



HALDIA INSTITUTE OF TECHNOLOGY

हल्दिया प्रौद्योगिकी संस्थान

হলদিয়া ইনস্টিটিউট অফ টেকনোলজি

Approved By : AICTE ● Affiliated To : MAKAUT (WBUT)

AUTOMATED STUDENT REGISTRATION SYSTEM

A Project Report for Industrial Training

Submitted by

UTKARSH VERMA (16/CSE/021)

SRIKANT KUMAR(16/CSE/031)

SOURAV KUMAR RAJAK(16/CSE/035)

SAURAV KUMAR (16/CSE/045)

SATYAM RAJ(16/CSE/046)

TABLE OF CONTENT

Content	Page no
● Acknowledgement	3
● Project Scope	4
● Requirement Specification	
➤ Functional Requirement	5-6
➤ Hardware Requirement	6
➤ Software Requirement	7
● Database Design	8-10
● Screen Shots	11-23
● Application Workflow	24-34
● Future Scope of Improvements	35
● Conclusion	36
● Appendices	37
● References	38

ACKNOWLEDGEMENT

I take this opportunity to express my deep gratitude and sincerest thanks to our project mentor, Mr. Ratan Kumar for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

The assignment cannot be completed without the effort and co-operation from our group members. Last but not the least, we would like to express our gratitude to our friends and respondents for the support and willingness to spend some times with us to fill in the questionnaires.

PROJECT SCOPE

Project “Automated Student Registration System” while doing this Project we faced lots of challenges but our ultimate goal was to satisfy all functionality demanded by stakeholder and make the project user friendly such that non-technical User easily understand and access. We have also 15 days’ time deadlines to complete this project and we complete project within this deadline. We deal this project on basis of our three types of Users Admin, Student, Clerk. All these users have certain responsibility and limitations or accessibility which we have to implements.

REQUIREMENT SPECIFICATION

❖ Functional Requirement

Function id	Requirement statement	Must/Optional	Comment
FR01	This web application must require a Home Page whether the user visiting is new one or existing one	Must	This Home Page requirement is must
FR02	This web app required a login where three kind user may access which have different accessibility or restriction to access and manipulation of data	Must	
FR03	Different Home page for Student, Admin and Clerk so that they easily access to their information's they required	Must	
FR04	In the case of database design we required 5 Tables mention below to store the details in our database so that our user can easily	Must	

	access those details when they required		
FR05	We have many functionalities for all our 3 users .Student can registered to any courses available ,Clerk can access student ,Admin can access all the details they can even manipulate all the courses available	Must	

❖ Hardware Requirement

Client Machine		Server Machine	
HDD	200 MB	HDD	320 GB
Processor	Dual Core or newer processor	Processor	Dual Core or newer processor
Memory	1 GB	Memory	4 GB

❖ Software Requirement

➤ **Front End**

- Html
- Cascading Style Sheet (CSS)
- Java Script

➤ **Back End**

- Java
- Java Database Connectivity (JDBC)
- Java Server Pages (JSP)
- Servlet

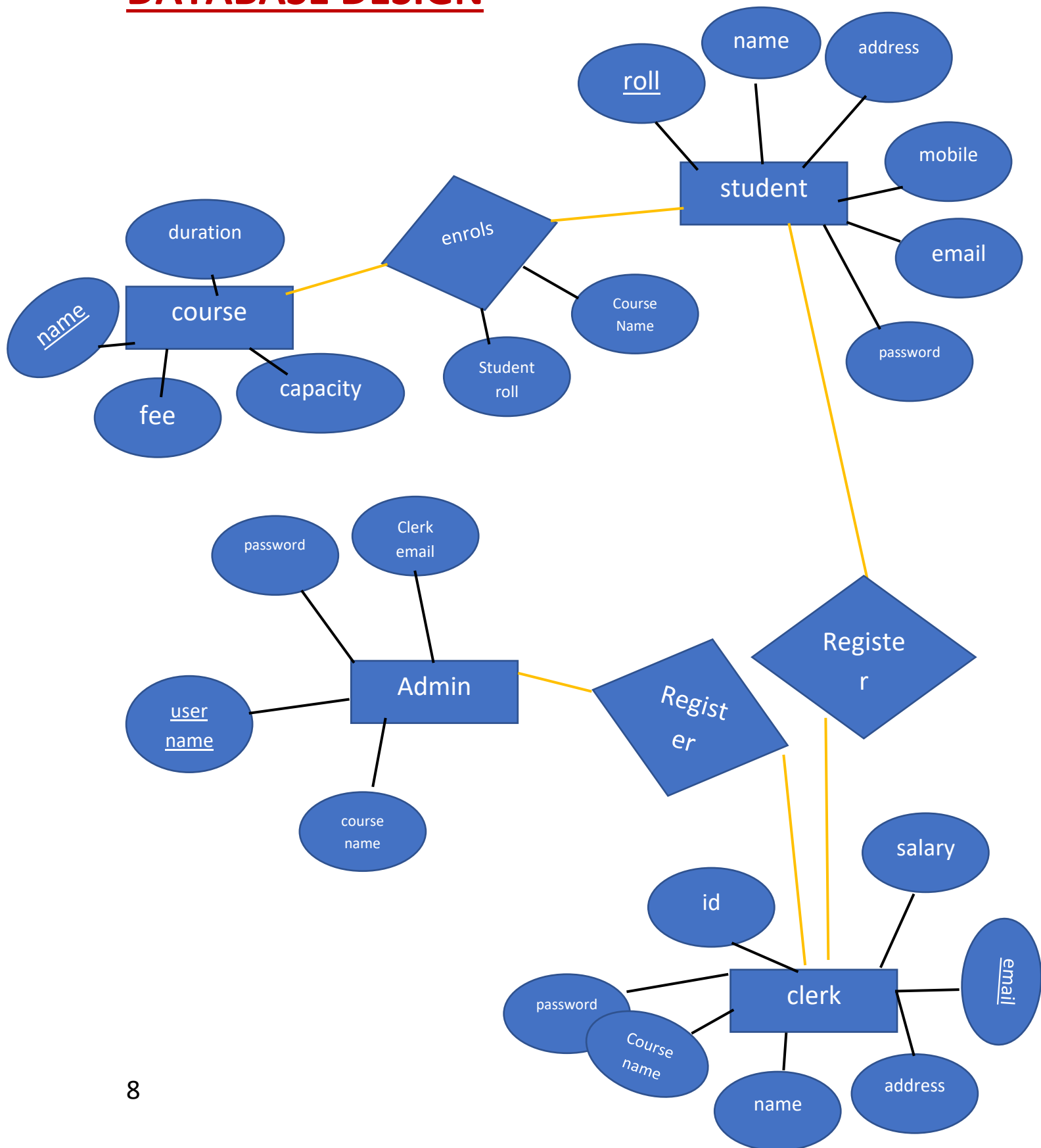
➤ **Application Server**

- Apache Tomcat

➤ **Database Server**

- MySQL

DATABASE DESIGN



✓ TABLE DESIGN

➤ Table: Clerk

Field	Type	Null	Key	Default
Id	Varchar (200)	NO	PRI	NULL
Name	Varchar (20)	YES		NULL
Age	Varchar (20)	YES		NULL
Address	Varchar (20)	YES		NULL
Salary	Varchar (20)	YES		NULL
Email	Varchar (20)	YES		NULL
password	Varchar (20)	YES		NULL

➤ Table: Course

Field	Type	Null	Key	Default	Extra
courseName	varchar(12)	NO	PRI	NULL	
courseDuration	varchar(12)	YES		NULL	
courseFee	int(11)	YES		NULL	
courseStartDate	varchar(12)	YES		NULL	
Capacity	int(11)	YES		NULL	
Avail	int(11)	YES		NULL	

➤ Table: Student

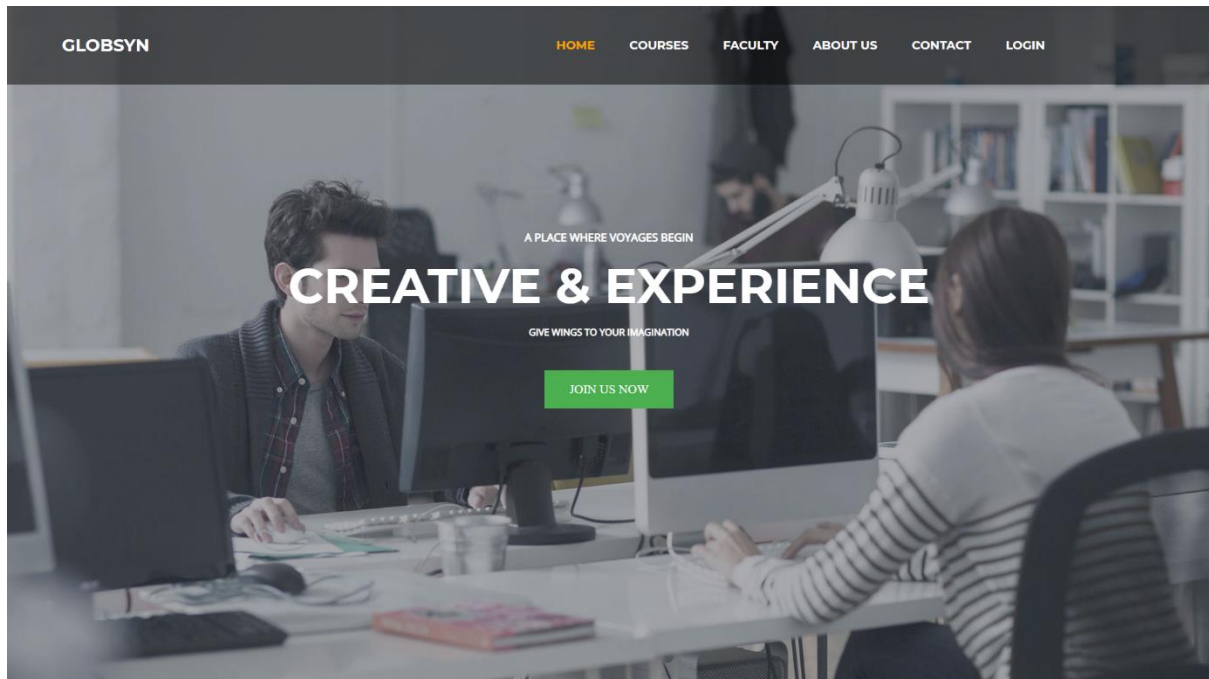
Field	Type	Null	Key	Default
Roll	Int (11)	NO	PRI	NULL
Name	Varchar (20)	YES		NULL
Address	Varchar (20)	YES		NULL
Mobile	Varchar (20)	YES		NULL
Email	Varchar (20)	YES		NULL
Password	Varchar (20)	YES		NULL

➤ **Table: Student Course**

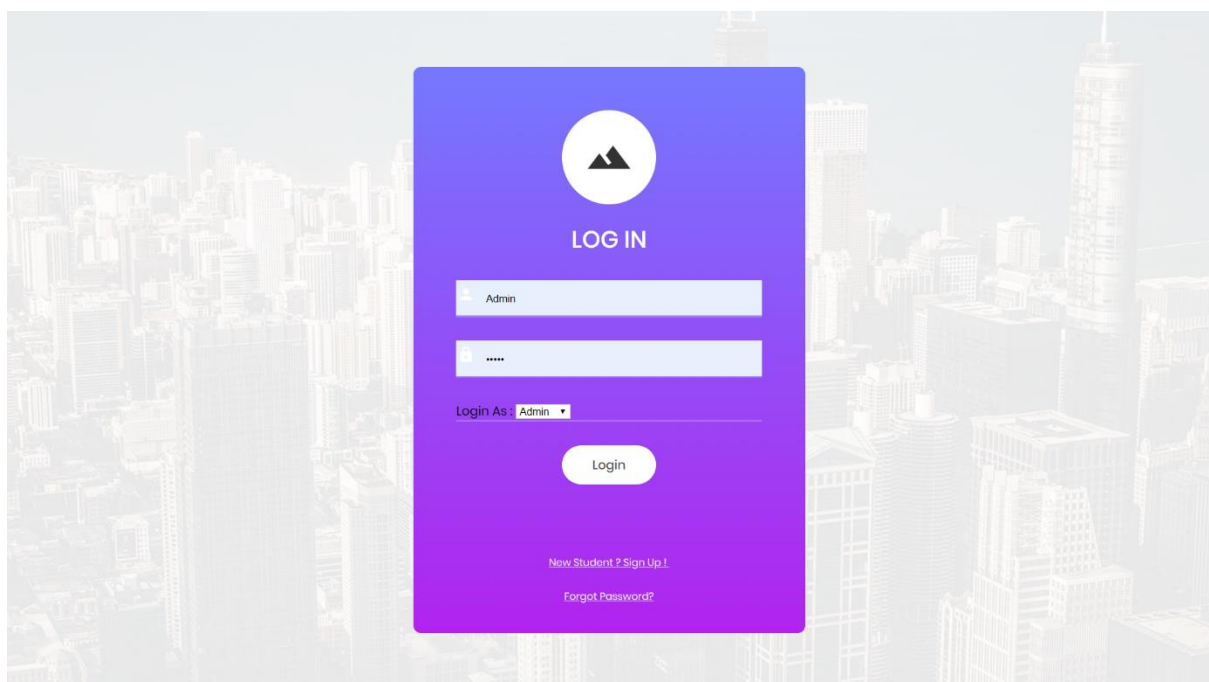
Field	Type	Null	Key	Default
Roll	Int (11)	YES		NULL
courseName	Varchar (20)	YES		NULL

SCREEN SHOTS

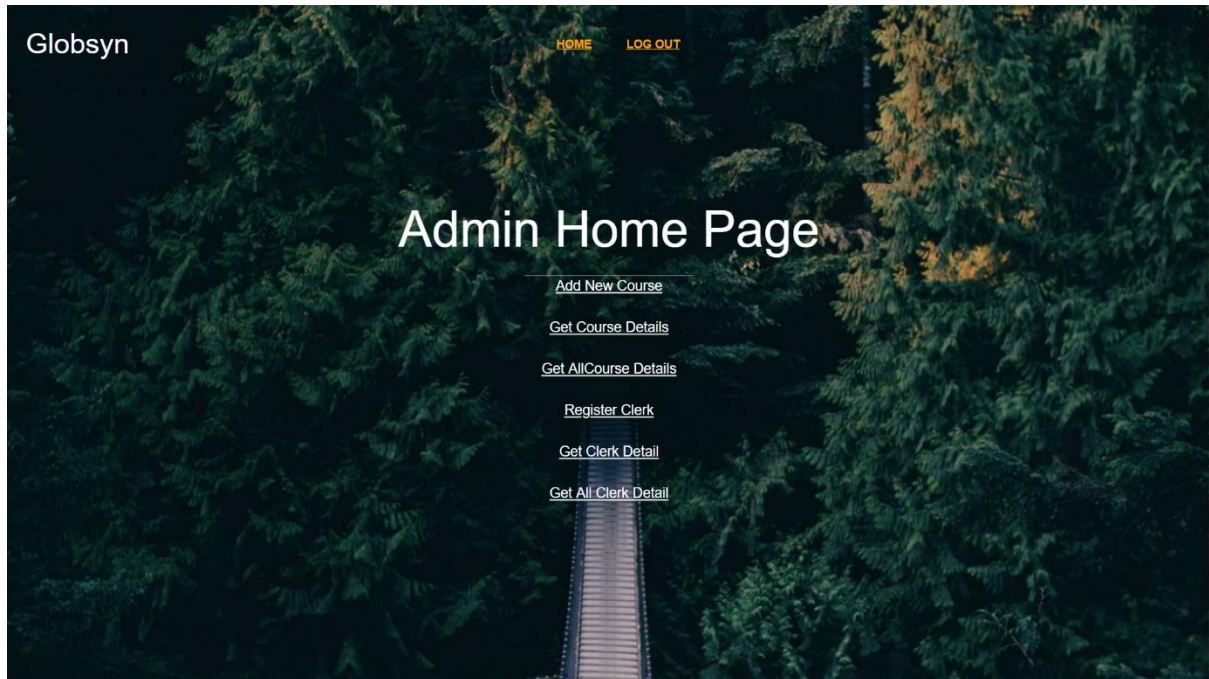
Home Page: index.jsp



Login Page: loginPage.jsp



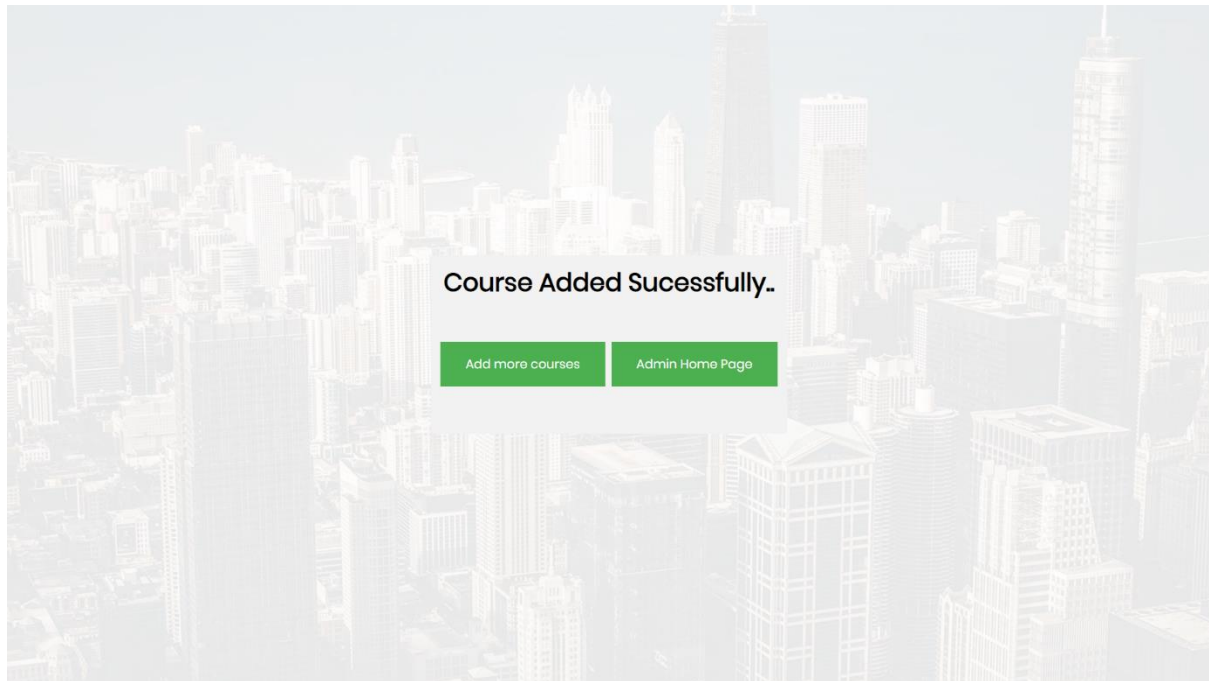
Admin Home: AdminHome.jsp



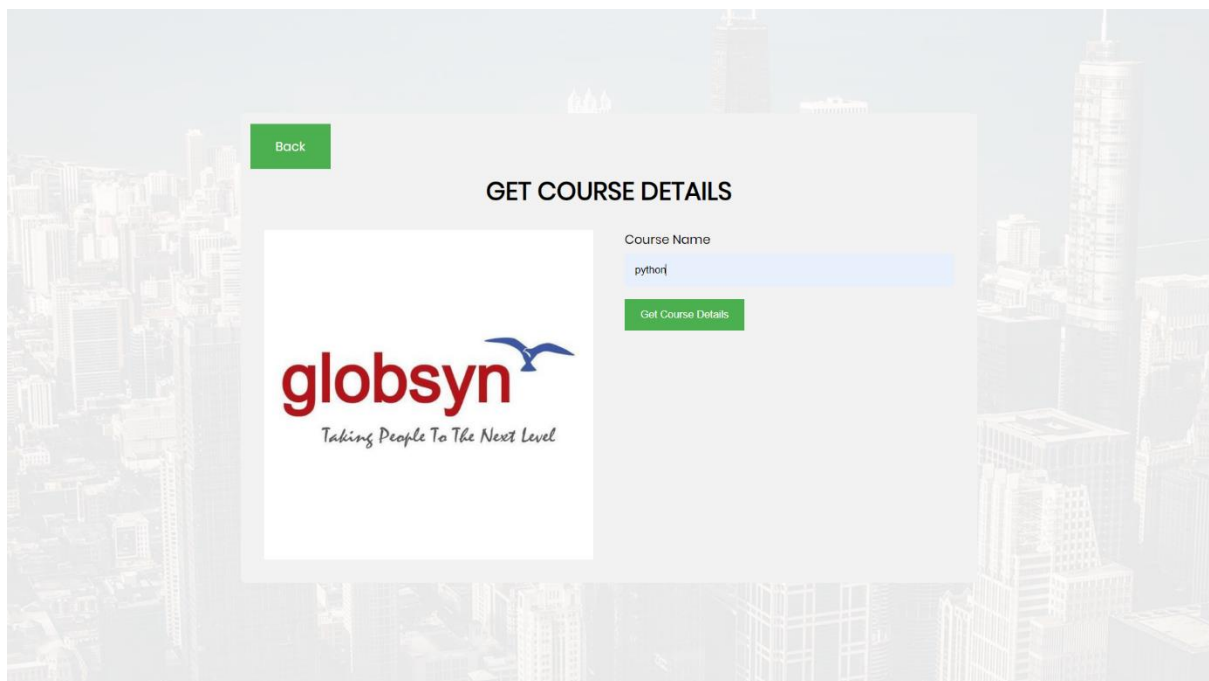
Add New Course: addCourse.jsp

The screenshot displays the 'ADD COURSES' form, which is a light gray rectangular box centered on a background of a city skyline. In the top left corner of the form is a green 'Back' button. The title 'ADD COURSES' is centered at the top of the form. On the left side of the form is the Globsyn logo, which consists of the word 'globsyn' in red lowercase letters with a blue bird icon above the 'y', and the tagline 'Taking People To The Next Level' in a smaller, italicized font below it. To the right of the logo are five input fields: 'Course Name', 'Course Duration', 'Course Fee', 'Start Date' (with a small '>' icon and a placeholder 'dd-mm-yyyy'), and 'Course Capacity'. A green 'Submit' button is located at the bottom right of the form.

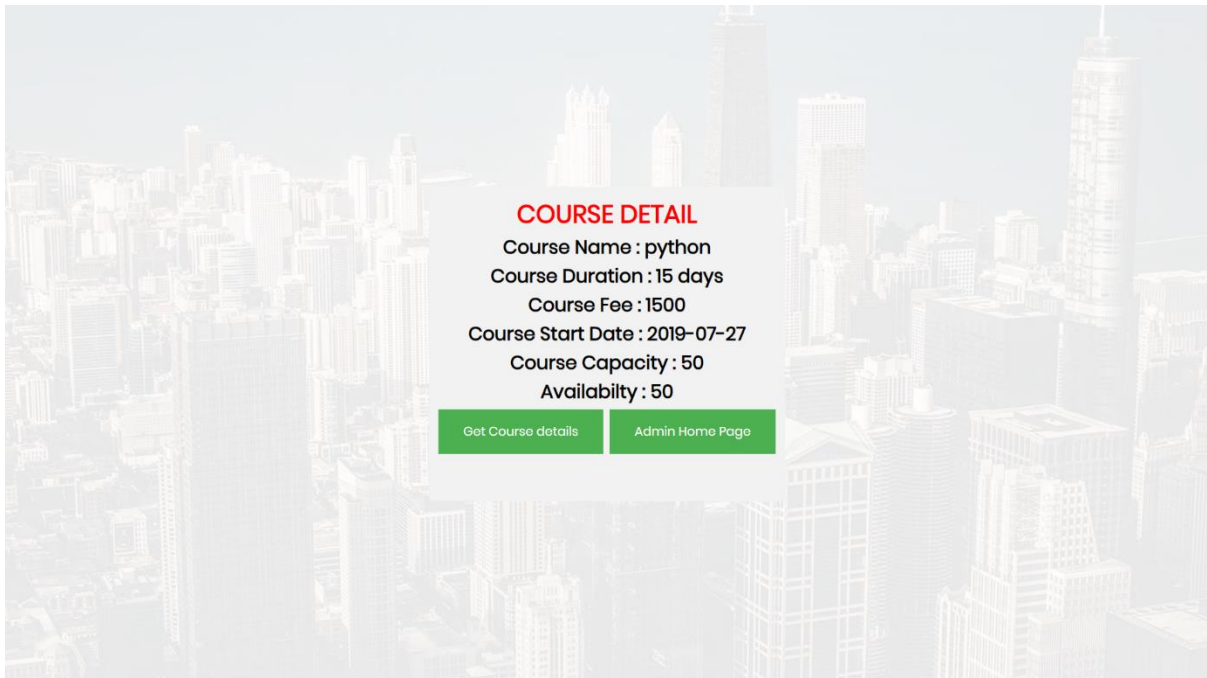
Couse Update Success: courseSuccess.jsp



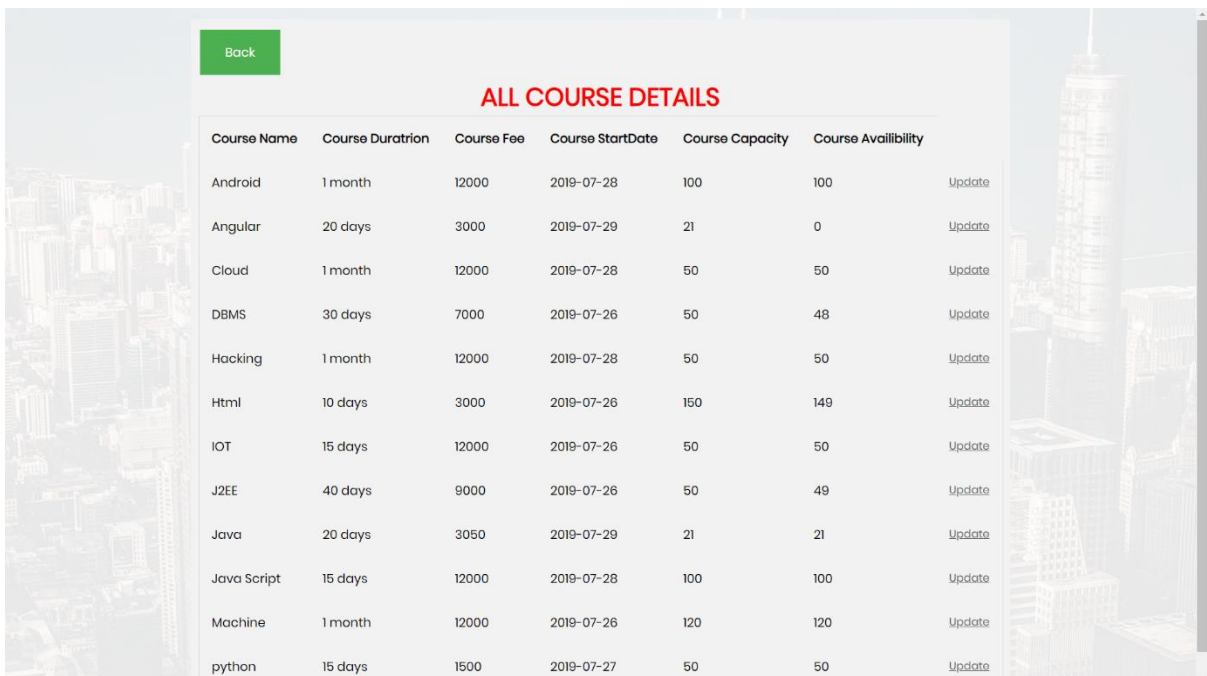
Get Course Details: getCourse.jsp



Course Details: courseDetail.jsp

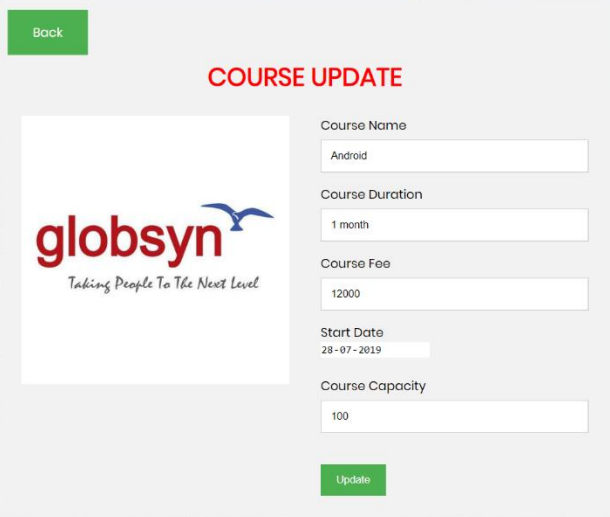
A screenshot of the 'courseDetail.jsp' page. The background is a faded image of a city skyline. In the center, there is a white box containing the following text: 'COURSE DETAIL' in red, followed by 'Course Name : python', 'Course Duration : 15 days', 'Course Fee : 1500', 'Course Start Date : 2019-07-27', 'Course Capacity : 50', and 'Availability : 50'. At the bottom of this box are two green buttons: 'Get Course details' and 'Admin Home Page'.

All Course Detail: allCourse.jsp

A screenshot of the 'allCourse.jsp' page. It features a 'Back' button in the top left corner. The main content is a table titled 'ALL COURSE DETAILS' in red. The table has six columns: 'Course Name', 'Course Duration', 'Course Fee', 'Course StartDate', 'Course Capacity', and 'Course Availability'. It lists 14 courses with their respective details. Each row has an 'Update' link in the 'Course Availability' column.

Course Name	Course Duration	Course Fee	Course StartDate	Course Capacity	Course Availability
Android	1 month	12000	2019-07-28	100	100 Update
Angular	20 days	3000	2019-07-29	21	0 Update
Cloud	1 month	12000	2019-07-28	50	50 Update
DBMS	30 days	7000	2019-07-26	50	48 Update
Hacking	1 month	12000	2019-07-28	50	50 Update
Html	10 days	3000	2019-07-26	150	149 Update
IOT	15 days	12000	2019-07-26	50	50 Update
J2EE	40 days	9000	2019-07-26	50	49 Update
Java	20 days	3050	2019-07-29	21	21 Update
Java Script	15 days	12000	2019-07-28	100	100 Update
Machine	1 month	12000	2019-07-26	120	120 Update
python	15 days	1500	2019-07-27	50	50 Update

Course Update: courseUpdate.jsp



The image shows a web form titled "COURSE UPDATE" overlaid on a faded cityscape background. The form includes a "Back" button in the top left. On the left side of the form is the "globsyn" logo with the tagline "Taking People To The Next Level". On the right side, there are five input fields: "Course Name" (containing "Android"), "Course Duration" (containing "1 month"), "Course Fee" (containing "12000"), "Start Date" (containing "28-07-2019"), and "Course Capacity" (containing "100"). An "Update" button is located at the bottom right of the form.

Back

COURSE UPDATE

globsyn
Taking People To The Next Level

Course Name
Android

Course Duration
1 month

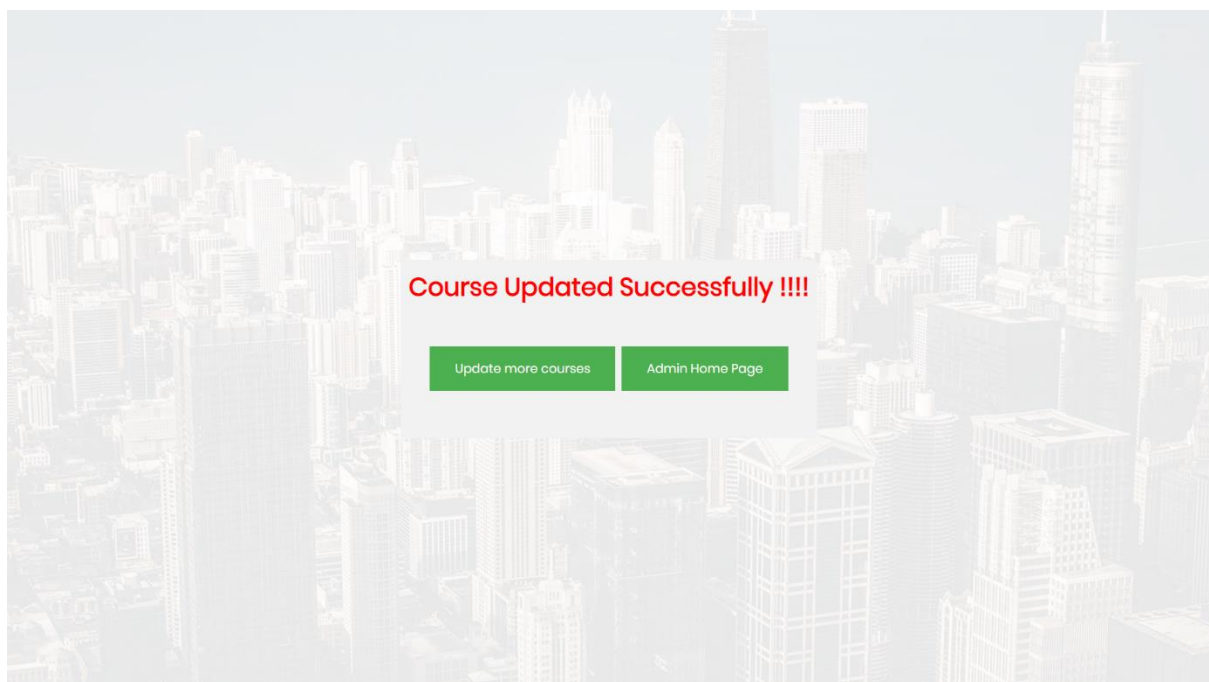
Course Fee
12000

Start Date
28-07-2019

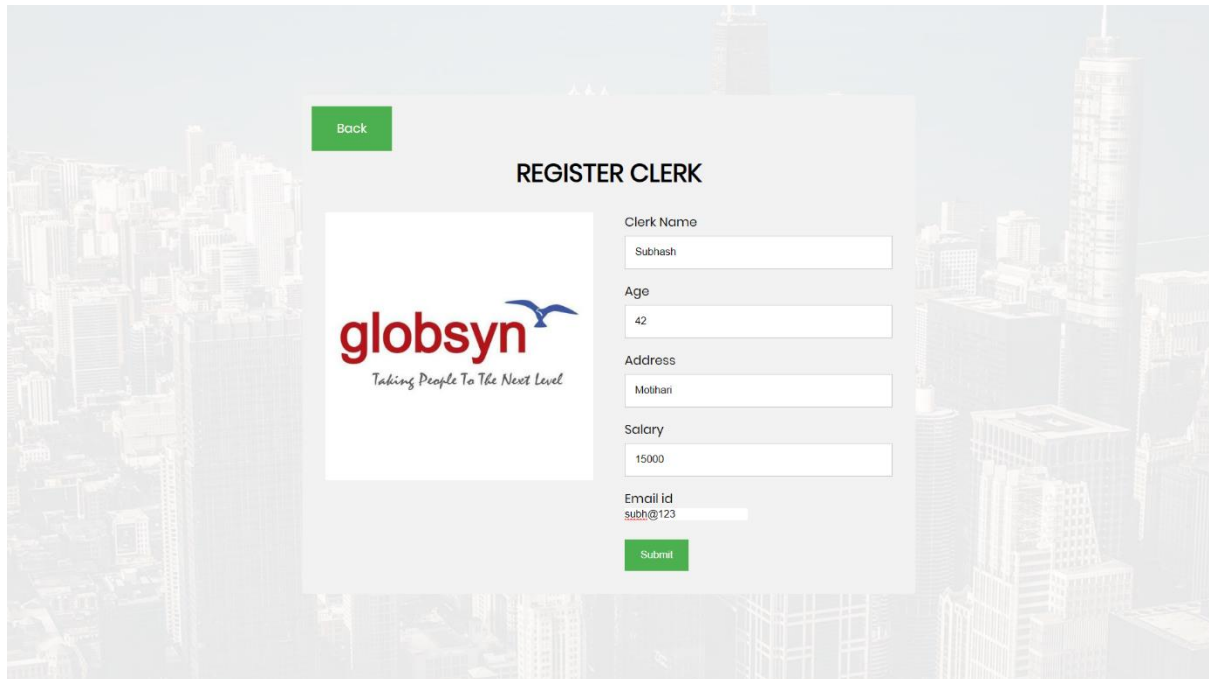
Course Capacity
100

Update

Course Updated Successful: updateCourseSucess.jsp

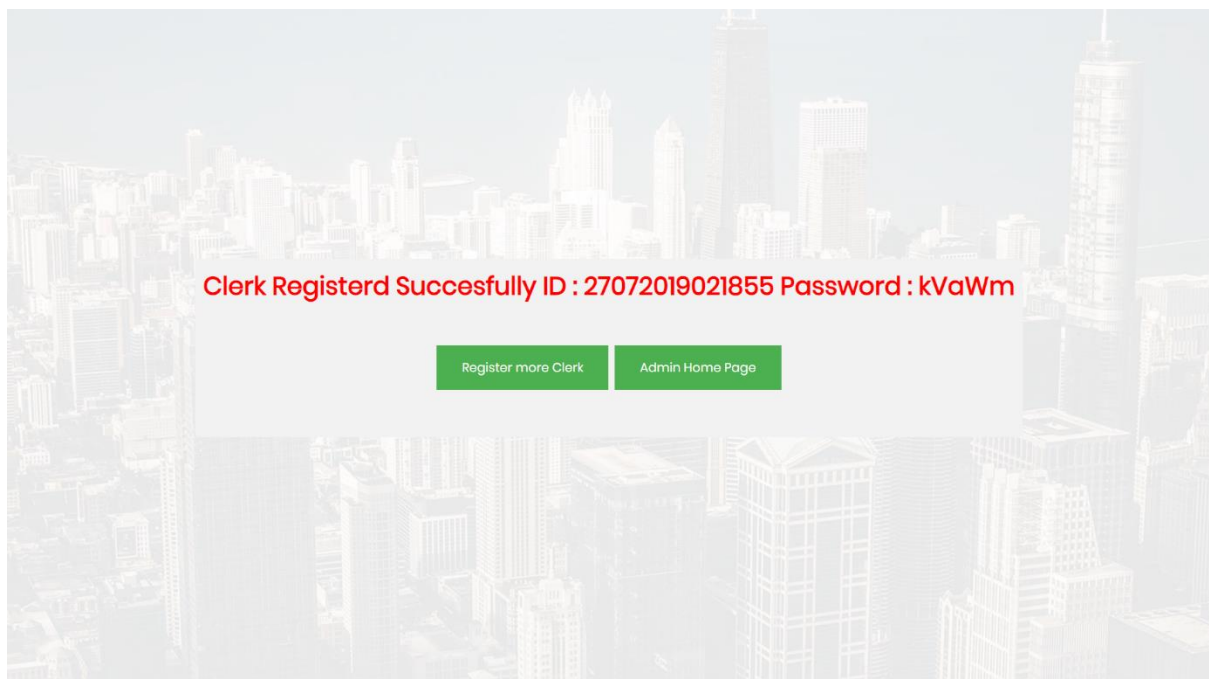


Register Clerk: ClerkRegister.jsp

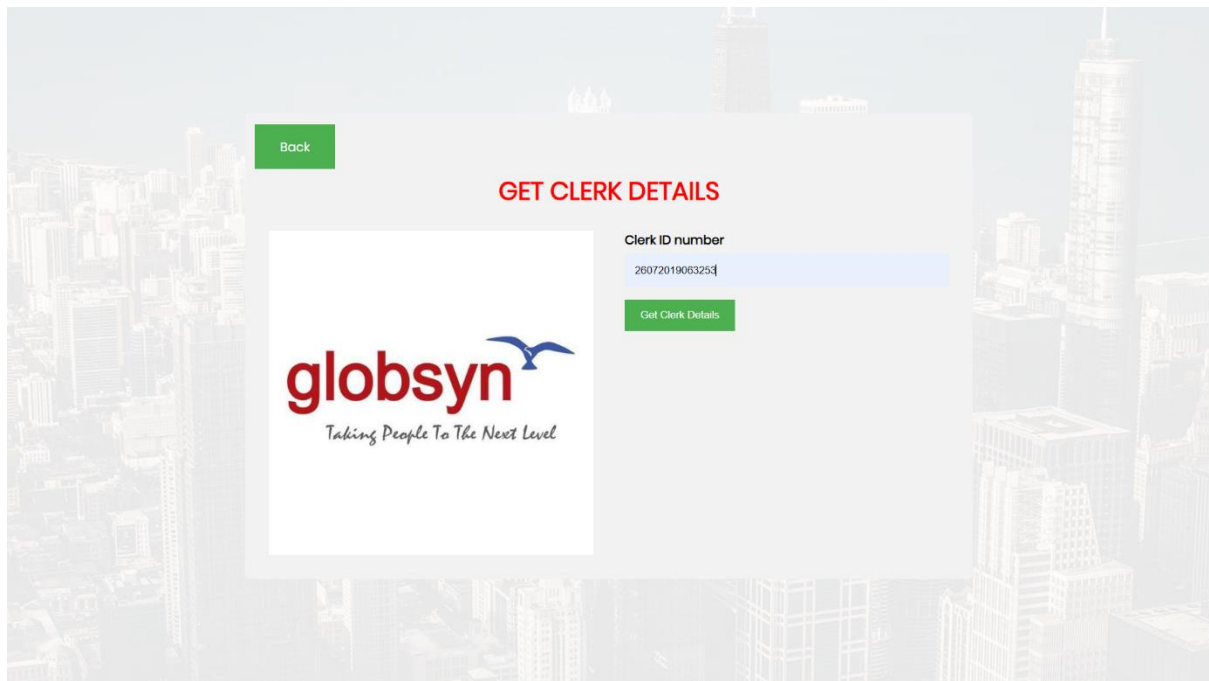


The screenshot shows a web form titled "REGISTER CLERK" overlaid on a city skyline background. The form includes a "Back" button in the top left. On the left side of the form is the "globsyn" logo with the tagline "Taking People To The Next Level". The right side contains several input fields: "Clerk Name" (filled with "Subhash"), "Age" (filled with "42"), "Address" (filled with "Mothari"), "Salary" (filled with "15000"), and "Email id" (filled with "subh@123"). A green "Submit" button is located at the bottom right of the form.

Clerk Registration success: clerkRegSuccess.jsp



Get Clerk Detail: getClerk.jsp



Back

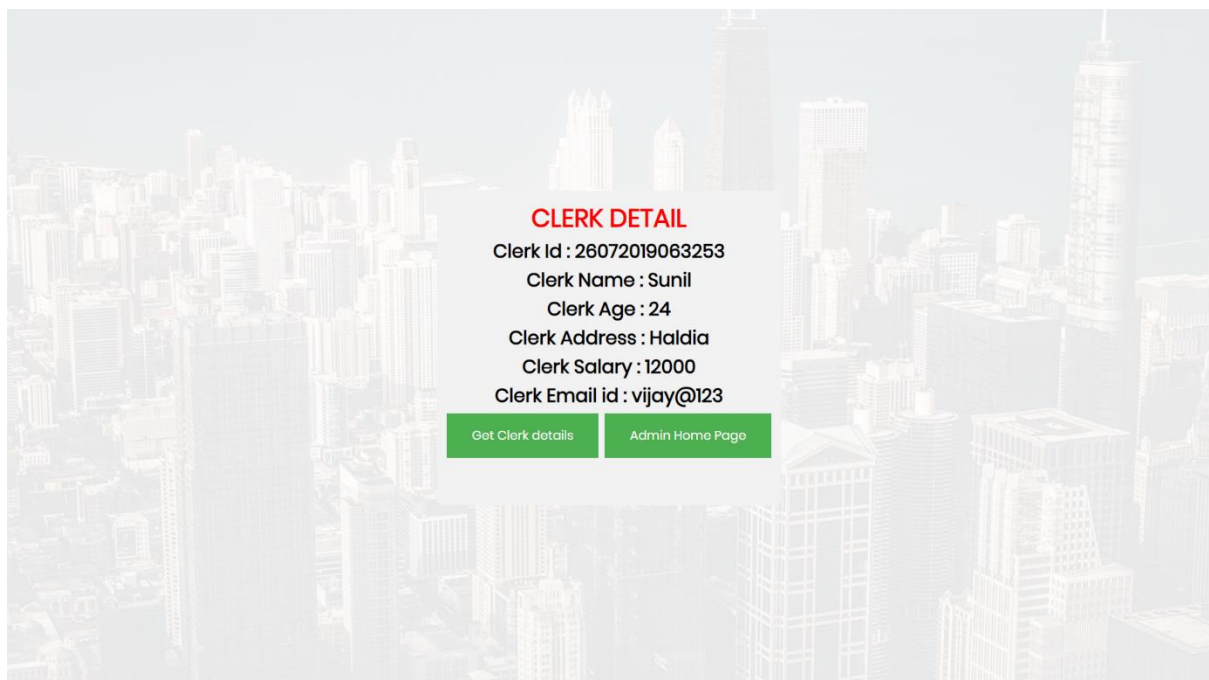
GET CLERK DETAILS

globsyn
Taking People To The Next Level

Clerk ID number
26072019063253

Get Clerk Details

Clerk Detail page: clerkDetails.jsp

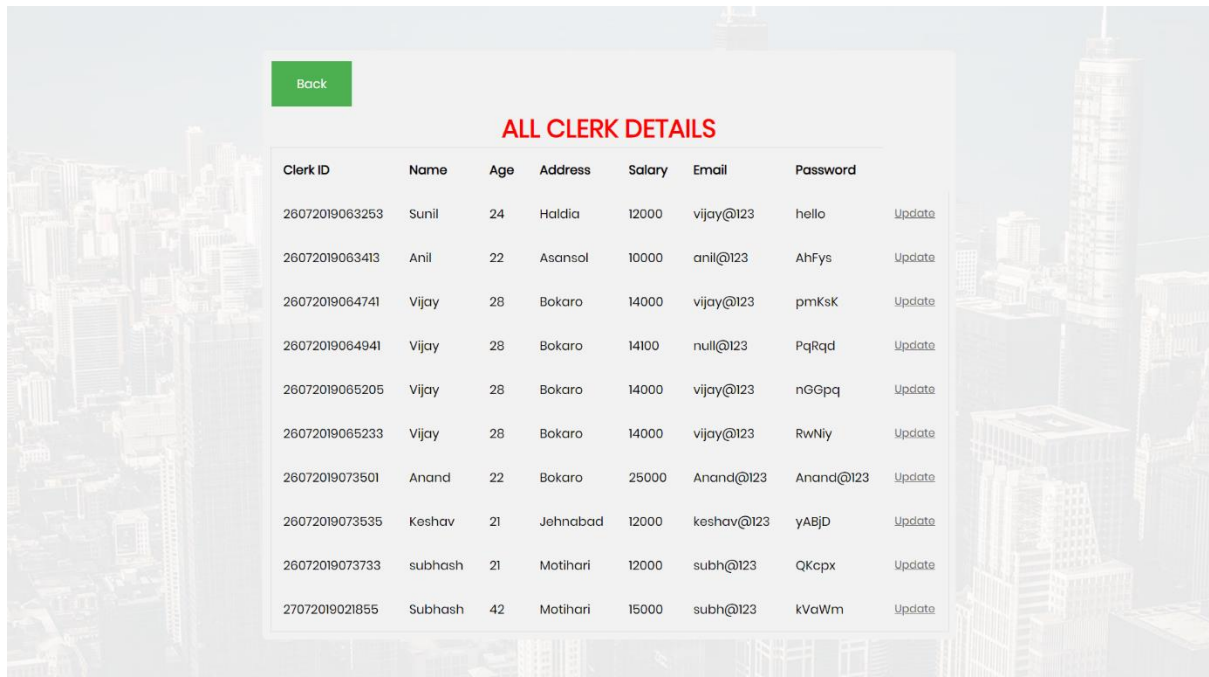


CLERK DETAIL

Clerk Id : 26072019063253
Clerk Name : Sunil
Clerk Age : 24
Clerk Address : Haldia
Clerk Salary : 12000
Clerk Email id : vijay@123

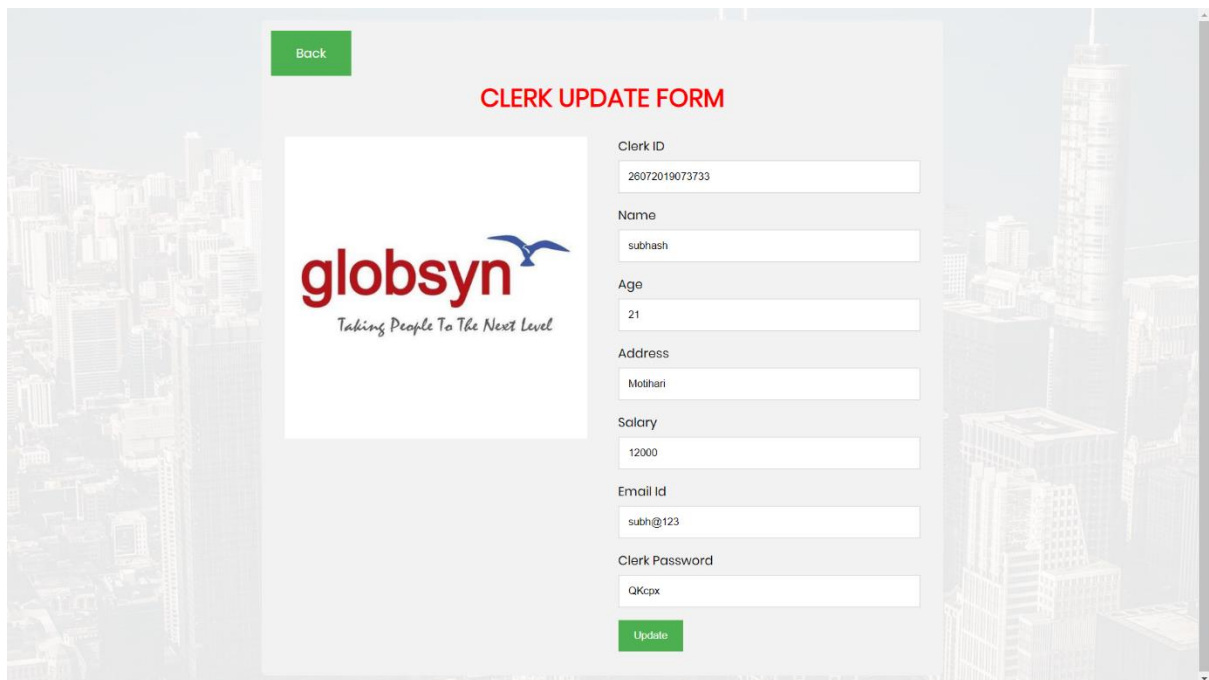
Get Clerk details Admin Home Page

Get All Clerk Detail: allClerk.jsp



Clerk ID	Name	Age	Address	Salary	Email	Password	
26072019063253	Sunil	24	Haldia	12000	vijay@123	hello	Update
26072019063413	Anil	22	Asansol	10000	anil@123	AhFys	Update
26072019064741	Vijay	28	Bokaro	14000	vijay@123	pmKsK	Update
26072019064941	Vijay	28	Bokaro	14100	null@123	PqRqd	Update
26072019065205	Vijay	28	Bokaro	14000	vijay@123	nGGpq	Update
26072019065233	Vijay	28	Bokaro	14000	vijay@123	RwNiy	Update
26072019073501	Anand	22	Bokaro	25000	Anand@123	Anand@123	Update
26072019073535	Keshav	21	Jehnabad	12000	keshav@123	yABjD	Update
26072019073733	subhash	21	Motihari	12000	subh@123	QKcpx	Update
27072019021855	Subhash	42	Motihari	15000	subh@123	kVaWm	Update

Clerk Update Form: clerkUpdate.jsp



Back

CLERK UPDATE FORM

globsyn
Taking People To The Next Level

Clerk ID
26072019073733

Name
subhash

Age
21

Address
Motihari

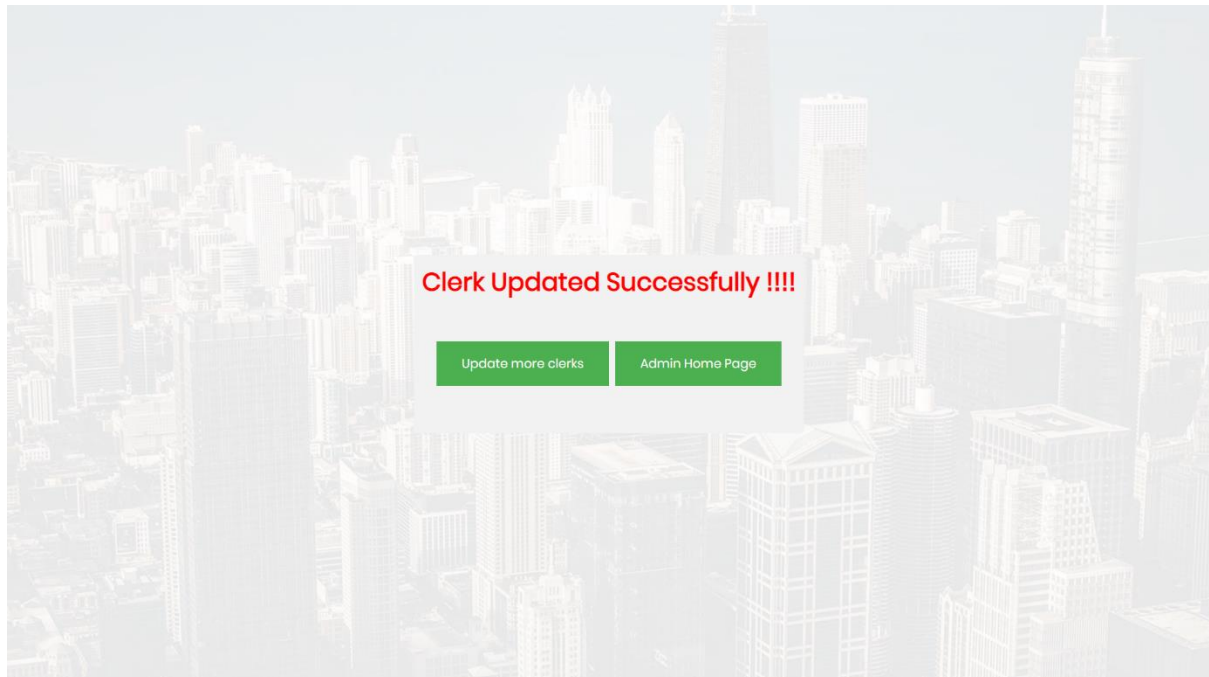
Salary
12000

Email Id
subh@123

Clerk Password
QKcpx

Update

Clerk Details updated Successfully: updateClerkSucess.jsp



studentRegister.jsp

Registration open

Register

Name:

Address:

Mobile:

Email:

Password:

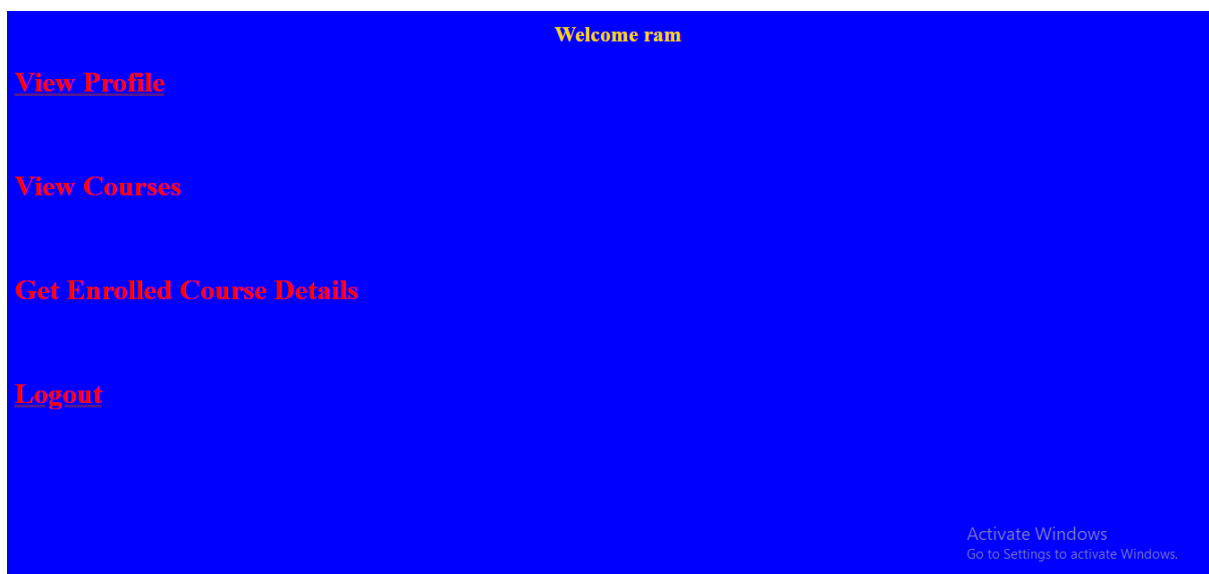
[Back](#)

Activate Windows
Go to Settings to activate Windows.



regSuccess.jsp(student registered successfully)

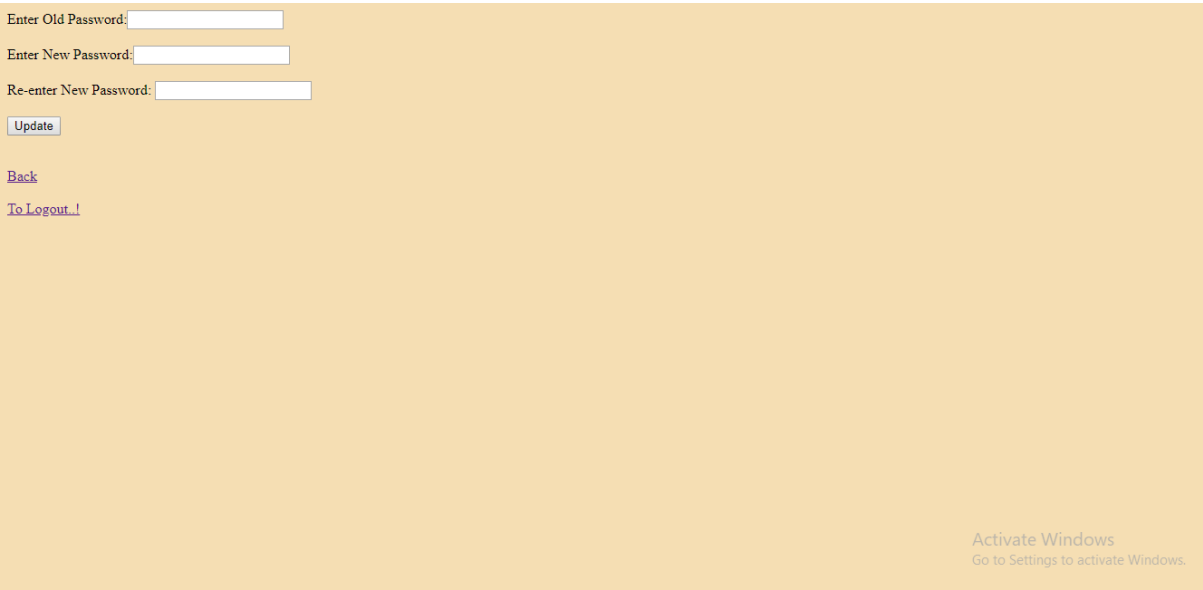
studentHome.jsp



clerkHome.jsp



updatePassword.jsp



allStudentdetails.jsp

[Continue](#)
[Logout](#)

Roll	Name	Address	Mobile No.	E-Mail
1000	ram	haldia	7678977	r@123
1001	Amit	kolkata	55765765	am@123
1002	Rahul	kolkata	76898989	rahul@123
1003	SOURAV	HIT COLLEGE	7477370591	sourav@123
1004	Rohan	delhi	78666229	rohan@123
1005	Rohan	delhi	78666229	rohan@123
1006	q7	kolkata	789654	7p@gm
1007	Mohan	kolkata	9091891293	ak@gmail.com

Activate Windows
Go to Settings to activate Windows.

enrollStudents.jsp

Student Enroll Screen

Enter Student Roll:

Enter Course Name:

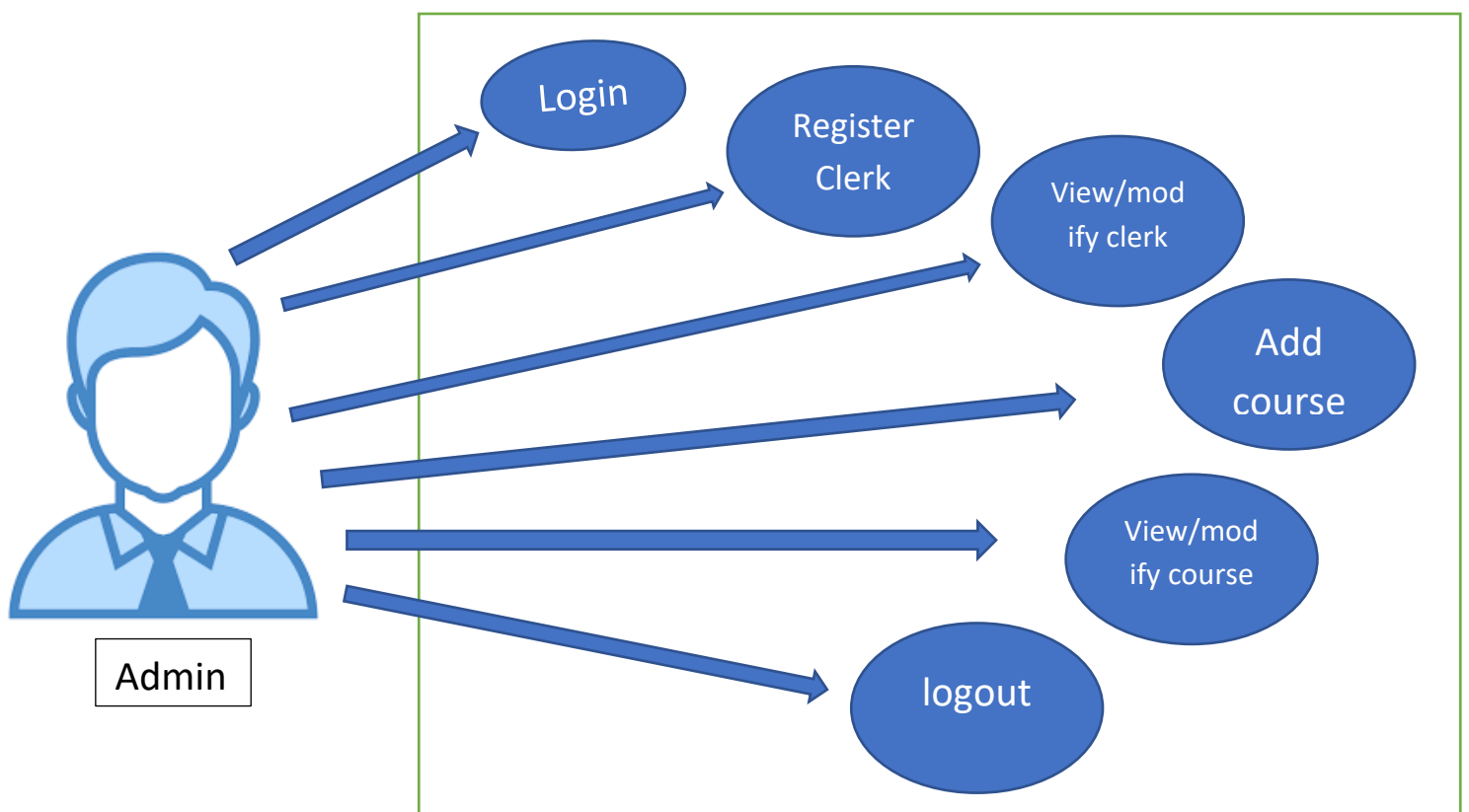
[Back](#)
[Logout](#)

Activate Windows
Go to Settings to activate Windows.

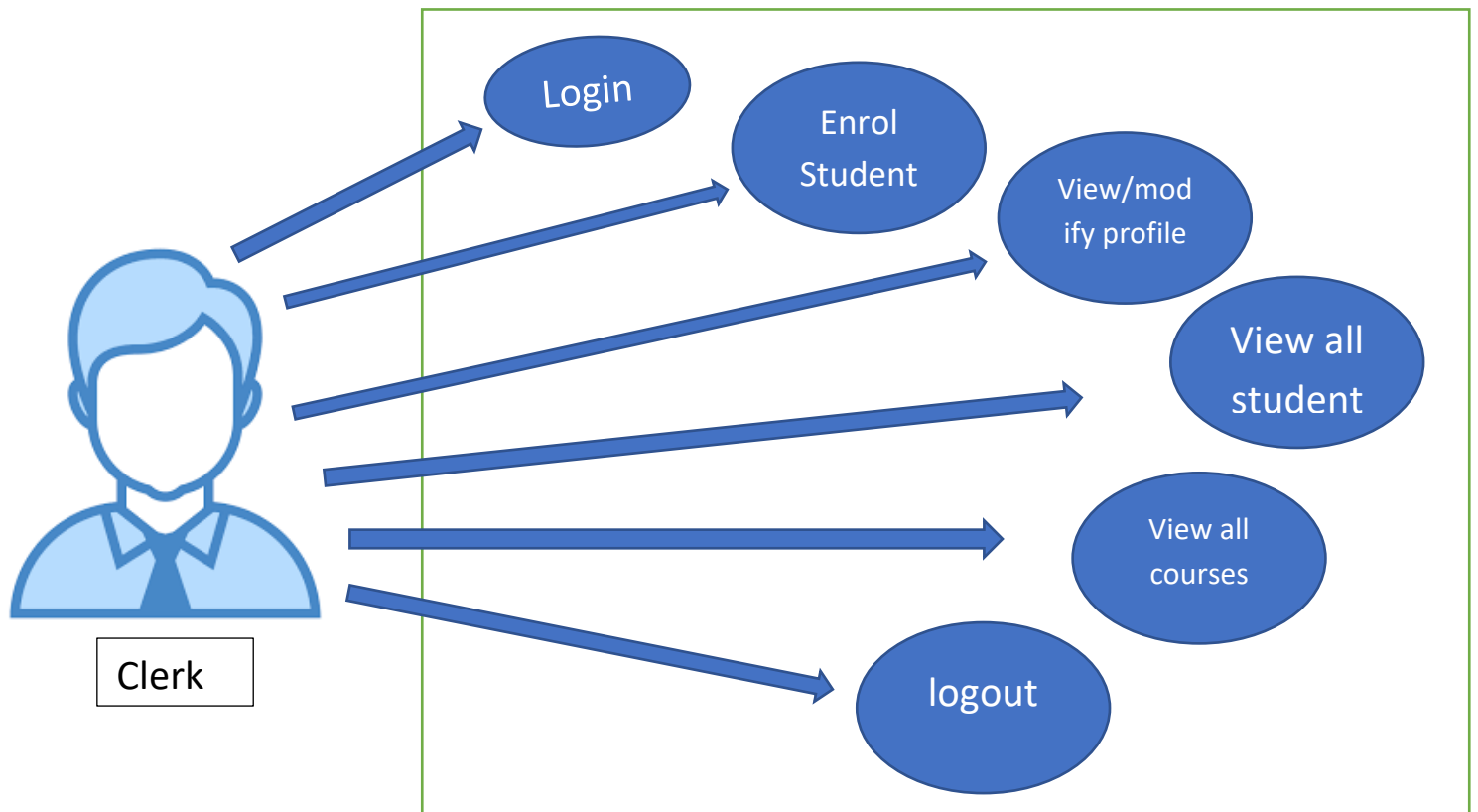
APPLICATION WORKFLOW

➤ USING CASE DIAGRAM

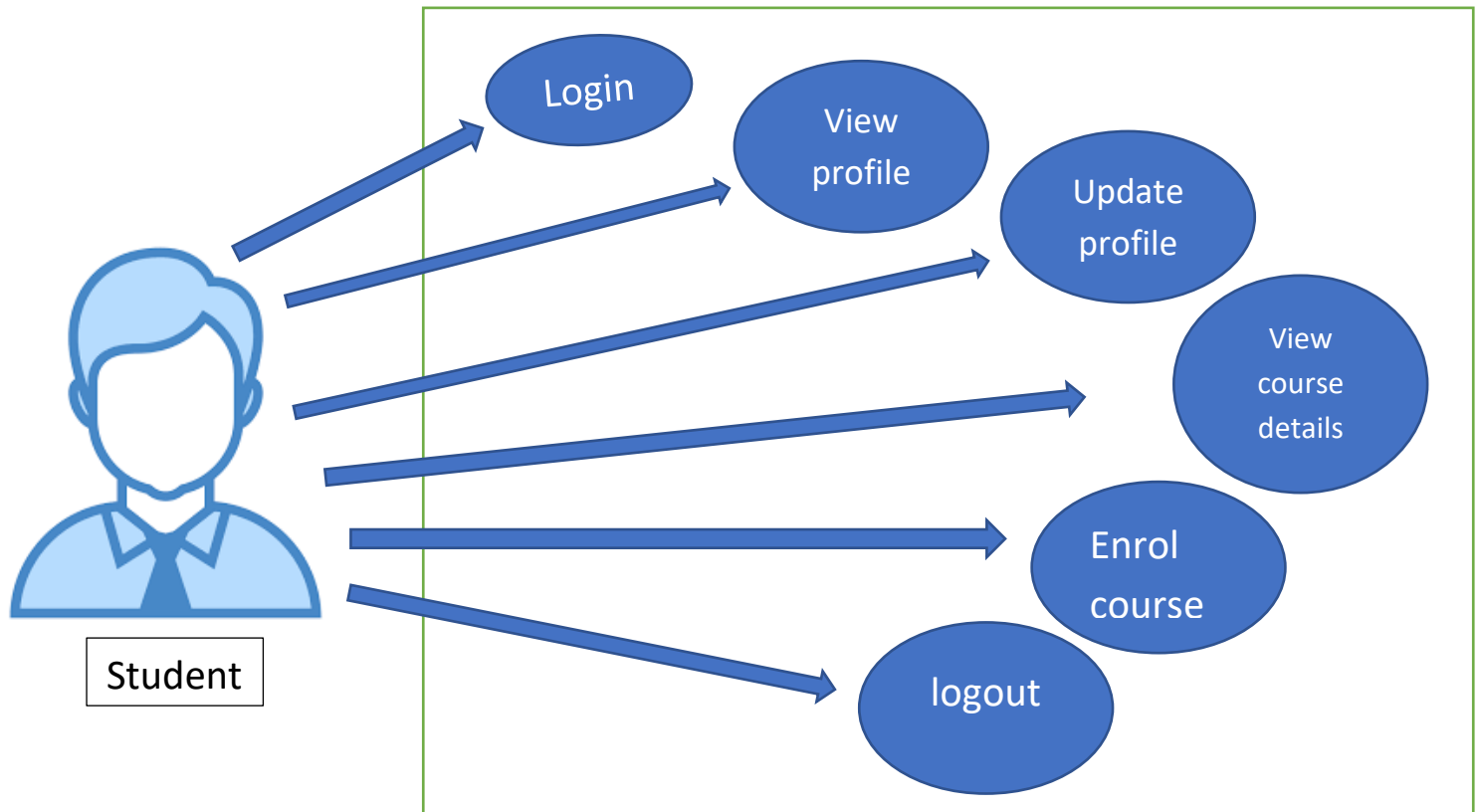
Case 1: Admin Case Diagram



Case 2: Clerk Case Diagram



Case 3: student case Diagram



➤ USING CASE DESCRIPTION

Admin

Case I: Login Description

CASE	<i>Login</i>
Goal	<i>All users logging into the system should be authenticated using a unique login-id and password (operations to be supported based on type of user)</i>
Preconditions	If the user type is 'Admin', credential details should exist. If the user type is 'Clerk', he/she should be registered by the admin. If the user type is 'Student', he/she should be registered by the himself or clerk.
Success End Condition	If the user type is 'Admin', then redirect to the Admin page. If the user type is 'Clerk', then redirect to the Clerk page. If the user type is 'Student', then redirect to the Student page.

Failed End Condition	<i>The end user is redirected to an Error Page, and/or is asked to re-enter login credentials.</i>
Primary, Secondary Actors	Admin, Clerk, Student.
Trigger	<i>Login button</i>

DESCRIPTION	Step	Action
	1	<i>Enter Login credentials (id & password)</i>
	2	<i>Click on Login button</i>
	3	<i>If id & password is Success, then identify user type Display appropriate(Admin/Clerk/student) home page</i>
	Step	Branching Action
	1	<i>If 'id' is not existing then return with requesting for registration</i>
	2	<i>If password is not matching return with suitable error message say 'Re-enter id & password'</i>
Related Information/Use cases	<i>Not Applicable</i>	
Priority	<i>P1</i>	
Performance	<i>5 seconds</i>	
Frequency	<i>10 / hour</i>	
Assumptions	<i>Admin/Customer/Student login credentials are available in the database and others are already registered with their credentials</i>	

Case Description to Add new courses

USE CASE	<i>The admin should be able to add new course and its detail.</i>	
Goal	<i>To enable Administrator to create and add new course.</i>	
Preconditions	<i>Administrator must be logged in to be able to create a new course.</i>	
Success End Condition	<i>"Redirect to Admin home page"</i>	
Failed End Condition	<i>"Redirect to Error Page"</i>	
Primary, Secondary Actors	<i>Administrator</i>	
Trigger	<i>'Add course button'</i>	
DESCRIPTION	Step	Action
	1	<i>Provide appropriate course details</i>
	2	<i>Click on Add course button</i>
	Step	Branching Action
	1	<i>If failed to add more course details</i>
	2	<i>Display appropriate message to the admin</i>
Related Information/Use cases	<i>Not applicable</i>	
Priority	<i>P1</i>	
Performance	<i>Approx. 4 sec</i>	

Frequency	2 / Month
Assumptions	Course Table is available in the database

Case Description to get all course Detail

USE CASE #	The admin should be able to get a course detail.	
Goal	To enable Administrator to get course detail using course name.	
Preconditions	Administrator must be logged in to be able to fetch course detail.	
Success End Condition	"Redirect to Course Detail Page"	
Failed End Condition	"Redirect to No course Exist"	
Primary, Secondary Actors	Administrator	
Trigger	'Get Course Detail Button'	
DESCRIPTION	Step	Action
	1	Provide appropriate Course Name
	2	Click on Get course Detail Button
	Step	Branching Action
	1	If failed to no course exist
	2	Display appropriate message to the admin
Assumptions	Course Details are available in the database	

Case Description to get all course Detail

USE CASE #	The admin should be able to get all course detail and modify it.	
Goal	To enable Administrator to get All course Detail.	
Preconditions	Administrator must be logged in to be able to fetch course detail.	
Success End Condition	"Redirect to Course Detail Page"	
Failed End Condition	"Redirect to Admin home Page"	
Primary, Secondary Actors	Administrator	
Trigger	'Get All Course Detail Link'	
DESCRIPTION	Step	Action
	1	Select Get all courses link.
	Step	Branching Action
	1	If failed to Admin Home Page.
	2	Display appropriate message to the admin
Assumptions	Course Details are available in the database	

Case Description to Update Course Detail

USE CASE #	The admin should be able to modify a course.
-------------------	--

Goal	<i>To enable Administrator to get update a particular course.</i>	
Preconditions	Administrator must be logged in to be able to fetch course detail.	
Success End Condition	<i>"Redirect to Update course page with its initial details"</i>	
Failed End Condition	<i>"Redirect to Couse Updating Fails"</i>	
Primary, Secondary Actors	Administrator	
Trigger	<i>'Update link in the table'</i>	
DESCRIPTION	Step	Action
	1	Click Get all course Detail link, from Admin home Page
	2	Click on Update button on the particular course needs to be updated
	3	Fill the necessary course details
	4	Select update Button to update
	Step	Branching Action
	1	If failed to Updating Fail Page
	2	Display appropriate message to the admin
Assumptions	Course Details are available in the database	

Case Description for Register Clerk

USE CASE #	<i>The admin should be able to register a clerk</i>	
Goal	<i>To enable Administrator to register a clerk with its details required.</i>	
Preconditions	Administrator must be logged in to be able register a clerk.	
Success End Condition	<i>"Redirect to Clerk Registered Successfully"</i>	
Failed End Condition	<i>"Redirect to Clerk Registration Fail Page"</i>	
Primary, Secondary Actors	Administrator, clerk	
Trigger	<i>'Register Clerk Button'</i>	
DESCRIPTION	Step	Action
	1	Click on Register Clerk Link from Admin Home page
	2	Give the credentials needed to register a clerk.
	3	Click on Register button to register
	Step	Branching Action
	1	If failed to Registration Fails Page
Assumptions	2	Display appropriate message to the admin
	Clerk table is present in the database	

Case Description to get Clerk Detail

USE CASE #	<i>The admin should be able to get a clerk detail.</i>	
Goal	<i>To enable Administrator to get clerk detail using clerk Id.</i>	
Preconditions	Administrator must be logged in to be able to fetch clerk detail.	
Success End Condition	<i>"Redirect to Clerk Detail Page"</i>	

Failed End Condition	<i>"Redirect to No clerk Exist"</i>	
Primary, Secondary Actors	<i>Administrator</i>	
Trigger	<i>'Get Clerk Detail Button'</i>	
DESCRIPTION	Step	Action
	1	<i>Provide appropriate Clerk Id number</i>
	2	<i>Click on Get clerk Detail Button</i>
	Step	Branching Action
	1	<i>If failed to no clerk exist</i>
	2	<i>Display appropriate message to the admin</i>
Assumptions	Clerk details are available in the database	

Case Description to get all Clerk Detail

USE CASE #	<i>The admin should be able to get all clerk detail and modify it.</i>	
Goal	<i>To enable Administrator to get All clerk Detail.</i>	
Preconditions	<i>Administrator must be logged in to be able to fetch clerk detail.</i>	
Success End Condition	<i>"Redirect to Clerk Detail Page"</i>	
Failed End Condition	<i>"Redirect to Admin home Page"</i>	
Primary, Secondary Actors	<i>Administrator</i>	
Trigger	<i>'Get All Clerk Detail Link'</i>	
DESCRIPTION	Step	Action
	1	<i>Select Get all courses link.</i>
	Step	Branching Action
	1	<i>If failed to Admin Home Page.</i>
	2	<i>Display appropriate message to the admin</i>
Assumptions	Clerk Details are available in the database	

Case Description to Update Clerk Detail

USE CASE #	<i>The admin should be able to modify a clerk.</i>	
Goal	<i>To enable Administrator to get update a particular clerk.</i>	
Preconditions	<i>Administrator must be logged in to be able to fetch clerk detail.</i>	
Success End Condition	<i>"Redirect to Update clerk page with its initial details"</i>	
Failed End Condition	<i>"Redirect to Clerk Updating Fails"</i>	
Primary, Secondary Actors	<i>Administrator</i>	
Trigger	<i>'Update link'</i>	
DESCRIPTION	Step	Action
	1	<i>Click Get all clerk Detail link, from Admin home Page</i>
	2	<i>Click on Update button on the particular clerk needs to be updated</i>
	3	<i>Fill the necessary clerk details</i>
	4	<i>Select update Button to update</i>
	Step	Branching Action
	1	<i>If failed to Updating Fail Page</i>

	2	<i>Display appropriate message to the admin</i>
Assumptions	Clerk Details are available in the database	

Student

Case I: Login Description

CASE	<i>Login</i>
Goal	<i>All users logging into the system should be authenticated using a unique login-id and password (operations to be supported based on type of user)</i>
Preconditions	If the user type is 'Admin', credential details should exist. If the user type is 'Clerk', he/she should be registered by the admin. If the user type is 'Student', he/she should be registered by the himself or clerk.
Success End Condition	If the user type is 'Admin', then redirect to the Admin page. If the user type is 'Clerk', then redirect to the Clerk page. If the user type is 'Student', then redirect to the Student page.
Failed End Condition	<i>The end user is redirected to an Error Page, and/or is asked to re-enter login credentials.</i>
Primary, Secondary Actors	Admin, Clerk, Student.
Trigger	<i>Login button</i>

Case Description to View & Update profile

USE CASE	<i>Student can able to view their details when they are successfully login and even they update some of the details of their profile</i>	
Goal	<i>To view the student details and even they update their some of the details</i>	
Preconditions	Student must be logged in to be able to update their profile.	
Success End Condition	<i>"Redirect to StudentProfile page or Studentprofileupdate page"</i>	
Failed End Condition	<i>"Redirect to Error Page if update is not able to done"</i>	
Primary, Secondary Actors	Administrator ,student,clerk	
Trigger	<i>'Add update button'</i>	
DESCRIPTION	Step	Action
	1	<i>Click on view profile their update profile button</i>
	2	<i>Click on Update</i>
	Step	Branching Action
	1	<i>If failed to update the profile</i>
	2	<i>Display appropriate message to the student</i>

Related Information/Use cases	<i>Not applicable</i>
Priority	<i>P1</i>
Performance	<i>Approx. 4 sec</i>

Case Description to View all courses and Register

USE CASE #	<i>Student can view all the courses available and register in them and they can even view the courses they had already registered full details</i>	
Goal	<i>To enable student to register for the courses</i>	
Preconditions	Student can be able to register on a particular courses	
Success End Condition	<i>"Redirect to Registration success"</i>	
Failed End Condition	<i>"Redirect to Not available in place of Register"</i>	
Primary, Secondary Actors	<i>Admin, student, clerk</i>	
Trigger	<i>'Get All enroll course details', 'register'</i>	
DESCRIPTION	Step	Action
	1	<i>Select Register link.</i>
	Step	Branching Action
	1	<i>If failed to Not applicable</i>
	2	<i>Display appropriate message to the student</i>
Assumptions	Course Details are available in the database	

Clerk

Case I: Login Description

CASE	<i>Login</i>
Goal	<i>All users logging into the system should be authenticated using a unique login-id and password (operations to be supported based on type of user)</i>
Preconditions	If the user type is 'Admin', credential details should exist. If the user type is 'Clerk', he/she should be registered by the admin. If the user type is 'Student', he/she should be registered by the himself or clerk.
Success End Condition	If the user type is 'Admin', then redirect to the Admin page. If the user type is 'Clerk', then redirect to the Clerk page. If the user type is 'Student', then redirect to the Student page.
Failed End Condition	<i>The end user is redirected to an Error Page, and/or is asked to re-enter login credentials.</i>
Primary, Secondary Actors	Admin, Clerk, Student.
Trigger	<i>Login button</i>

Case Description to View & Update profile

USE CASE	<i>Clerk can able to view their details when they are successfully login and even they update some of the details of their profile</i>	
Goal	<i>To view the clerk details and even they update their some of the details</i>	
Preconditions	Clerk must be logged in to be able to update their profile.	
Success End Condition	<i>"Redirect to ClerkProfile page or Clerkprofileupdate page"</i>	
Failed End Condition	<i>"Redirect to Error Page if update is not able to done"</i>	
Primary, Secondary Actors	Administrator , clerk	
Trigger	'Add update button'	
DESCRIPTION	Step	Action
	1	Click on view profile their update profile button
	2	Click on Update
	Step	Branching Action
	1	If failed to update the profile
	2	Display appropriate message to the student
Related Information/Use cases	Not applicable	

Case Description to get all course Detail

USE CASE #	<i>The clerk should be able to get all course detail and modify it.</i>	
Goal	<i>To enable Clerk to get All course Detail.</i>	
Preconditions	Clerk must be logged in to be able to fetch course detail.	
Success End Condition	<i>"Redirect to Course Detail Page"</i>	
Failed End Condition	<i>"Redirect to Clerk home Page"</i>	
Primary, Secondary Actors	Administrator ,clerk	
Trigger	'Get All Course Detail Link'	
DESCRIPTION	Step	Action
	1	Select Get all courses link.
	Step	Branching Action
	1	If failed to clerk Home Page.
	2	Display appropriate message to the admin
Assumptions	Course Details are available in the database	

Case Description to enroll any Student

USE CASE #	<i>The clerk should be able to enroll any student to a particular courses if he/she not registered</i>	
Goal	<i>To enable Clerk to enroll a particular student on a course</i>	
Preconditions	Clerk must be logged in to be able to fetch course detail.	
Success End Condition	<i>"Redirect to Enroll student screen Page"</i>	
Failed End Condition	<i>"Redirect to Clerk home Page"</i>	
Primary, Secondary Actors	Administrator ,clerk	
Trigger	'Get Enroll Link'	
DESCRIPTION	Step	Action
	1	Select Enroll link.
	Step	Branching Action
	1	If failed to clerk Home Page.
	2	Display appropriate message to the clerk
Assumptions	Course Details are available in the database	

FUTURE SCOPE OF IMPROVEMENT

Since the project has been done using html, css, Java Script for the front-end development. And the back-end is developed using JDBC, Java, JSP i.e.; using Apache tomcat server.

The technologies used are in high demand in the current scenario with the scope of enhancement.

So, therefore this project can be further enhanced using more advance java technologies.

CONCLUSION

This new system is surely to benefit the organization, with various registration and modification methodology. The system has various level of access restriction. The addition or modification is directly reflected in database design to the handle multiple query demanded by various level of accessor.

APPENDICES

- MVC: Model View Controller
- JDBC: Java Database Connectivity
- JSP: Java Servlet Page
- Servlet

REFERENCES/BIBLIOGRAPHY

- <https://javatpoint.com/java-tutorial>
- <https://stackoverflow.com/>
- <https://w3schools.com/>
- <https://geekforgeeks.com/>