COURSE DESCRIPTION	COURSE FEES
101	JAVA	Programing language	299
102	C Programming	Programming language	199
120	python	Very easy	4000

[Home](#) [Log Out](#)

**THANK
YOU**

