

# IGNOU

Project Proposal (Synopsis) + Project Report

## BACHELOR IN COMPUTER APPLICATION (BCA) (Program code)

On

***Institute Management System***

***or***

***Java Based User-Faculty Institute Workspace  
(Web Application)***

### Submitted By

Name: - Ujjwal Pandey

Enrollment No: - 185512614

Course Code: - BCSP64

Regional Center: - Delhi RC-01

Contact: - +91 8375990500, ujjwalpandey.aps@gmail.com

INDIRA GANDHI NATIONAL OPEN UNIVERSITY  
(SCHOOL OF COMPUTER AND INFORMATION SCIENCE (SOCIS),  
(1ST FLOOR, DEC BUILDING, IGNUO, MAIDAN GARHI, New Delhi – 110068.)

## Synopsis after corrections

Sequence of documents in it

1. Original copy of Approved Project Proposal Proforma,
2. Certificate of Originality,
3. Guide's Resume,
4. Synopsis, and
5. Project Report



SCHOOL OF COMPUTER AND INFORMATION SCIENCES  
IGNOU, MAIDAN GARHI, NEW DELHI – 110 068

B070133-S-20

II. PROFORMA OF BCA PROJECT PROPOSAL (BCSP-064)  
(Project's Title and Guide's Details)

OCT-DEC-20

Enrolment No.: 185512614..... Regional Centre Code: RC-1... Study Centre: Project (0750P)

1. Name of Student: Ujjwal Pandey.....

2. Address of the student: C-434, Sangam Vihar - New Delhi - 110080.....

3.(a) E-mail: Ujjwal.pandey@apsk.com (b) Telephone/ Mobile No.: 8375990500.....

4. Title of the Project : Institute Management System or Java Based User faculty Institute workspace.

5. Name of Project Guide: Vikas Kumar. 5.(b) Designation of Project Guide: Technical Lead

6. Address of Project Guide: B5/48, Sector 3, Rohini, Delhi - 110085.....

7. Qualification of the Guide\*: Ph.D.  M.Tech  B.Tech  MCA  Any other

(Compulsory to Attach bio-data of Guide)

\*Note : i. All the above mentioned Degrees must have been awarded in Computer Science/IT only.  
ii. A Guide should not guide more than 8 students of BCA at any point of time.

8. Industrial / Teaching experience of the Guide (in Years) 10 years.

9. Software Used for this Project: Eclipse and IntelliJ IDEA.....

Note : 1. Use of Visual Basic and MS-Access as Front End and Back End respectively is forbidden. But, you are permitted to use Visual Basic with other Software. Also, you can use MS-Access with other software.

2. Use of C or C++ Programming Language for Project Related to Database Management is strictly forbidden.

*Ujjwal Pandey*  
Signature of the Student  
Date: 30/12/20

*Vikas Kumar*  
Signature of the Guide  
Date: 29/12/20

**Important:**

1. Attach this Proforma along with Guide's Bio-data and Project Synopsis in the Project Report.
2. Not more than one student is permitted to work on a project.
3. Complete project as per the comments of Synopsis evaluator, then only submit your Project Report.

For Office Use Only

Approved

Not approved

*Ujjwal Pandey*  
Signature, Designation, Stamp of the  
Project Proposal Evaluator  
Date: 23/12/20

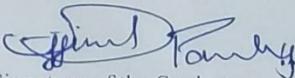
**Suggestions for reformulating the Project:**

1. Show relationship b/w tables in database
2. Show relationship b/w tables in database
3. Show relationship b/w tables in database
4. Show relationship b/w tables in database

**IX CERTIFICATE OF ORIGINALITY**

This is to certify that the project report entitled Institute Management system for Java Based User faculty Institute Workspac  
Submitted to **Indira Gandhi National Open University** in partial fulfilment of the requirement  
for the award of the degree of **BACHELOR OF COMPUTER APPLICATIONS ( BCA )**, is  
an original work carried out by Mr./ Ms. Ujjwal Pandey  
Enrolment No.: 185512614 under the guidance of Mr./ Ms. Vikas Kumar

The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfilment of the requirement of any course of study.



Signature of the Student

**Ujjwal Pandey**

Name and Address

of the student: 67-434,

Sangam Vihar,  
New Delhi - 110080

Enrolment No.: 185512614



Signature of the Guide

**Vikas Kumar**

Name, Designation and  
Address of the Guide

**Technical Lead**  
**(Advanced Technologist III)**  
B5/98, Sector 3,  
Rohini, Delhi - 85.  
"vikas.chaturvedi8587@gmail.com"

## Guid Resume

# Vikas Kumar

Technical Lead



### CONTACTS

B5/98, Sector 3, Rohini, Delhi - 85  
vikas.chaturvedi8587@gmail.com  
+91 - 8130806660

### ABOUT ME

Dynamic & Techno-Savvy IT professional offering around 10 years of experience in .Net/SprintBoot Technology.

### WORK EXPERIENCE

Boeing India Pvt. Ltd., Bangalore  
Advance Technologist III  
November 2018 - Present



- Member of architect group, which is responsible to design new Products and solutions.
- CI/CD with PCF and Azure.
- Requirement gathering from business users and stakeholders.
- Interact with AI/ML team to convert their models into visualization and dashboard building.
- Managing JIRA for backlog and story establishing.
- Extensively involved in code reviews across teams.
- Actively involved in presenting sessions on upcoming technologies.

ARI Network Services, Gurugram

**Software Engineer III/Technical Lead**

November 2016 – August 2018



- Designed & developed enterprise level API, the product which is used by enterprise customers worldwide. Hosted the API on AWS API gateway.
- Involved in the sprint planning and create user story and tasks.
- Actively involved in mentoring team and tech talks.
- Actively involved in the designing and development of the modules.
- Extensively involved in code reviews across teams.
- Actively involved in presenting sessions on upcoming technologies.

3Pillar Global, Noida UP

**SSE > Module Lead > Technical Lead**

August 2012 – October 2016



- Involved in the sprint planning and create user story and tasks.
- Involved in defect fixing and Change Requests.
- Provide support to peers.
- Actively involved in the designing and development of the modules.
- Involved in optimizing already developed modules and supportive to other team members and good team player.
- Actively involved in presenting sessions on upcoming technologies.

ISTS Infotech/ Clear2Pay, Noida UP

**Software Developer**

February 2011 – August 2012

**Software Trainee**

March 2008 – March 2009



**Software Developer**

- Involved in the Analysis of business modules.
- Actively involved in the designing and development of the modules.
- Involved in defect fixing and Change Requests.
- Involved in optimizing already developed modules and supportive to other team members and good team player.

**Software Trainee**

- Involved in various Projects undertaken by the organization performing different functions including Designing & Development.
- Holds the credential of developing/ managing different Application Modules as per internal requirements and for the clients of the organization.

Viknet Software Solutions

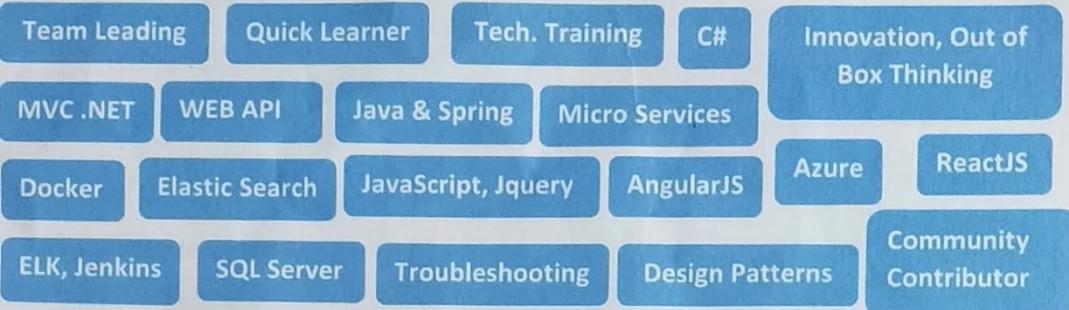
**Software Developer**

June 2009 – February 2011



- Successfully handling the task of developing/managing different Application Modules as per client requirement.
- Gaining exposure in end-to-end development of software products from requirement analysis to system study, designing, coding, testing and debugging.
- Involved in requirement analysis, database design, coding, testing, implementation and maintenance of several software applications.
- Communicating with the clients, interviewing end-users to know the requirements and expectations of the clients.
- Ensuring technical solutions are designed for performance, reliability, scalability, maintainability, supportability, business continuity, and business agility while leveraging industry's best practices.
- Assuring both quality and customer service while leading and driving development tasks and projects to successful completion. Prepare documentation for the design plan, review specifications for modifications, integrating components and testing.

### SKILLS & Competencies



### Education Credentials

- MCA – Sikkim Manipal University
- GNIIT – NIIT
- B. Com (2006) – Delhi University
- CLASS XII (2002) – C.B.S.E. Board
- CLASS X (2000) – C.B.S.E. Board

### Community Contributions



### Follow Me



[vikas-chaturvedi-23344824 OR Profile Link](#)



[@vikaspart2 Or Profile Link](#)



[@vikaspart2](#)

Qkumar  
29/12/20

## **TABLE OF CONTENTS**

	<b><u>Page No.</u></b>
1. Introduction of Project	....09
2. Objectives of Project	....10
3. Project category	....11
4. Tools/Platform of Project – H/W and S/W	....15
5. Design of Project	
a. DFD	....17
b. ER DIAGRAM	....20
c. DATA STRUCTURE	....21
d. Summarizing tables Relationship	....23
e. Number of Modules & Their Description	....24
f. Process Logic of Modules	....25
6. Limitation of the Projects	....28
7. Future Scope of Project	....28
8. Bibliography	....29

## **INTRODUCTION**

### **Institute Management System**

**Institute Management System** is the software made for the user and faculty to find, manage, and perform a maximum of functions. Many modules are included in the software to perform full operation like, For user student: - searching courses, getting full details about the respective course, getting enrolled in the respective course by full-filling a form with required detail which on completion provide a pdf file having all submitted details with a unique enrolment no, find details about the institute, get details to connect with the institute in different ways, For faculty: - they can see no. of students in the institute in a list and details of a student with their specific enrollment number, list of faculties in the institute and their detail, with all these only the owner will be able to add and delete new faculty in the list of the institute having real-time all data visible to him/her.

At very first when a user will open the institute site, the site will open its home page which will contain a nice image with a logo and inspiring line at the center nothing else, but on scrolling it will show other content like achievements, few courses, hyperlink to all the courses page, about the institute and few location contactable details with a hyperlink containing full contact page which will have a form for user or guardian to contact online through an online form, mail, and mobile numbers to contact in the institute to clear all the query.

## **OBJECTIVE OF THE PROJECT**

As the title of the project suggests, the objective of this software (web application) is to automate all the works (excluding teaching) of the Institute related to the management of admission getting details etc. Some of the features of this software would be: -

- Creating a system to automate the services offered by the Institute, it would be more flexible than human to human contact as of today's corona situation.
- It allows a user search for their loved and required course within the institute and get enrolled in it without coming in contact with other humans, and reduce the spread of COVID-19.
- It keeps track of the students enrolled to date and faculty in the institute.

## PROJECT CATEGORY

- The project is based on a **three-tier architecture**. The three-tier architecture where the application is divided into three logical constituents
  - User tier – Provide services such as user interface. (JSP, CSS, JavaScript, etc.) also known as **View**.
  - Business tier – Implement business rules which will control how our web app will work (Java and JavaScript) also known as **Controller**.
  - Data tier – Provide handling and validation of data getting and saving data in the database. (MySQL in this case) also known as **Model**.
- Why three-tier not two-tier model or Disadvantages of two-tier architecture model.
  - Two-tier puts extra load on the server
  - Difficult to implement incremental improvements.
  - Applications are bound to the data source.

## REASONS FOR USING Eclipse

Eclipse is a powerful Java project management and build management IDE. That means we can manage java project builds very easily using eclipse IDE. Eclipse can help us to minimize our project and build management time and efforts as comparison to any other non-ide code editor. It is fine to manage the project manually if it is small. But If the project is huge like our and future project or there are many, then it is very hard for a developer to manage each of them manually.

## Advantages of Eclipse IDE project

- ✓ It makes the project build process easy.
- ✓ It provides an easy and uniform build system.
- ✓ It provides quality project document Information.
- ✓ Managing project dependencies.
- ✓ Provides guild lines for better project management practices.
- ✓ Facilitate easy and transparent migration to new features.
- ✓ It allows to building projects using project object model (POM).
- ✓ It downloads required dependency's jar files automatically from Maven central repositories.
- ✓ Gives auto-correction and suggestions for code completion.

## THE ADVANTAGES OF RDBMS (in my case- MySQL)

A database system is essentially a sophisticated, computerized record-keeping system, a repository for a collection of computerized data files. A database system maintains information and makes that information available on-demand, for this purpose a database system provides a set of facilities to perform such operations.

The benefits of a database system over any traditional system are obvious as the database is integrated as well as shared, thus a database eliminates redundancy, and also as a consequence, the database lets multiple users access the same piece of data.

The most important advantage of the database is to maintain integrity, i.e.; it ensures that the change made to the database by authorized users does not result in a loss of data consistency and guard against accidental damage to the database.

## **RDBMS has the following facilities**

- Creation of files, Addition of data, Deletion of data, Modification of data.
- Retrieving data collectively or selectively.
- The data stored can be sorted or indexed at the user's discretion or direction.
- Various reports can be produced from the system. These may either be standardized reports or that may be specifically generated according to specific user definition.
- The mathematical function can be performed and the data stored in the database can be manipulated with functions to perform the desired calculations.
- To maintain data integrity and database use.
- Data integrity for multiple users.
- Providing a form-based interface for easy accessibility and data entry.

## **Major Advantages of Using MySQL (and why not any other RDBMS tech)**

MySQL is a free-to-use, open-source database that facilitates effective management of databases by connecting them to the software. It is a stable, reliable and powerful solution with advanced features like the following:

### **1. Data Security**

MySQL is globally renowned for being the most secure and reliable database management system used in popular web applications like WordPress, Facebook and Twitter.

### **2. On-Demand Scalability**

MySQL offers unmatched scalability to facilitate the management of deeply embedded apps using a smaller footprint even in massive warehouses that stack terabytes of data. On-demand flexibility is the star feature of MySQL.

### 3. High Performance

MySQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance. Whether it is an eCommerce website that receives a million queries every single day or a high-speed transactional processing system, MySQL is designed to meet even the most demanding applications while ensuring optimum speed, full-text indexes, and unique memory caches for enhanced performance.

### 4. Round-the-clock Uptime

MySQL comes with the assurance of 24X7 uptime and offers a wide range of high availability solutions like specialized cluster servers and master/slave replication configurations.

### 5. Comprehensive Transactional Support

MySQL tops the list of robust transactional database engines available on the market. With features like complete atomic, consistent, isolated, durable transaction support, multi-version transaction support, and unrestricted row-level locking, it is the go-to solution for full data integrity. It guarantees instant deadlock identification through server-enforced referential integrity.

### 6. Complete Workflow Control

With the average download and installation time being less than 30 minutes, MySQL means usability from day one. Whether your platform is Linux, Microsoft, Macintosh, or UNIX, MySQL is a comprehensive solution with self-management features that automate everything.

### 7. Reduced Total Cost of Ownership

By migrating current database apps to MySQL, enterprises are enjoying significant cost savings on new projects.

### 8. The Flexibility of Open Source

All the fears and worries that arise in an open-source solution can be brought to an end with MySQL's round-the-clock support and enterprise indemnification. The secure processing and trusted software of MySQL combine to provide effective transactions for large volume projects. It makes maintenance, debugging and upgrades fast and easy while enhancing the end-user experience.

# TOOLS/PLATFORM, HARDWARE, AND SOFTWARE REQUIREMENT SPECIFICATION

## Tools/Platform

Project is developed using **Eclipse & MySQL 2017 command line** for storing data.

### **→Use of Technologies/Tools**

*It is going to be a java web application.*

1. Unconditionally java
2. JSP
3. CSS
4. JavaScript
5. MySQL
6. Hibernate

### **→Project type**

Java-based web application

### **→Use of IDE**

Eclipse

### **→Use of Browser**

Google Chrome, and  
Microsoft Edge

## Hardware requirement Specification

Altogether a Personal Computer with following components:

To use Browser on Windows®, you'll need:

- Windows 7, Windows 8, Windows 8.1, Windows 10 or later
- An Intel Pentium 4 processor or later that's SSE3 capable (as we need only a browser)
- 2 GB Memory (RAM)
- Hard Disk 125 GB
- Color Monitor
- Keyboard
- Mouse
- Printer

## Software requirement Specification

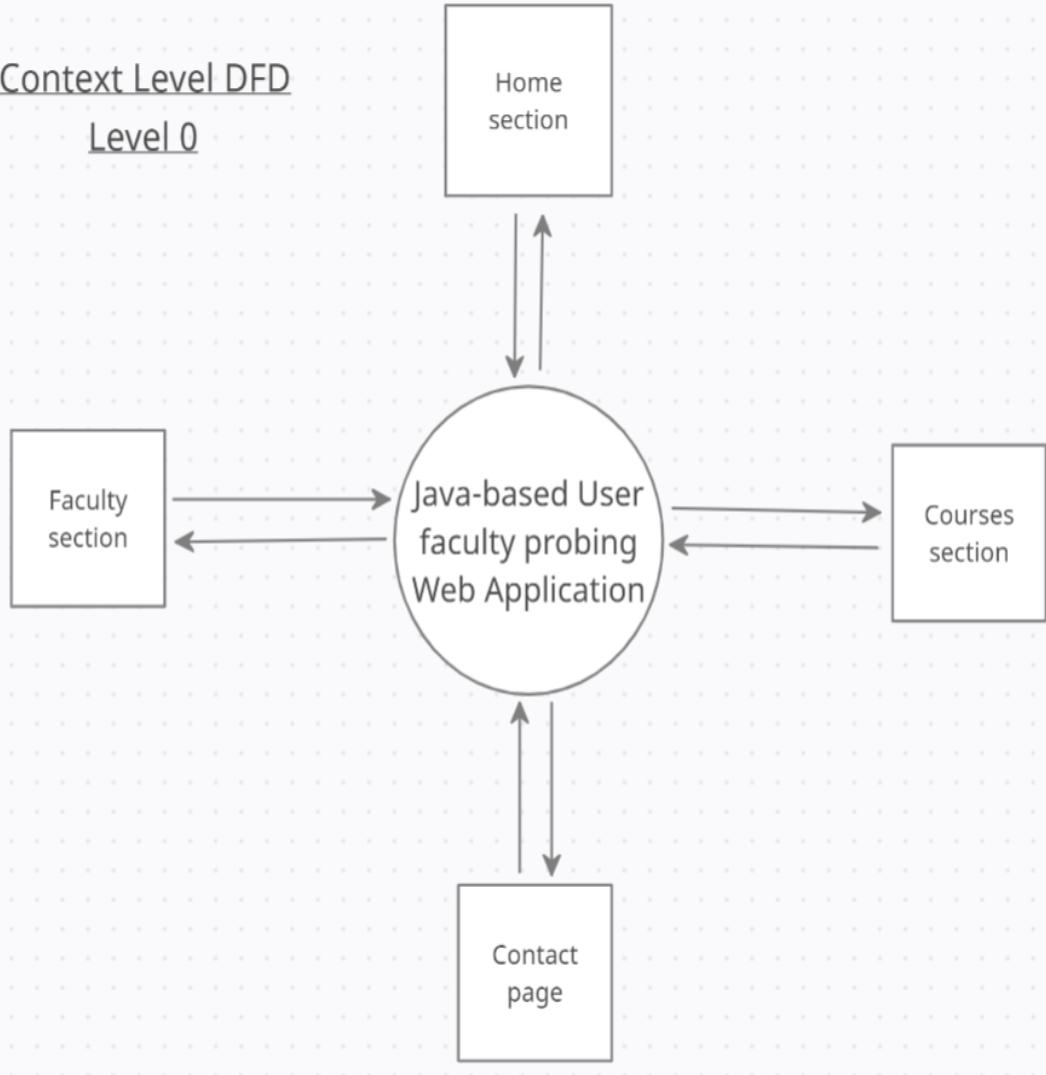
The software which was required for using the software is a new advanced **web browser** like Google Chrome, Microsoft Edge, Opera, mac Safari.

## CONTEXT LEVEL DFD

Created on www.app.createley.com

Context Level DFD

Level 0

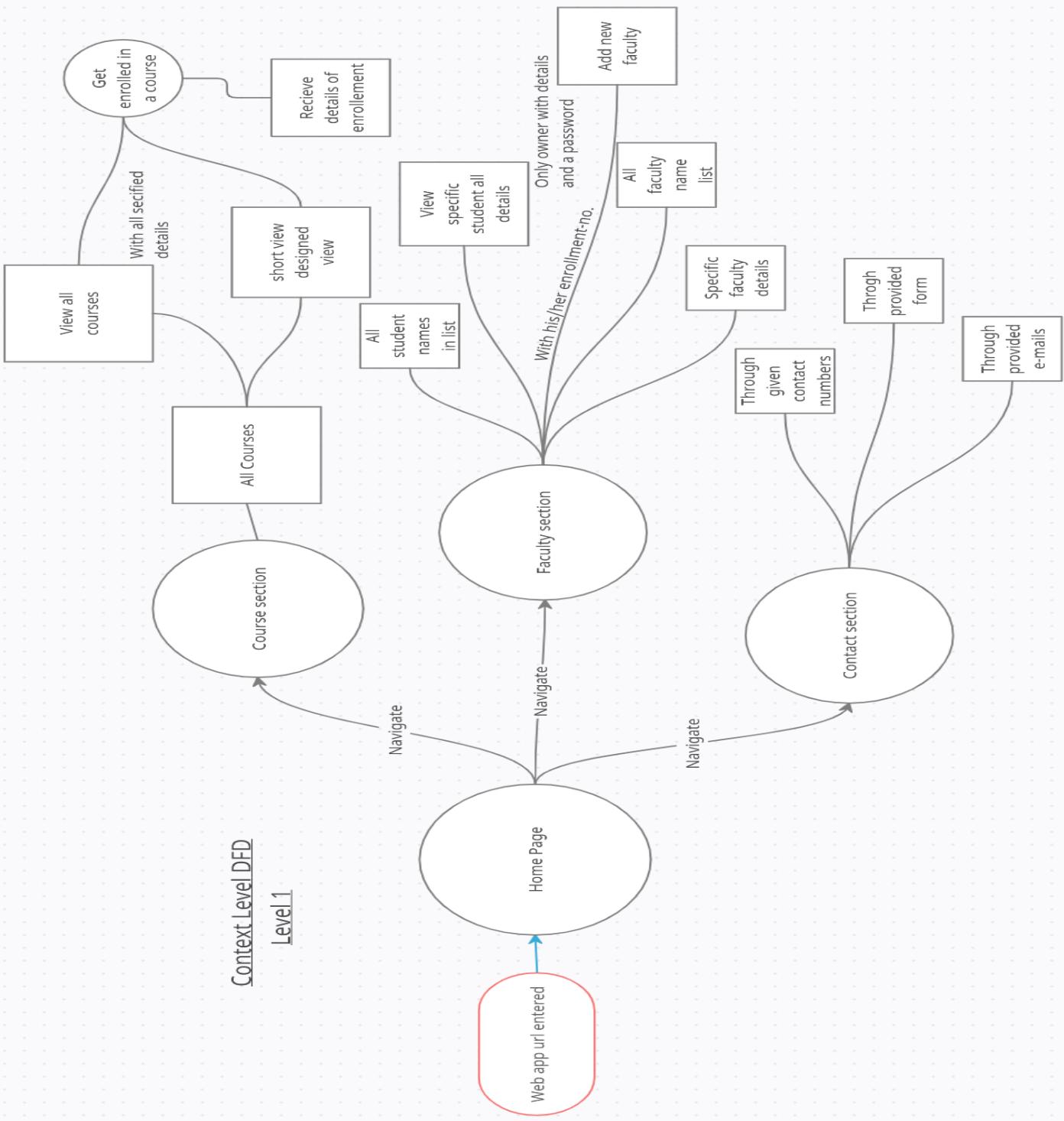


*DFD level 0*

*BB*

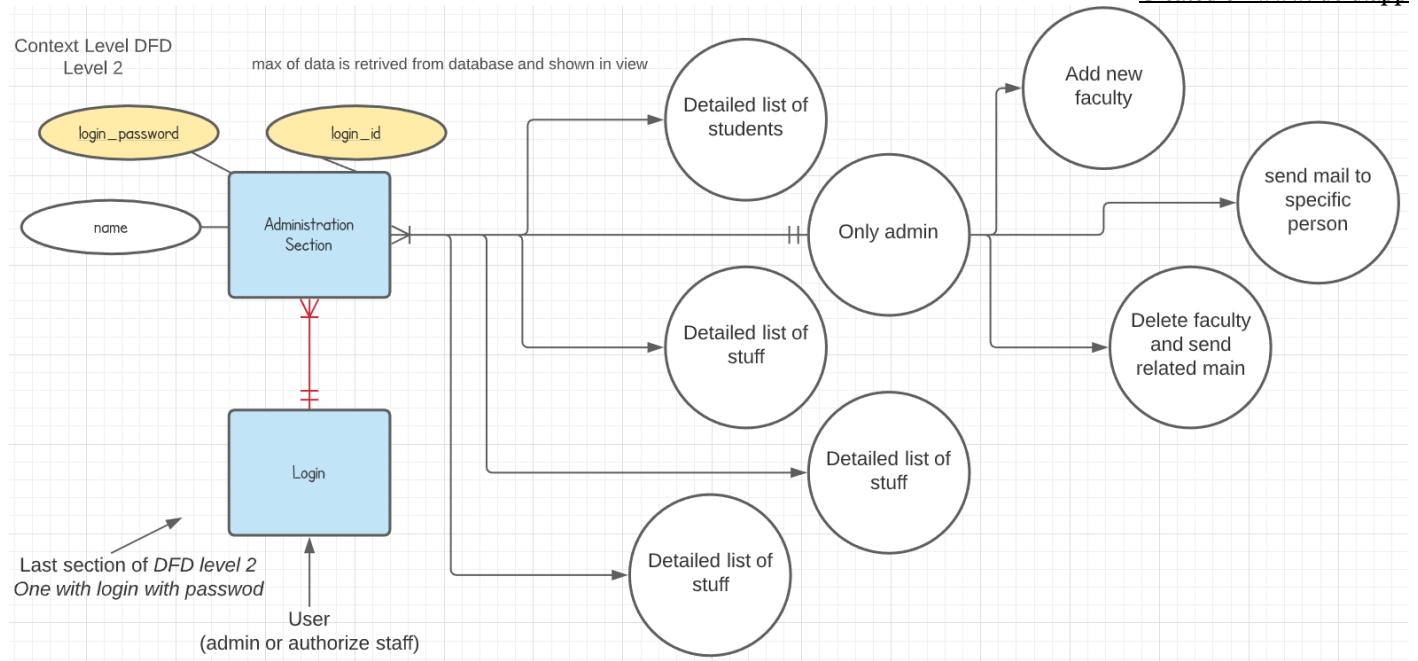
## CONTEXT LEVEL DFD

Created on www.app.createy.com



DFD level 1

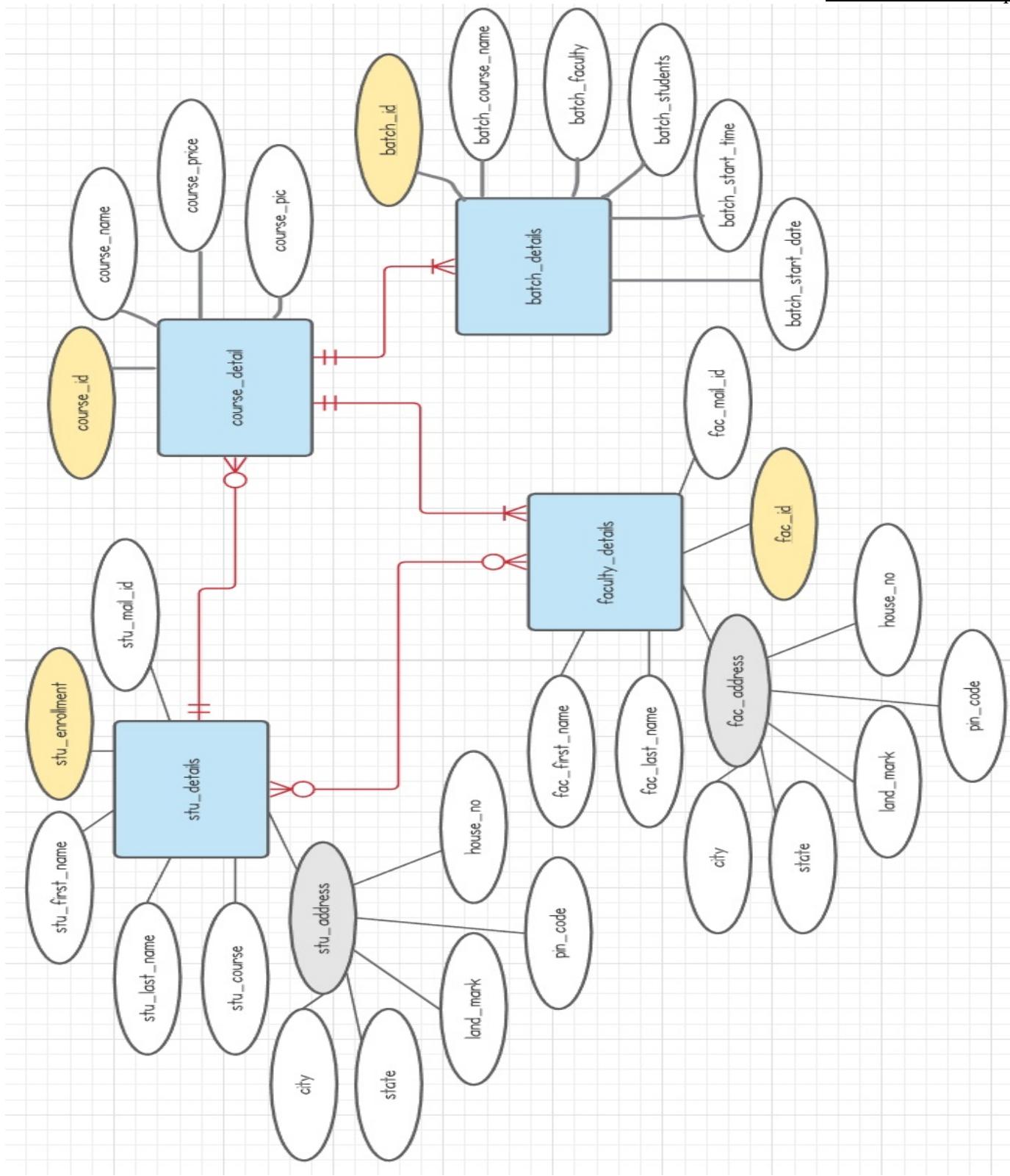
## CONTEXT LEVEL DFD



DFD level-2

## ER-Diagram

Created on [www.lucid.app](http://www.lucid.app)



## DATA-STRUCTURES

→ Tables are subject to be increased or decreased while code development.

### **Table Name: userdetails**

Column Name	Data Type	Constraints	Description
Email	VARCHAR (255)	primary key NOT NULL	unique identity of each user
city	VARCHAR (35)		City where user lives
contactNumber	VARCHAR (15)		User contact number
firstName	VARCHAR (20)		User's first name
houseNo	VARCHAR (15)		House number of the user
landMark	VARCHAR (255)		Landmark of use address
lastName	VARCHAR (20)		User's last name
password	VARCHAR (255)		User's password
pinCode	VARCHAR (255)		Pin code
registrationDate	VARCHAR (255)		Date when user registered
state	VARCHAR (255)		User's address state
userType	VARCHAR (255)		What type of user is Admin/Student/Faculty

### **Table Name: queryfromform**

Column Name	Data Type	Constraint	Description
qSoNo	Int	PRIMARY KEY NOT NULL	Unique id of query
qContact	VARCHAR (255)		Contact details given by person submitting query
qEmail	VARCHAR (255)		Email id provided by the person querying
qName	VARCHAR (255)		Name of querier
qQueryExplained	VARCHAR (255)		Brief of what they want to know
qQueryHead	VARCHAR (255)		Title of the query

### **Table Name: hibernate\_sequence**

<b>Column Name</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Description</b>
next_val	bigint	Not null	Used to number other tables unique id automatically

### **Table Name: courselist**

<b>Column Name</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Description</b>
courseld	INT	Primary key	Unique identity of the course
courseAddedDate	DATE		Course added date
courselimage	VARCHAR (255)		Path of image
courseName	VARCHAR (255)		Title of course
coursePrice	INT		Course price
courseDescription	VARCHAR (1000)		Detail to show on same card of course detail

### **Table Name: boughtcourses**

<b>Column Name</b>	<b>Data Type</b>	<b>Constraints</b>	<b>Description</b>
buyingID	INT	Primary key NOT NULL	Unique identity of the course bought
courseld	int	NOT NULL	Unique identity no of the course
courselimage	VARCHAR (255)		Path of image
courseName	VARCHAR (255)		Title of course
coursePrice	Int		At what price it was bought
courseDescription	VARCHAR (255)		Detail of particular course

userEmail	VARCHAR (255)		Who bought the course
-----------	---------------	--	-----------------------

## Summarizing tables Relationship

**User Details table:** It's contains all the basic details of all 3 types of users, its most important details are email, password and user-Type.

**QueryFromForm Table:** It contains the query asked by any user related to any suspicion curiosity, etc. they got while surfing site, finding course, login, logout, batch detail, faculty detail, if they want to post some suggestion, problem related to anything, etc.

This table's form is accessible to all the users and table can only be seen and edited by the admin.

**hibernate\_sequence:** This is auto access table which is accessed by the other table sequence columns on every new entry, and all this is done by hibernate automatically. This table is related to all other tables.

**courselist:** This table is used to store the data related to the courses and accessed by the jsp pages and course buying servlet to fetch and store data in the table of '*bought courses table*', this table works on some of the conditional based codes also in jsp. This table relation with bought courses is dependent on time – mean the data at that time is save for the particular user and if changed will not affect the user saved data.

**Boughtcourses:** This table is used by faculty and student these two users get their bought courses from this table and shows in the jsp respective pages.

## NUMBER OF MODULES AND THEIR DESCRIPTION

01. List of courses with details
02. Admission
03. Any user view list of teachers with their respective subjects
04. List of batches
05. Viewing basic student and students' details
06. Viewing the basic details of teacher and teachers
07. Viewing batch details
08. Secure Login
09. Viewing advanced detail of student and all students
10. Viewing advanced detail of teacher and all faculties
11. Adding more faculty with administration login
12. Deleting resigned faculty and sending mail to them by admin
13. Update staff and student data

## PROCESS LOGIC OF EACH MODULE

The project will be based on a Multi-User approach, which means that multiple users can use the application simultaneously. Each user will be assigned a specific role and will have limited permissions to operate the web application. User details are as below: –

**Administrator** – He will have full privilege to operate the entire system as per the requirements of the system. Only he/she will have permission to add new Login IDs (staffs) to the system, change passwords, change connection settings, and perform any other administrative functions which will be provided by the application.

**Staff** – He/she can view any details with few limitations desired by them regarding students or teacher (other staff) details. They don't have any power to change any data.

### **Secure Login**

This module of this Project will authorize users and block any unauthorized users after checking. With this, we can maintain only authorized login. This will help to maintain the security of the system and institute data.

### **List of courses with details**

This module's business logic takes data from the model and shows it on view on a specified JSP page. It comprises of query language of MySQL with java (used hibernate framework which makes work easier with HQL).

### **Admission**

This module consists of a form on the view page through which the user will interact and fill in all the given details and submit it online with just a single click. Then details will be added to the database is provided to the admission-taking student.

### **Any user view list of teachers with their respective subjects**

This module is interactable through a single button and the data will be visible on the screen with the help of java, hibernate extracting data from database and showing on the JSP page.

### **List of batches**

This module is as simple as other view lists the same process but the query used and data extracted is changed. The most visible change is in their way of presentation.

### **Viewing basic student and students' details**

This module consists of extracting and presenting on the page. It does not have the authority to show all the details of a student like a fee, address, etc.

### **Viewing basic teacher and teachers' details**

This module consists of extracting and presenting on the page. It does not have the authority to show all the details of a faculty like a salary, date of joining, address, etc.

### **Secure Login**

This module is used while login in by authorities and is properly secure, all the data used to check login details are fetched from the database which can't be changed by any unauthorized person. Only the admin or owner of the institute can do so.

### **Viewing advanced detail of student and all students**

This is used by only the owner/admin of the institute. It is different from all other list presenting modules as it fetches all the student detail available in the database without hiding anything from the owner.

### **Viewing advanced detail of teacher and all faculties**

This is used by only the owner/admin of the institute. It is different from all other list presenting modules as it fetches all the student detail available in the database without hiding anything from the owner.

### **Adding more faculty with administration login**

This is used by the owner only. He/she can add new faculty after confirming they're joining in institute. When he/she adds their details in the database, a mail is sent to them as congratulation.

### **Deleting resigned or left faculty and sending mail to them**

Same as adding the only difference is that it gives a warning before deleting the data and then on confirmation it deletes the data with a mail to left faculty. The mail received by the leaving faculty contains 'Thanks for being with us'

### **Updating staff and student data**

This module performs more than one work. First, it takes the enrollment id or any unique id filled in the section and then fetches their data from the database which after successful fetching presents on the screen, and gets an update after editing by the owner.

## **LIMITATION OF THE PROJECT**

- Needs internet connection
- Needs advanced browser which supports new functions added after 2005

## **FUTURE SCOPE OF THE PROJECT**

This project is designed and developed in such a manner that it provides maximum efficiency & speed and has a vast scope for further development. A number of modules can be added without any modifications in the database and with a minimum modification in its code. It is armed with a powerful query support system and is capable of supporting advanced and complex queries for much more advanced processes and reports.

This application fits into the current scenario, which is the Coronavirus and information age. This application with some modifications can be used by other institutes to automate their services, increase their efficiency, and to make their presence felt in the present fast online world.

In the future, I have decided to add more features like a login system for students too, online classes, live chat with teachers, performance, progress, and so on.

All the present and addable features will definitely help the world to run faster and give a better of teaching with data saving.

## BIBLIOGRAPHY

### **WEBSITEs used to get help while creating projects –**

1. [www.google.com](http://www.google.com)
2. <http://www.microsoft.com/en-in/download/>
3. <https://www.javatpoint.com/mysql-tutorial>
4. <https://app.creately.com/diagram/lZYu0470DJU/edit>
5. [https://lucid.app/lucidchart/9e136d93-4545-4cd1-b26f-8bb69ac6484d/edit?beaconFlowId=E7D9D2D1265439AA&page=0\\_0#](https://lucid.app/lucidchart/9e136d93-4545-4cd1-b26f-8bb69ac6484d/edit?beaconFlowId=E7D9D2D1265439AA&page=0_0#)
6. Many channels and videos from youtube.

### **Referenced Books**

1. All books of NIIT (I am a student of NIIT). They taught me:
  - a. Core java
  - b. Advanced java
  - c. RWD
  - d. RDBMS
  - e. MYSQL
  - f. JSP
  - g. HIBERNATE
2. IGNOU Books

referenced books (SQL Server 2005)–

1. Leonard Lobal, Andrew J. Brust, Stephen Forte, *Programming Microsoft SQL Server 2005 (Pro Developer)*, 29 Oct 2008
2. Robert Vieira, *Professional Microsoft SQL Server 2005 Programming*, 23 Aug 2009
3. IGNOU Books

## **Main Functions my project have:**

1. Login and Register Separate Page
2. Home page
  - a. Introduction to the site, login and register button
  - b. Top rankers
  - c. Achievements
  - d. Hyperlink for new courses
  - e. Register and login button on top right corner
3. Admin home page
  - a. Add new faculty
  - b. View all User details,
  - c. View all student details,
  - d. View all faculty,  
Update data of any user
  - e. Deleting any type of user details
4. Course page (explore and buy course of your likened)
  - a. List of courses (*in card view*)
  - b. With click on more detail button it will transfer the user to the new jsp page
    - i. With full detail page of course, we can take it.
5. Admission page
  - a. User gets his/her admission in the online courses for free while creatin an account on the site.
6. Check your courses
  - a. In the home page of student (loads automatic on page load)
  - b. Also, in the home page of faculty.
7. Faculty page
  - a. Check student details (*can't edit any detail*)
  - b. Check faculty details (*can't edit any detail*)
  - c. *Update own detail*

# IGNOU Project Report

On

***Institute Management System***

***or***

***Java Based User-Faculty Institute Workspace  
(Web Application)***

BY  
UJJWAL PANDEY

## **TABLE OF CONTENT of PROJECT REPORT**

1. Introduction	...33
2. Objectives	...34
3. Tools/Environment Used	...35
4. Analysis Document	...36
5. Program Code	...39
6. Tables used (screenshots)	...133
7. Output screens	...138
8. Testing	...197
9. Input-Output	...199
10. Security implementation	...200
11. Limitations of the Project	...201
12. Future Application of the Project	...201
13. Bibliography	...202

## **1. INTRODUCTION:**

Java based user faculty workspace is developed in favor of the Institute management team, which will help them to save, keep and manage the records of the students and faculty about their personal details, courses and other things. It helps them to switch from the manual work to automatic work, which was very difficult to save, keep, maintain, retrieve and update data of student, faculty and administration.

This solution is developed on the plight of the institute management and student data, through this software they can see and update their data whenever they want without visiting to anyone and waiting to be acquaintance by them.

For example, "IGNOU" has 1,00,000+ students and faculty which are doing their courses and jobs respectively from 'IGNOU' but what when they want to check some of their details for any reason? What they find something outdated? What when they want to update some details? What if they want to take admission in new course? The answer of all this question is before this web app they have to do all these things with the help management team, by visiting personally and then go for all the process one by one which was time taking on repeat, and hence there is a lot of strain on the person/team who were running the management to work with papers.

Thus, there are a lot of repetitions which can be easily avoided with the help of my project. This particular project deals with the problems on managing a specific institute works and avoids the problems which occur when carried manually

Identification of the drawbacks of the existing system leads to the designing of computerized system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing system.

## **2. Objective:**

This system is designed for managing various activities in the Institute. For the past few years, the number of educational institutes are increasing rapidly, therefore the number of persons and team are also increasing which have to undergo the manual process of storing, retrieving, and updating data creating a lot of strain on them plus taking much time and wasting it.

So, my software (with high security) can help the in different ways,

- By adding the user through internet without sharing detail with anyone,
- Increase efficiency of students, faculty and admin.
- Strength and strain of manual Labour can be reduced,
- Backup data can be easily generated,
- Data consistency,
- Easy data updating,
- Easy record keeping,
- If the visitor is satisfied with the course they can enroll,
- Search their likely course whenever they want,
- Contact institute faculty without visiting them in personal,
- Easy keep tracking of every user by admin, and
- Bring several functional and non-functional work with site

### **3. Tools/Environment Used to create project**

#### **1. Hardware used to create this project**

- a. Laptop
- b. Books

#### **2. Software Configuration**

- a. Eclipse 2020-03
- b. **Maven project** (help in managing dependencies efficiently)
- c. Font-awesome
- d. Hibernate
- e. MySQL 8.0
- f. Java
- g. Advanced Java
- h. CSS
- i. JavaScript
- j. Browsers (Edge and Firefox)

(These are Tools/ Environment used in project)

## **4. Analysis Document**

The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

The proposed system has the following requirements:

- System needs store information about new User.
- System needs to maintain quantity record
- System needs to keep the record of Courses
- System needs to update the record of Courses
- System needs to update the user type
- System needs to be connected with internet and database in order to work
- System needs secured network to work without interruptions.

## **Scope**

It may help collecting perfect user and course details. In a very short time, the collection will be obvious, simple and suitable. It will help a new admin person to know the management and work of passed year perfectly and vividly. It will help new and current management system in managing the details of everything perfectly and easily just with the help of browser and internet. It will be also reducing the cost of collecting the data, collecting procedure time and give ability to handle more and more courses and user at time. It will go on smoothly.

- The system generates types of information that can be used for various purposes.
- Easy to operate

- Easy to understand by user and operator.

## **Data Dictionary:**

This is normally represented as the data about data. It is also termed as metadata some times which gives the data about the data stored in the database. It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or the processes.

Data dictionary entries:

- Words should be defined to understand for what they need and not the variable need by which they may be described in the program.
- Each word must be unique. We cannot have two definition of the same client.
- Aliases or synonyms are allowed when two or more enters shows the same meaning. For example, a vendor number may also be called as customer number.
- A self-defining word should not be decomposed. It means that the reduction of any information in to subpart should be done only if it is really required that is it is not easy to understand directly.

## **User Interface Design**

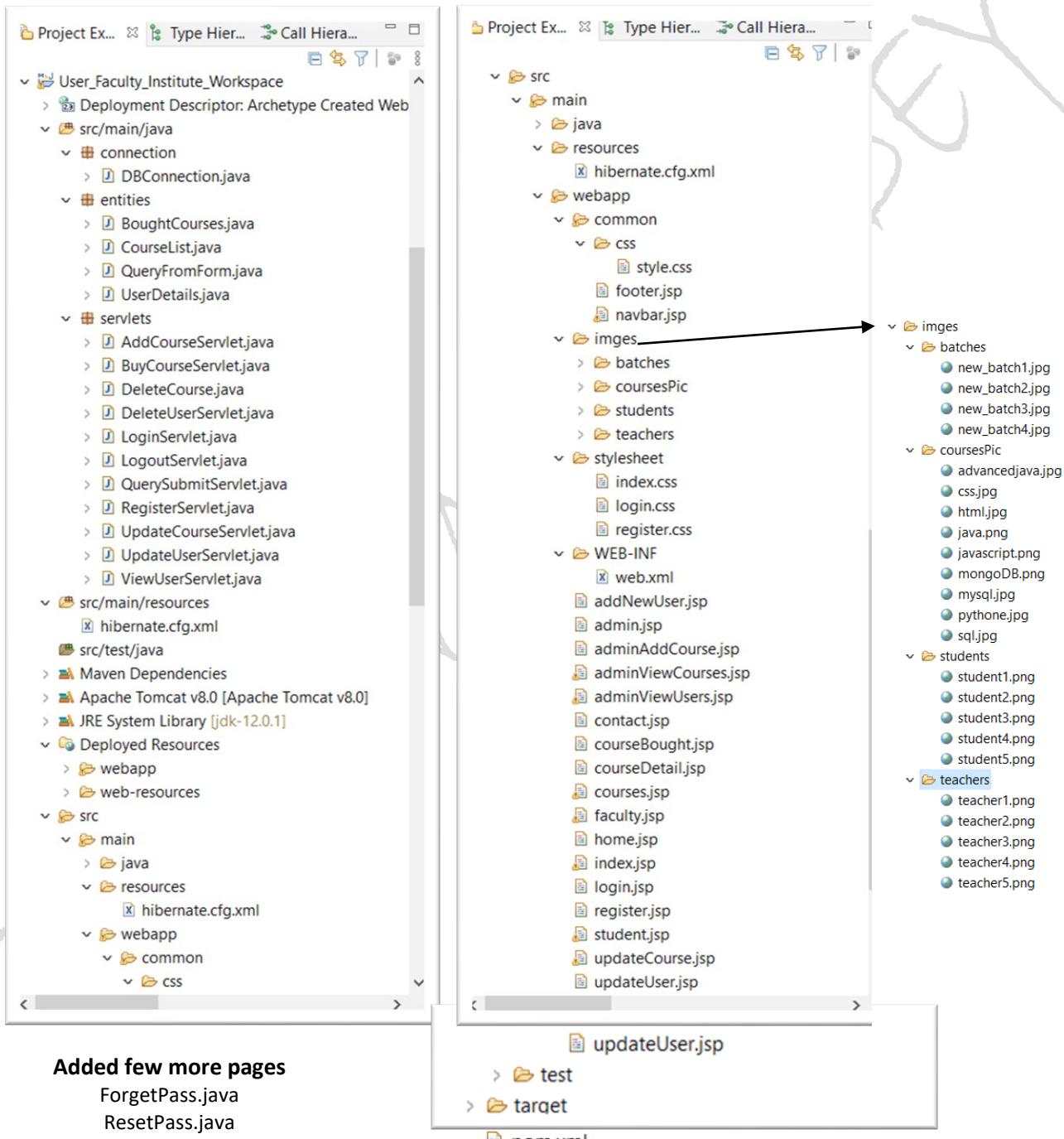
User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. *The overall flow of screens and messages is called a dialogue.*

The following steps are various *guidelines for User Interface Design*:

- 1. The system user should always be aware of what to do next.
- 2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
- 3. Message, instructions or information should be displayed long enough to allow
- the system user to read them.
- 4. Use display attributes sparingly.
- 5. Default values for fields and answers to be entered by the user should be specified.
- 6. A user should not be allowed to proceed without correcting an error

# 5. Program Code

## Basic Structure of my project



**Added few more pages**

ForgetPass.java  
ResetPass.java  
Validate\_code\_source.java  
Payment.jsp  
Forget.jsp

## Codes of 'connection' package

### DBConnection.java

```
package connection;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class DBConnection {
    //default Constructor
    public DBConnection() {
    }
    // I used static so that I can call this function without making an object
    // and calling this method like an function
    public static SessionFactory factory;
    public static SessionFactory getFactory() {
        if (factory == null) {
            factory = new Configuration().configure().buildSessionFactory();
        }
        return factory;
    }
    public void closeFactory() {
        if (factory.isOpen())
            factory.close();
    }
}
```

## Codes of 'entities' package

### BoughtCourses.java

```
package entities;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class BoughtCourses {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int buyingID;
    private int courseID;
    private String courseName;
    private String courseDescription;
    private String courseImage;
    private int coursePrice = 2999;
    private String userEmail;

    public int getBuyingID() {
        return buyingID;
    }

    public void setBuyingID(int buyingID) {
        this.buyingID = buyingID;
    }
}
```

```
public int getCouseID() {  
    return courseID;  
}  
  
public void setCourseID(int courseID) {  
    this.courseID = courseID;  
}  
  
public String getCourseName() {  
    return courseName;  
}  
  
public void setCourseName(String courseName) {  
    this.courseName = courseName;  
}  
  
public int getCoursePrice() {  
    return coursePrice;  
}  
  
public void setCoursePrice(int coursePrice) {  
    this.coursePrice = coursePrice;  
}  
  
public String getCouseDescription() {  
    return courseDescription;  
}  
  
public void setCouseDescription(String courseDescription) {  
    this.courseDescription = courseDescription;  
}  
  
public BoughtCourses() {  
}
```

```
public String getCourseImage() {  
    return courseImage;  
}  
  
public void setCourseImage(String courseImage) {  
    this.courseImage = courseImage;  
}  
  
public String getUserEmail() {  
    return userEmail;  
}  
  
public void setUserEmail(String userEmail) {  
    this.userEmail = userEmail;  
}  
  
public BoughtCourses(int courseId, String courseName,  
    String courseDescription, String courseImage, int coursePrice,  
    String userEmail) {  
    super();  
    this.courseId = courseId;  
    this.courseName = courseName;  
    this.couseDescription = courseDescription;  
    this.courseImage = courseImage;  
    this.coursePrice = coursePrice;  
    this.userEmail = userEmail;  
}
```

### CourseList.java

```
package entities;

import javax.persistence.*;
import java.util.Date;

@Entity
@Table
public class CourseList {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(unique = true)
    private int courseID;
    private String courseName;
    private String courseDescription;
    private String courseImage;
    private int coursePrice = 2999;
    @Temporal(TemporalType.DATE)
    @Column(name = "courseAddedDate")
    private Date courseAddedDate;

    public int getCourseID() {
        return courseID;
    }

    public void setCourseID(int courseID) {
        this.courseID = courseID;
    }
}
```

```
public String getCourseName() {  
    return courseName;  
}  
  
public void setCourseName(String courseName) {  
    this.courseName = courseName;  
}  
  
public String getCouseDescription() {  
    return courseDescription;  
}  
  
public void setCouseDescription(String courseDescription) {  
    this.courseDescription = courseDescription;  
}  
  
public String getCourseImage() {  
    return courseImage;  
}  
  
public void setCourseImage(String courseImagePath) {  
    this.courseImage = courseImagePath;  
}  
  
public int getCoursePrice() {  
    return coursePrice;  
}  
  
public void setCoursePrice(int coursePrice) {  
    this.coursePrice = coursePrice;  
}  
  
public Date getCourseAddedDate() {  
    return courseAddedDate;  
}
```

```
public void setCourseAddedDate(Date courseAddedDate) {  
    this.courseAddedDate = courseAddedDate;  
}  
  
public CourseList() {  
}  
  
public CourseList(String courseName, String courseDescription,  
    String courseImage, int coursePrice, Date courseAddedDate) {  
    super();  
    this.courseName = courseName;  
    this.courseDescription = courseDescription;  
    this.courseImage = courseImage;  
    this.coursePrice = coursePrice;  
    this.courseAddedDate = courseAddedDate;  
}  
  
public CourseList(String courseName, String courseDescription, int coursePrice) {  
    super();  
    this.courseName = courseName;  
    this.courseDescription = courseDescription;  
    this.coursePrice = coursePrice;  
}  
  
@Override  
public String toString() {  
    return "CourseList [courseID=" + courseID + ", courseName=" + courseName  
        + ", courseDescription=" + courseDescription + ", courseImage="  
        + courseImage + ", coursePrice=" + coursePrice + "]";  
}  
}
```

### QueryFromQuery.java

```
package entities;

import javax.persistence.*;

@Entity
@Table
public class QueryFromForm {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(nullable = false)
    private int qSoNo;
    private String qName;
    private String qEmail;
    private String qContact;
    private String qQueryHead;
    @Column(columnDefinition = "TEXT")
    private String qQueryExplained;

    public int getqSoNo() {
        return qSoNo;
    }

    public void setqSoNo(int qSoNo) {
        this.qSoNo = qSoNo;
    }

    public String getqName() {
        return qName;
    }

    public void setqName(String qName) {
        this.qName = qName;
    }

    public String getqEmail() {
        return qEmail;
    }

    public void setqEmail(String qEmail) {
        this.qEmail = qEmail;
    }

    public String getqContact() {
        return qContact;
    }
}
```

```
}

public void setqContact(String qContact) {
    this.qContact = qContact;
}

public String getqQueryHead() {
    return qQueryHead;
}

public void setqQueryHead(String qQueryHead) {
    this.qQueryHead = qQueryHead;
}
public String getqQueryExplained() {
    return qQueryExplained;
}

public void setqQueryExplained(String qQueryExplained) {
    this.qQueryExplained = qQueryExplained;
}

public QueryFromForm() {
    super();
}

public QueryFromForm(String qName, String qEmail, String qContact,
                     String qQueryHead, String qQueryExplained) {
    super();
    this.qName = qName;
    this.qEmail = qEmail;
    this.qContact = qContact;
    this.qQueryHead = qQueryHead;
    this.qQueryExplained = qQueryExplained;
}

@Override
public String toString() {
    return "QueryFromForm [qSoNo=" + qSoNo + ", qName=" + qName
           + ", qEmail=" + qEmail + ", qContact=" + qContact
           + ", qQueryHead=" + qQueryHead + ", qQueryExplained="
           + qQueryExplained + "]";
}
```

### UserDetails.java

```
package entities;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;
import java.text.SimpleDateFormat;
import java.util.Date;

@Entity
@Table
public class UserDetails {

    @Id
    @Column(nullable = false, unique = true)
    private String email;
    @Column(nullable = false, length = 20)
    private String firstName;
    @Column(nullable = true, length = 20)
    private String lastName;
    @Column(nullable = false)
    private String password;
    @Column(nullable = true)
    private String contactNumber;
    @Column(nullable = false)
    private String registrationDate;
    @Column(nullable = false)
```

```
private String userType;  
  
@Column(nullable = true, length = 15)  
  
private String houseNo;  
  
@Column(nullable = true)  
  
private String landMark;  
  
@Column(nullable = true, length = 35)  
  
private String city;  
  
@Column(nullable = true)  
  
private String state;  
  
@Column(nullable = true)  
  
private String pinCode;  
  
  
public UserDetails() {  
    super();  
}  
  
public String getHouseNo() {  
    return houseNo;  
}  
  
public void setHouseNo(String houseNo) {  
    this.houseNo = houseNo;  
}  
  
public String getLandMark() {  
    return landMark;  
}  
  
public void setLandMark(String landMark) {  
    this.landMark = landMark;  
}
```

```
public String getCity() {  
    return city;  
}  
  
public void setCity(String city) {  
    this.city = city;  
}  
  
public String getState() {  
    return state;  
}  
  
public void setState(String state) {  
    this.state = state;  
}  
  
public String getPinCode() {  
    return pinCode;  
}  
  
public void setPinCode(String pinCode) {  
    this.pinCode = pinCode;  
}  
  
public String getFirstName() {  
    return firstName;  
}  
  
public void setFirstName(String firstName) {  
    this.firstName = firstName;  
}  
  
public String getLastName() {  
    return lastName;  
}
```

```
public void setLastName(String lastName) {  
    this.lastName = lastName;  
}  
  
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;  
}  
  
public String getContactNumber() {  
    return contactNumber;  
}  
  
public void setContactNumber(String contactNumber) {  
    this.contactNumber = contactNumber;  
}  
  
public String getUserType() {  
    return userType;  
}  
  
public void setUserType(String userType) {  
    this.userType = userType;  
}
```

```
}

public String getRegistrationDate() {

    return registrationDate;

}

public void setRegistrationDate(Date registerDate) {

    SimpleDateFormat formatter = new SimpleDateFormat(
        "dd-MM-yyyy, HH:mm:ss");

    this.registrationDate = formatter.format(registerDate);

}

public UserDetails(String firstName, String lastName, String email,
    String password, Date registrationDate, String userType) {

    super();

    this.firstName = firstName;

    this.lastName = lastName;

    this.email = email;

    this.password = password;

    this.userType = userType;

    this.setRegistrationDate(registrationDate);

    this.registrationDate = this.getRegistrationDate();

}

public UserDetails(String email, String firstName, String lastName,
    String password, String contactNumber, String userType, String houseNo,
    String landMark, String city, String state, String pinCode) {

    super();

    this.email = email;

    this.firstName = firstName;
```

```
        this.lastName = lastName;  
  
        this.password = password;  
  
        this.contactNumber = contactNumber;  
  
        this.userType = userType;  
  
        this.houseNo = houseNo;  
  
        this.landMark = landMark;  
  
        this.city = city;  
  
        this.state = state;  
  
        this.pinCode = pinCode;  
  
    }  
  
    public String getFullName() {  
  
        return firstName + " " + lastName;  
  
    }  
  
    public String toStringFullAddress() {  
  
        return "Address:- " + getHouseNo() + ", " + getLandMark() + ", "  
               + getCity() + ", " + getState() + " " + getPinCode();  
  
    }  
  
    @Override  
  
    public String toString() {  
  
        return "UserDetails [email=" + email + ", firstName=" + firstName  
               + ", lastName=" + lastName + ", password=" + password  
               + ", contactNumber=" + contactNumber + ", regisraionDate="  
               + regisraionDate + ", userType=" + userType + ", houseNo="  
               + houseNo + ", landMark=" + landMark + ", city=" + city  
               + ", state=" + state + ", pinCode=" + pinCode + "]";  
  
    }  
}
```

## Codes of 'servlets' package

### AddCourseServlet.java

```
package servlets;

import connection.DBConnection;
import entities.CourseList;
import org.hibernate.Session;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.util.Date;

public class AddCourseServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public AddCourseServlet() {
        super();
    }
    protected void doPost(HttpServletRequest request,
                         HttpServletResponse response) throws ServletException, IOException {
        String title = (String) request.getParameter("cTitle");
        String description = (String) request.getParameter("cExplanation");
    }
}
```

```
String image = (String) request.getParameter("subjectImage");

int price = Integer.parseInt(request.getParameter("cPrice"));

CourseList cl;

cl = new CourseList(title, description, image, price, new Date());

//System.out.println(cl.toString());

Session sess = DBConnection.getFactory().openSession();

sess.beginTransaction();

sess.save(cl);

sess.getTransaction().commit();

sess.close();

HttpSession session = null;

session = request.getSession();

session.setAttribute("course-added", "New course " + title + " added");

response.sendRedirect("admin.jsp");

}

}
```

### BuyCourseServlet.java

```
package servlets;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;
```

```
import org.hibernate.Session;
import org.hibernate.Transaction;
import connection.DBConnection;
import entities.BoughtCourses;
import entities.CourseList;

public class BuyCourseServlet extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public BuyCourseServlet() {
        super();
    }

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }

    protected void doGet(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {
        String userId = request.getParameter("userId");
        HttpSession session = null;
        session = request.getSession();
        CourseList cl = (CourseList) session.getAttribute("courseDetails");
        try {
            Session sess = DBConnection.getFactory().openSession();
            Transaction tr = sess.beginTransaction();
            BoughtCourses bcCourse = new BoughtCourses(cl.getCourseID(),
                                                       cl.getCourseName(), cl.getCouseDescription(),
                                                       cl.getCourseImage(), cl.getCoursePrice(), userId);
            sess.save(bcCourse);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
        tr.commit();

        sess.close();

    } catch (Exception e) {

        e.printStackTrace();

    }

    session.setAttribute("CourseBought",
        "Course '" + cl.getCourseName() + "' bought successfully.");

    response.sendRedirect("courses.jsp");

}

}
```

### DeleteCourse.java

```
package servlets;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.hibernate.Session;
import connection.DBConnection;
import entities.CourseList;
```

```
public class DeleteCourse extends HttpServlet {  
  
    private static final long serialVersionUID = 1L;  
  
    public DeleteCourse() {  
  
        super();  
    }  
  
    protected void doGet(HttpServletRequest request,  
                         HttpServletResponse response) throws ServletException, IOException {  
  
        Session sess = DBConnection.getFactory().openSession();  
  
        int cld = Integer.parseInt(request.getParameter("courseID"));  
  
        sess.beginTransaction();  
  
        CourseList cl = new CourseList();  
  
        cl.setCourseID(cld);  
  
        sess.delete(cl);  
  
        sess.getTransaction().commit();  
  
        sess.close();  
  
        HttpSession session = null;  
  
        session = request.getSession();  
  
        session.setAttribute("course-delete-success",  
                            "Course with course id - " + cld + " deleted successfully");  
  
        response.sendRedirect("adminViewCourses.jsp");  
    }  
}
```

### DeleteUserServlet.java

```
package servlets;

import connection.DBConnection;
import entities.UserDetails;
import org.hibernate.Session;
import org.hibernate.Transaction;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;

public class DeleteUserServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public DeleteUserServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request,
                         HttpServletResponse response) throws ServletException, IOException {
        try {
            String email = request.getParameter("courseID");
            Session sess = DBConnection.getFactory().openSession();
            Transaction tr = sess.beginTransaction();
            UserDetails userDelete = new UserDetails();
        }
    }
}
```

```
userDelete.setEmail(email);

int check = 10;

UserDetails persist = sess.load(UserDetails.class, email);

String utype2 = persist.getUserType();

if (persist != null) {

    sess.delete(persist);

    check = 1;

}

tr.commit();

sess.close();

HttpSession session = null;

if (check == 1) {

    session = request.getSession();

    session.setAttribute("Users-Success", "User with email: " +
        + email + " deleted successfully!");

} else {

    session = request.getSession();

    session.setAttribute("Users-Success", "User with email: " +
        + email + " not deleted successfully!");

}

response.sendRedirect("adminViewUsers.jsp?whoUser=" + utype2 + "");

// response.sendRedirect("adminViewCourses.jsp");

System.out.println("Redirected succes.");

} catch (Exception e) {

    e.printStackTrace();

}

}
```

### LoginServlet.java

```
package servlets;

import connection.DBConnection;
import entities.UserDetails;
import org.hibernate.Session;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;

@WebServlet
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public LoginServlet() {
        super();
    }

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {
        String email = request.getParameter("uEmail");
        String password = request.getParameter("uPassword");
    }
}
```

```
// no need of transaction as we are not saving anything.

Session sess = DBConnection.getFactory().openSession();

UserDetails userDetails = (UserDetails) sess.get(UserDetails.class,
    email);

HttpSession logedInDetails = null;

if (userDetails != null) {

    if (userDetails.getPassword().equals(password)) {

        logedInDetails = request.getSession();
        logedInDetails.setAttribute("logedInUser", userDetails);

        if (userDetails.getUserType().equals("student")) {

            response.sendRedirect("student.jsp");

        } else if (userDetails.getUserType().equals("faculty"))

            response.sendRedirect("faculty.jsp");

        else if (userDetails.getUserType().equals("admin"))

            response.sendRedirect("admin.jsp");

    } else {

        sess.getTransaction().rollback();
        logedInDetails = request.getSession();
        logedInDetails.setAttribute("login-failed",
            "Wronge Password try again...");
        response.sendRedirect("login.jsp");
    }
} else {

    logedInDetails = request.getSession();
    logedInDetails.setAttribute("login-failed",
        "This user does not exists!!!");
    response.sendRedirect("login.jsp");
}
```

```
System.out.println(
```

"Database returned object is empty\n Now we need to create a temp session and send use it to show an alert in the login page that the details does not match. wrong mail id password.");

```
    }  
}  
}
```

### LogoutServlet.java

```
package servlets;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import java.io.IOException;  
  
public class LogoutServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
    public LogoutServlet() {  
        super();  
    }  
  
    protected void doPost(HttpServletRequest request,  
                         HttpServletResponse response) throws ServletException, IOException {  
        try {
```

```
HttpSession session = request.getSession();
session.removeAttribute("logedInUser");
session.setAttribute("logout-msg", "Logout sucessfull...");
response.sendRedirect("login.jsp");

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

}

protected void doGet(HttpServletRequest request,
                      HttpServletResponse response) throws ServletException, IOException {
try {
    HttpSession session = request.getSession();
    session.removeAttribute("logedInUser");
    session.setAttribute("logout-msg", "Logout sucessfull...");
    response.sendRedirect("login.jsp");

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

}
```

### QuerySubmitServlet.java

```
package servlets;

import connection.DBConnection;
import entities.QueryFromForm;
import org.hibernate.Session;
import org.hibernate.Transaction;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;

public class QuerySubmitServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public QuerySubmitServlet() {
        super();
    }

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String contact = request.getParameter("contact");
        String queryHeading = request.getParameter("queryHead");
        String query = request.getParameter("queryExplained");
    }
}
```

```
QueryFromForm q = new QueryFromForm(name, email, contact, queryHeading,  
query);  
  
Session session = DBConnection.getFactory().openSession();  
  
Transaction tr = session.beginTransaction();  
  
session.save(q);  
  
HttpSession sessionh = request.getSession();  
  
sessionh.setAttribute("querySubmitted",  
"Query submitted we will reach you soon.");  
  
response.sendRedirect("contact.jsp");  
  
tr.commit();  
  
session.close();  
  
}  
}
```

### RegisterServlet.java

```
package servlets;  
  
import java.io.IOException;  
import java.io.Serializable;  
import java.util.Date;  
import java.util.Properties;  
  
import javax.mail.Authenticator;
```

```
import javax.mail.Message;  
import javax.mail.MessagingException;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Transport;  
import javax.mail.internet.InternetAddress;  
import javax.mail.internet.MimeMessage;  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
  
import org.apache.commons.validator.routines.EmailValidator;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
import org.hibernate.exception.ConstraintViolationException;  
  
import connection.DBConnection;  
import entities.UserDetails;  
  
public class RegisterServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    public RegisterServlet() {  
        super();  
    }  
  
    public static boolean validateEmailApache(String email) {
```

```
email = email.trim();

EmailValidator eValidator = EmailValidator.getInstance();

if (eValidator.isValid(email)) {

    return true;

} else {

    return false;

}

protected void doPost(HttpServletRequest request,
                      HttpServletResponse response) throws ServletException, IOException {

HttpSession session = null;

try {

    String uFirstName = request.getParameter("uFirstName");

    String uLastName = request.getParameter("uLastName");

    String uEmail = request.getParameter("uEmail");

    String uPassword = request.getParameter("uPassword");

    String uType = request.getParameter("uType");

    boolean validateEmailApache = validateEmailApache(uEmail);

    if (!validateEmailApache) {

        session = request.getSession();

        session.setAttribute("reg-failed",
                            "Registration Unsucessful, Given mail id is not valid email address.");

        response.sendRedirect("register.jsp");

    } else {

        Session sess = DBConnection.getFactory().openSession();

        UserDetails userDetails = (UserDetails) sess
```

```
.get(UserDetails.class, uEmail);

if (userDetails != null) {

    session = request.getSession();

    session.setAttribute("reg-failed", "This user email "

        + uEmail

        + "' is already in use take something else.");

    response.sendRedirect("register.jsp");

} else {

    UserDetails uDetails;

    if (uType != null) {

        uDetails = new UserDetails(uFirstName, uLastName,

            uEmail, uPassword, new Date(), uType);

        // DBConnection gives the factory.

        // Create session from the factory.

        // Create a transaction from the session.

        Transaction tr = sess.beginTransaction();

        // or sess.beginTransaction();

        Serializable iddd = sess.save(uDetails);

        System.out.println(iddd);

        tr.commit();

        // or sess.getTransaction().commit();

        sess.close();
    }
}
```

```
session = request.getSession();
session.setAttribute("reg-sucess",
                    "New '" + uType + "' Registered Sucesfull");
response.sendRedirect("admin.jsp");
sendmail(uEmail, uPassword,
          "new ".concat(uType).concat(" - ")
          .concat(uLastName).concat(uLastName));
} else {
    uDetails = new UserDetails(uFirstName, uLastName,
                               uEmail, uPassword, new Date(), "student");
    // DBConnection gives the factory.
    // Create session from the factory.
    // Create a transaction from the session.
    Transaction tr = sess.beginTransaction();
    // or sess.beginTransaction();
    Serializable iddd = sess.save(uDetails);
    System.out.println(iddd);

    tr.commit();
    // or sess.getTransaction().commit();

    sess.close();
}

session = request.getSession();
```

```
session.setAttribute("reg-sucess",
    "Registration Sucesfull");
response.sendRedirect("register.jsp");
sendmail(uEmail, uPassword,
    uFirstName.concat(uLastName));
}

}

} catch (ConstraintViolationException cve) {
    System.out.println("Duplicate maal");
} catch (Exception e) {
    System.out.println("2nd exception is thrown\n");
    e.printStackTrace();
    session = request.getSession();
    // Will execute when connection with database is failed due to
    // various reason.
    session.setAttribute("reg-failed",
        "Registration Unsuccesful, Something went wrong with server!! Come back in 10
min....");
    response.sendRedirect("register.jsp");
}

}

public void sendmail(String recipient, String password, String name)
throws MessagingException {
Properties prop = new Properties();
prop.setProperty("mail.smtp.auth", "true");
prop.setProperty("mail.smtp.starttls.enable", "true");
```

```
prop.setProperty("mail.smtp.host", "smtp.gmail.com");

prop.setProperty("mail.smtp.port", "587");

// Sender

final String myAddress = "ujjwalpandey.apps@gmail.com";

final String pass = "Goldenstar@123";

javax.mail.Session sess = javax.mail.Session.getInstance(prop,
    new Authenticator() {
        @Override
        protected PasswordAuthentication getPasswordAuthentication() {
            return new PasswordAuthentication(myAddress, pass);
        }
    });
Message mess = prepareMessage(sess, myAddress, recipient, password,
    name);
Transport.send(mess);

}

private static Message prepareMessage(javax.mail.Session sess,
    String myAddress, String recipient, String pass, String name) {
    try {
        Message mess = new MimeMessage(sess);
        mess.setFrom(new InternetAddress(myAddress));
        mess.setRecipient(Message.RecipientType.TO,
            new InternetAddress(recipient));
        mess.setSubject("Welcome to Developers' Point ");
        // mess.setText("This is going to be the body.");
    }
}
```

```
String mailBody = "<h1> Welcome " + name
+ " at Developers' Point </h1>"
+ "<center> Initial detail of your account with mail id: "
+ recipient + " are: " + "<br> Name: <b>" + name + "</b>"
+ "<br> Password: <b>" + pass + "</b>"
+ "<br> <a href='http://localhost:8081/User_Faculty_Institute_Workspace/login.jsp'>click here</a/> "
+ "to login.<br>"
+ "<b>Tip: Change password every 4 month to stay secure.</b>"
+ "</center>";

mess.setContent(mailBody, "text/html");

return mess;

} catch (MessagingException e) {
e.printStackTrace();
System.out.println("Some extra");
}

return null;
}
```

### UpdateCourseServlet.java

```
package servlets;

import connection.DBConnection;
import entities.CourseList;
import org.hibernate.Session;
```

```
import org.hibernate.Transaction;  
  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.servlet.http.HttpSession;  
import java.io.IOException;  
import java.util.Date;  
  
public class UpdateCourseServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    public UpdateCourseServlet() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    protected void doPost(HttpServletRequest request,  
                          HttpServletResponse response) throws ServletException, IOException {  
        int id = Integer.parseInt(request.getParameter("couresID"));  
        String title = request.getParameter("cTitle");  
        String descr = request.getParameter("cExplanation");  
        int price = Integer.parseInt(request.getParameter("cPrice"));  
        int cDiscount = Integer.parseInt(request.getParameter("cDiscount"));  
        String stuLevel = request.getParameter("stuLevel");  
    }  
}
```

```
String cLearnTime = request.getParameter("cLearnTime");

String cCreator = request.getParameter("cCreator");

String cRequirements = request.getParameter("cRequirements");

int cGurDays = Integer.parseInt(request.getParameter("cGurDays"));

String cLang = request.getParameter("cLang");

String cLanguage = request.getParameter("cLanguage");

String subjectImage = request.getParameter("subjectImage");

Session sess = DBConnection.getFactory().openSession();

Transaction tr = sess.beginTransaction();

CourseList updateIt = sess.get(CourseList.class, id);

updateIt.setCourseName(title);

updateIt.setCouseDescription(descr);

updateIt.setCoursePrice(price);

updateIt.setDiscount(cDiscount);

updateIt.setLevel(stuLevel);

updateIt.setLearnTime(cLearnTime);

updateIt.setCreatorName(cCreator);

updateIt.setRequirements(cRequirements);

updateIt.setMoneyBackGuaranteeDays(cGurDays);

updateIt.setLang(cLang);

updateIt.setLanguage(cLanguage);

updateIt.setCourseImage(subjectImage);

updateIt.setUpdatedCourseDate(new Date());

System.out.println(updateIt);

tr.commit();
```

```
sess.close();

HttpSession session = request.getSession();
session.setAttribute("course-delete-success",
"Course with id: " + id + " updated successfully.");
response.sendRedirect("adminViewCourses.jsp");

}

}
```

### UpdateUserServlet.java

```
package servlets;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.hibernate.Session;
import org.hibernate.Transaction;
import connection.DBConnection;
```

```
import entities.UserDetails;

public class UpdateUserServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public UpdateUserServlet() {
        super();
    }

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {
        String fName = request.getParameter("uFirstName");
        String lName = request.getParameter("uLastName");
        String uEmail = request.getParameter("uEmail");
        String password = request.getParameter("uPassword");
        String uType = request.getParameter("uType");
        String uContact = request.getParameter("uContact");
        String uHouse = request.getParameter("uHouse");
        String uLandMark = request.getParameter("uLandmark");
        String uCity = request.getParameter("uCity");
        String uState = request.getParameter("uState");
        String uPinCode = request.getParameter("uPinCode");

        try {
            Session sess = DBConnection.getFactory().openSession();
            Transaction tr = sess.beginTransaction();

```

```
UserDetails udu = new UserDetails();  
  
udu = (UserDetails) sess.get(UserDetails.class, uEmail);  
  
udu.setEmail(uEmail);  
udu.setFirstName(fName);  
udu.setLastName(lName);  
  
udu.setPassword(password);  
udu.setContactNumber(uContact);  
udu.setUserType(uType);  
udu.setHouseNo(uHouse);  
udu.setLandMark(uLandMark);  
udu.setCity(uCity);  
udu.setState(uState);  
udu.setPinCode(uPinCode);  
  
sess.update(udu);  
System.out.println("updated");  
tr.commit();  
sess.close();  
  
String previousPageUrl = request.getParameter("previousPageUrl");  
response.sendRedirect(previousPageUrl);  
//System.out.println(previousPageUrl);  
  
} catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

}

### ViewUserServlet.java

```
package servlets;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class ViewUserServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public ViewUserServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request,
                         HttpServletResponse response) throws ServletException, IOException {
        System.out.println("Inside doGet");
        String whoUser = request.getParameter("whoUser");
    }
}
```

```
// Data from UserDetail.java

if (whoUser.equals("student")) {
    response.sendRedirect("student.jsp");
} else if (whoUser.equals("faculty")) {
    response.sendRedirect("faculty.jsp");
} else {
    response.sendRedirect("admin.jsp");
}
}
```

## Codes of 'main/resources' folder

### hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD//EN"
        "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>
        <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
        <property
name="connection.url ">jdbc:mysql://localhost:3306/institute_workspace</property>
        <!-- Table Name (Create yourself) -->
        <property name="connection.username">root</property>
        <property name="connection.password">Goldenstar@1234</property>
        <property name="dialect">org.hibernate.dialect.MySQL8Dialect</property>
        <property name="hbm2ddl.auto">update</property>

        <!-- <property name="show_sql">true</property> -->
        <property name="format_sql">true</property>
        <!-- POJO to create classes -->
        <mapping class="entities.UserDetails"/>
    </session-factory>
</hibernate-configuration>
```

```
<mapping class="entities.CourseList"/>
<mapping class="entities.BoughtCourses"/>
<mapping class="entities.QueryFromForm"/>

<!-- File where annotation is performed. --&gt;
<!-- &lt;mapping resource = "hibernate.student"/&gt; --&gt;

&lt;!-- !IMPORTANT:- resource attribute is used in mapping option when using
      hibernate.hbm.xml file form mapping. class attribute is used in mapping option
      when using annotation for mapping. --&gt;

&lt;/session-factory&gt;
&lt;/hibernate-configuration&gt;</pre>
```

## Codes of 'webapp/common' folder

### CSS/style.css

```
@charset "ISO-8859-1";
/*
   Navigation bar and Footer Styling.
*/
* {
    box-sizing: border-box;
    margin: 0px;
}
form {
    width: 100%;
    padding: 1rem 3rem;
}
form .form-group {
    padding: 2rem 0rem;
}
```

```
.add-shadow>div {  
    box-shadow: 1px 1px 3px black;  
}  
  
.bg-custom {  
    background: #a446d7;  
}  
  
.btn-custom {  
    /*background: #ef3bef;*/  
    background: white;  
}  
  
.navbar .nav-item .nav-link {  
    font-size: 1rem;  
    color: white;  
}  
  
body {  
    font-family: sans-serif;  
    margin: 0;  
    background: #fbfbfb;  
}  
  
h1 {  
    text-align: center;  
    margin-top: 50px;  
}  
  
p {  
    text-align: center;  
    margin-bottom: 60px;  
}  
  
h4 {  
    text-align: center;  
    line-height: 80px;  
    font-weight: normal;  
}  
  
.masonry { /* Masonry container */  
    -webkit-column-count: 4;  
    -moz-column-count: 4;  
    column-count: 4;  
    margin: 1.5em;  
    padding: 0;  
    -moz-column-gap: 1.5em;  
    -webkit-column-gap: 1.5em;  
    column-gap: 1.5em;  
    font-size: .85em;  
}  
  
.item {  
    display: inline-table;  
}
```

```
background: #fff;
padding: 1em;
margin: 0 0 1.5em;
width: 100%;
transition: 1s ease all;
-webkit-transition: 1s ease all;
box-sizing: border-box;
-moz-box-sizing: border-box;
-webkit-box-sizing: border-box;
box-shadow: 2px 2px 4px 0 #ccc;
}

.item img {
    max-width: 100%;
    height: auto;
}

.down {

}

@media only screen and (max-width: 320px) {
    .masonry {
        -moz-column-count: 1;
        -webkit-column-count: 1;
        column-count: 1;
    }
}

@media only screen and (min-width: 321px) and (max-width: 768px) {
    .masonry {
        -moz-column-count: 2;
        -webkit-column-count: 2;
        column-count: 2;
    }
}

@media only screen and (min-width: 769px) and (max-width: 1200px) {
    .masonry {
        -moz-column-count: 3;
        -webkit-column-count: 3;
        column-count: 3;
    }
}

@media only screen and (min-width: 1201px) {
    .masonry {
        -moz-column-count: 4;
        -webkit-column-count: 4;
        column-count: 4;
    }
}
```

### **footer.jsp**

```
<style>
.down {
    transition: all .35s ease;
}

.down:hover {
    margin-bottom: -10px;
    box-shadow: 1px 2px 2px black;
    transform: scale(1.2);
}
</style>
<!-- Footer -->
<footer class="bg-light text-center text-lg-start">
    <div class="text-center p-3"
        style="background-color: rgba(0, 0, 0, 0.2);">
        <!-- Facebook -->
        <a class="down btn btn-primary rounded-circle"
            style="background-color: #3b5998"
            href="https://www.facebook.com/ujjwal.pandey.1656/" target="_blank" role="button"><i
                class="fab fa-facebook"></i></a>

        <!-- Twitter -->
        <a class="down btn btn-primary rounded-circle"
            style="background-color: #55acee"
            href="https://twitter.com/Ujjwalp13341664" target="_blank" role="button"><i
                class="fab fa-twitter"></i></a>

        <!-- Instagram -->
        <a class="down btn btn-primary rounded-circle"
            style="background-color: #ac2bac"
            href="https://www.instagram.com/pandey.ujjwalpandey/" target="_blank" role="button"><i
                class="fab fa-instagram"></i></a>

        <!-- Linkedin -->
        <a class="down btn btn-primary rounded-circle"
            style="background-color: #0082ca" href="https://www.linkedin.com/in/ujjwal-
            pandey-8bb562138/" target="_blank" role="button"><i
                class="fab fa-linkedin-in"></i></a>

        <!-- Stack overflow -->
        <a class="down btn btn-primary rounded-circle"
            style="background-color: #fffac4"
            href="https://stackoverflow.com/users/12213190/ujjwal-pandey" target="_blank" role="button"><i
                class="fab fa-stack-overflow"></i></a>

        <!-- Github -->
    </div>
</footer>
```

```
<a class="down btn btn-primary rounded-circle"
    style="background-color: #333333"
    href="https://github.com/ujjwalpandeyjava" target="_blank" role="button"><i
        class="fab fa-github"></i></a><br> <br> © 2020 Copyright: <span
        class="text-dark">Developers Point </span>
    </div>
</footer>
<!-- Footer ends here --&gt;

<!-- Bootstrap 5 JavaScript and Popper.js --&gt;
&lt;script src="webjars/bootstrap/5.0.0-beta3/js/bootstrap.min.js"&gt;&lt;/script&gt;
&lt;script src="webjars/popper.js/2.5.4/umd/popper.min.js"&gt;&lt;/script&gt;</pre>
```

**navbar.jsp**

```
<!-- Bootstrap 5 CDN Dependencies -->
<%@page import="entities.UserDetails"%>
<link href="webjars/bootstrap/5.0.0-beta3/css/bootstrap.min.css"
      rel="stylesheet">
<!-- Font-awesome CDN Dependencies -->
<link href="webjars/font-awesome/5.15.2/css/all.min.css"
      rel="stylesheet">
<!--
<link
      href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"
      rel="stylesheet">-->
<!-- Custom -->
<link rel="stylesheet" href="common/css/style.css">
<link rel="image/x-icon" href="fa fa-icon">
<!-- Navigation Bar starts here -->
<nav class="px-3 navbar navbar-expand-lg navbar-dark bg-custom">
    <div class="container-fluid">
        <a class="navbar-brand fst-italic" href="index.jsp"><i
            class="fa fa-bookmark"> Developers' Point</i></a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
               data-bs-target="#navbarSupportedContent"
               aria-controls="navbarSupportedContent" aria-expanded="false"
               aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <%
                    UserDetails userDetails = (UserDetails)
session.getAttribute("logedInUser");
                    if (userDetails != null) {
```

```
%>
<li class="nav-item"><a class="nav-link" href="home.jsp" > <i
    class="fa fa-home" aria-hidden="true"> Home</i>
</a></li>
<%
}
%>
<li class="nav-item"><a class="nav-link" aria-current="page"
    href="courses.jsp"><i class="fa fa-user-graduate">
Courses</i></a></li>
<!-- <li class="nav-item"><a class="nav-link" href="newBatches.jsp">
    <i class="fa fa-plus" aria-hidden="true"> New
Batches</i> -->
</a>
</li>
<li class="nav-item"><a class="nav-link" href="contact.jsp" >
    <i class="fa fa-shopping-basket"> Contact</i>
</a></li>
</ul>

<%
if (userDetails != null) {
%>
<div class="d-flex">
    <a class="btn btn-custom btn-outline-dark mx-1"
        data-bs-toggle="modal" data-bs-target="#exampleModal"><i
        class="fa fa-user-circle"> <%=userDetails.getFullName()%></i>
    </a> <a
        href="LogoutServlet" class="btn btn-custom btn-outline-dark
        mx-1"><i
        class="fa fa-sign-out-alt"> logout</i> </a>
</div>
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1"
    aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title text-center fs-2"
                    id="exampleModalLabel">Profile
            <button type="button" class="btn-close" data-bs-
                dismiss="modal"
                aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <table class="table table-striped text-center">
                    <thead>
                        <tr>
                            <th scope="col"
                                colspan="2"><i
                                class="fa fa-user-circle fs-4"> DETAILS</i></th>
                        </tr>
                    </thead>
                    <tbody>
                        <tr>
                            <td>Name:</td>
                            <td> <%=userDetails.getFullName()%></td>
                        </tr>
                        <tr>
                            <td>Email:</td>
                            <td> <%=userDetails.getEmail()%></td>
                        </tr>
                        <tr>
                            <td>Phone:</td>
                            <td> <%=userDetails.getPhone()%></td>
                        </tr>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
</div>
```

```
ID:</th>

<td><%=userDetails.getEmail()%></td>

<td><%=userDetails.getFirstName()%></td>

since:</th>

<td><%=userDetails.getRegistrationDate()%></td>

userDetails.getPassword();

char[pass.length()];

pass.length(); i++) {

%></td>

<th scope="row">Email
</th>
<th scope="row">Name:</th>
<th scope="row">User
<th scope="row">Password</th>
<td colspan="2">
String passStar = "";
String pass =
char[] ch = new
for (int i = 0; i <
passStar += "*";
}/*
</td>
</tr>-->
</tbody>
</table>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Close</button>
<a class="btn btn-info px-3 w-100" role="button"
href="updateUser.jsp?userId=<%=userDetails.getEmail()%>">Edit</a>
</div>
</div>
<%>
} else {
%>
<div class="d-flex">
```

```
<1"><i href="register.jsp" class="btn btn-custom btn-outline-dark mx-1"><i class="fa fa-user-plus"> Register</i> <a href="Login.jsp" class="btn btn-custom btn-outline-dark mx-1"><i class="fa fa-user-circle"> login</i> </a>
      </div>
    <%>
  }
%>
</div>
</div>
</nav>
<!-- Navigation Bar ends here --&gt;</pre>
```

## Codes of 'webapp/stylesheet' folder

### Index.css

```
/*Font */
@import url("https://fonts.googleapis.com/css2?family=Belleza&display=swap");
/*
  font-family: 'Belleza', sans-serif;
*/
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

.primary-container {
  width: 100%;
  min-height: 80vh;
  padding: 7px;
  background-color: aqua;
}
```

```
.primary-container>.section {
    margin: 30px 3px;
    padding: 15px;
    border-radius: 9px;
    display: flex;
    flex-direction: column;
    align-items: center;
}

.primary-container>.section-1 {
    background-color: white;
}

.rows {
    display: flex;
    flex-wrap: wrap;
    margin: 12px 16px;
    border-radius: 3px;
    padding: 9px 10px;
    background-color: #8ddfc8;
    justify-content: space-around;
    width: 100%;
}

.imagediv {
    height: 180px;
    padding: 10px;
    width: auto;
}

.imagediv>img {
    height: 100%;
    width: auto;
    border-radius: 3px;
    transition: all 0.7s ease;
}

img:hover {
    transform: scale(1.1);
}

.of1, .of2 {
    max-width: 40%;
    margin-top: 6px;
    justify-content: center;
}

.of1 {
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
}

.of2>div>p {
```

```
        text-align: left !important;
    }

.of1>div:nth-child(2) {
    padding: 40px;
}

.of1>div:nth-child(2), .of2>div:nth-child(2) {
    padding: 8px;
}

.primary-container>.section-2 {
    background-color: #f4ebeb;
}

.glow {
    color: #e8ffde;
}

.section-2>div {
    background-color: #0097a7;
    margin: 4px;
    color: white;
    font-weight: 600;
    padding: 12px 4px;
    border-radius: 4px;
    width: 100%;
}

.teachers>div>img {
    margin-left: -14px;
    width: 70px;
    height: auto;
    border-radius: 50%;
}

@media only screen and (max-width: 375px) {
    .teachers>div>img {
        width: 60px;
    }
}

@media only screen and (min-width: 768px) {
    .teachers>div>img {
        width: 100px;
    }
}

.students div>div>img {
    max-width: 300px;
    height: auto;
    padding: 9px;
    box-shadow: 0px 0px 5px black;;
    background-color: white;
```

```
}

.size-Custom {
    font-size: x-large;
}

.primary-container>.section-3 {
    background-color: white;
}

.primary-container>.section-4 {
    background-color: #f4ebeb;
}

.primary-container>.section-5 {
    background-color: white;
}

.primary-container>.section-6 {
    background-color: #f4ebeb;
}
```

**UJJWAL PANDEY**

**Login.css**

```
.main-data {
    height: 85vh;
    width: 100%;
    background-image: Linear-gradient(111deg, rgb(64, 151, 147),
        rgb(164, 70, 215), rgb(64, 151, 147));
    box-shadow: inset 0px 0px 10px -3px black;
}

.form-outer {
    min-height: 70%;
    height: auto;
    min-width: 300px;
    width: 60%;
    max-width: 70%;
    padding: 10px;
    background-image: Linear-gradient(252deg, rgb(218 172 243),
        rgb(113, 223, 218));
    box-shadow: inset 0px 0px 10px -3px black;
    /*border: 1px solid black;*/
    display: flex;
    flex-direction: column;
    justify-content: center;
```

```
        min-width: 300px;
    }

section {
    display: block;
    width: 100%;
    padding: 13px 35px;
}

section>label {
    max-width: 40%;
    font-weight: normal;
    font-size: x-large;
    /*margin-left: 20px;*/
    font-style: oblique;
}

section>input {
    width: 100%;
    border: none;
    outline: none;
    padding: 3px 13px;
    border-bottom: 1px solid blue;
}

section>input:focus {
    border-bottom: 2px solid blue
}

input[type="submit"] {
    background-color: #c9d0fc;
    width: 40%;
    border: 1px solid lime;
    box-shadow: 0 0 4px lime;
    padding: 4px 15px;
    font-size: 1.25rem;
    border-radius: 20px;
    margin-left: 50%;
    transform: translate(-50%, -50%);
    border: 1px solid lime;
    outline: none;
}

input[type="submit"]:hover {
    background-color: lime;
    color: black;
}

form>h2 {
    font-weight: 500;
    text-align: center;
}
```

+

### register.css

```
.main-data {  
    height: 81vh;  
    width: 100%;  
    box-shadow: inset 0px 0px 10px -3px black;  
    background: linear-gradient(to right, rgb(0, 159, 255), rgb(236, 47, 75));  
    background: linear-gradient(to right, rgb(43 164 236), #ee2b46cf);  
}  
  
.form-outer {  
    max-width: 70%;  
    width: 60%;  
    min-height: 70%;  
    min-width: 300px;  
    padding: 10px;  
    background-image: linear-gradient(252deg, rgb(218 172 243),  
        rgb(113, 223, 218));  
    box-shadow: inset 0px 0px 10px -3px black;  
    /*border: 1px solid black;*/  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
}  
  
.form-outer>h2 {  
    font-weight: 300;  
    text-align: center;  
}  
  
section {  
    display: block;  
    width: 100%;  
}  
  
section>label {  
    max-width: 40%;  
    font-weight: lighter;  
    margin-left: 20px;  
    font-style: oblique;  
}  
  
section>input {  
    width: 100%;  
    border: none;  
    outline: none;  
    padding: 3px 13px;  
    border-bottom: 1px solid blue;  
}
```

```
}

section>input:focus {
    border-bottom: 2px solid blue
}

input[type="submit"] {
    text-align: center;
    align-self: center;
    width: 40%;
    background-color: #c9d0fc;
    /*border: 1px solid lime;*/
    box-shadow: 0 0 4px Lime;
    padding: 4px 15px;
    font-size: 1.15rem;
    border-radius: 20px;
    margin-left: 50%;
    transform: translate(-50%, -50%);
}

input[type="submit"]:hover {
    background-color: Lime;
    color: black;
}

input[type="submit"]:focus {
    background-color: #00ff0075;
}

.hidden {
    display: none;
}

.mustFill {
    color: red;
}
```

## Codes of 'webapp/WEB-INF' folder

### Web.xml

```
<!DOCTYPE web-app PUBLIC
        "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
        "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
    <display-name>Archetype Created Web Application</display-name>
    <servlet>
        <servlet-name>RegisterServlet</servlet-name>
```

```
<display-name>RegisterServlet</display-name>
<description />
<servlet-class>servlets.RegisterServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>LoginServlet</servlet-name>
    <display-name>LoginServlet</display-name>
    <description />
    <servlet-class>servlets.LoginServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>LogoutServlet</servlet-name>
    <display-name>LogoutServlet</display-name>
    <description />
    <servlet-class>servlets.LogoutServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>QuerySubmitServlet</servlet-name>
    <display-name>QuerySubmitServlet</display-name>
    <description />
    <servlet-class>servlets.QuerySubmitServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>ViewUserServlet</servlet-name>
    <display-name>ViewUserServlet</display-name>
    <description />
    <servlet-class>servlets.ViewUserServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>AddCourse</servlet-name>
    <display-name>AddCourse</display-name>
    <description />
    <servlet-class>servlets.AddCourse</servlet-class>
</servlet>
<servlet>
    <servlet-name>AddCourseServlet</servlet-name>
    <display-name>AddCourseServlet</display-name>
    <description />
    <servlet-class>servlets.AddCourseServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>DeleteCourse</servlet-name>
    <display-name>DeleteCourse</display-name>
    <description />
    <servlet-class>servlets.DeleteCourse</servlet-class>
</servlet>
<servlet>
    <servlet-name>UpdateCourse</servlet-name>
    <display-name>UpdateCourse</display-name>
    <description />
    <servlet-class>servlets.UpdateCourse</servlet-class>
</servlet>
<servlet>
    <servlet-name>UpdateCourseServlet</servlet-name>
```

```
<display-name>UpdateCourseServlet</display-name>
<description />
<servlet-class>servlets.UpdateCourseServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>DeleteUser</servlet-name>
    <display-name>DeleteUser</display-name>
    <description />
    <servlet-class>servlets.DeleteUser</servlet-class>
</servlet>
<servlet>
    <servlet-name>DeleteUserServlet</servlet-name>
    <description />
    <servlet-class>servlets.DeleteUserServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>UpdateUserServlet</servlet-name>
    <servlet-class>servlets.UpdateUserServlet</servlet-class>
</servlet>
<servlet>
    <servlet-name>BuyCourseServlet</servlet-name>
    <servlet-class>servlets.BuyCourseServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>RegisterServlet</servlet-name>
    <url-pattern>/RegisterServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>LoginServlet</servlet-name>
    <url-pattern>/LoginServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>LogoutServlet</servlet-name>
    <url-pattern>/LogoutServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>QuerySubmitServlet</servlet-name>
    <url-pattern>/QuerySubmitServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>ViewUserServlet</servlet-name>
    <url-pattern>/ViewUserServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AddCourse</servlet-name>
    <url-pattern>/AddCourse</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AddCourseServlet</servlet-name>
    <url-pattern>/AddCourseServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>DeleteCourse</servlet-name>
    <url-pattern>/DeleteCourse</url-pattern>
```

```
</servlet-mapping>
<servlet-mapping>
    < servlet-name>UpdateCourse</servlet-name>
    < url-pattern>/UpdateCourse</url-pattern>
</servlet-mapping>
<servlet-mapping>
    < servlet-name>UpdateCourseServlet</servlet-name>
    < url-pattern>/UpdateCourseServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    < servlet-name>DeleteUser</servlet-name>
    < url-pattern>/DeleteUser</url-pattern>
</servlet-mapping>
<servlet-mapping>
    < servlet-name>DeleteUserServlet</servlet-name>
    < url-pattern>/DeleteUserServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    < servlet-name>BuyCourseServlet</servlet-name>
    < url-pattern>/BuyCourseServlet</url-pattern>
</servlet-mapping>
<servlet-mapping>
    < servlet-name>UpdateUserServlet</servlet-name>
    < url-pattern>/UpdateUserServlet</url-pattern>
</servlet-mapping>

</web-app>
```

## Codes of 'webapp' folder

### addNewUser.jsp

```
<%
UserDetails userDetailsService1 = (UserDetails) session.getAttribute("logedInUser");
if ((userDetailsService1 == null) || (!userDetailsService1.getUserType().equals("admin"))) {
    session.setAttribute("login-failed",
    "Only Admin can access this page. Login as admin");
    response.sendRedirect("LogoutServlet");
    //response.sendRedirect("login.jsp");
}
%><%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link rel="stylesheet" type="text/css" href="stylesheet/register.css">
<style type="text/css">
.mustFill {
    color: red;
}
```

```
</style>
<title>Admin add user | DevelopersPoint</title>
</head>
<body>
    <%@include file="common/navbar.jsp"%>

    <div class="container">
        <!-- Form begins here -->
        <form method="post" action="RegisterServlet">
            <h2 class="text-center">Add new user with respective details</h2>
            <section>
                <label>First Name:<span class="mustFill">*</span></label> <br>
                <input type="text" placeholder="First name" required
                    name="uFirstName">
            </section>
            <br>
            <section>
                <label>Last Name:</label> <br> <input type="text"
                    placeholder="Last name" name="uLastName">
            </section>
            <br>
            <section>
                <label>E-mail:</label><span class="mustFill">*</span> <br> <input
                    type="email" placeholder="E-mail Address" required
                    name="uEmail">
            </section>
            <br>
            <section>
                <label>Password:</label><span class="mustFill">*</span> <br> <input
                    type="password" min="8" max="16" required
                    placeholder="8-16 character long" name="uPassword"
                    onkeyup="CheckCount()"> <label class="hidden"
                    id="alert_custom" style="color: orange;">The password must
                    be between 8-16 character long!!</label>
            </section>
            <br>
            <section>
                <label>Type of user:-</label><span class="mustFill">*</span> <br>
                <select class="form-select" name="uType"
                    aria-label="Default select example">
                    <option selected>Select user type</option>
                    <option value="student">Student</option>
                    <option value="faculty">Faculty</option>
                    <option value="admin">Admin</option>
                </select>
            </section>
            <br> <br> <input type="submit"
                class="btn btn-pill text-center vw-20" value="Submit" id="submitMe">
        </form>
    </div>
    <%@include file="common/footer.jsp"%>
</body>
</html>
```

## admin.jsp

```
<%>
UserDetails userDetailsService1 = (UserDetails) session.getAttribute("loggedInUser");
if ((userDetailsService1 == null) || (!userDetailsService1.getUserType().equals("admin"))) {
    session.setAttribute("login-failed",
    "Only Admin can access this page. Login as admin");
    response.sendRedirect("LogoutServlet");
    //response.sendRedirect("login.jsp");
}
%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<style type="text/css">
.courses, .user {
    background-color: #bfd4ec;
    margin: 10px;
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
    border-radius: 5px;
    text-align: center;
}
.customFlexCard {
    min-width: 200px;
    text-align: center;
    margin: 4px 7px;
}
.card {
    padding: 0px;
}
</style>
<title>Admin | DevellpersPoint</title>
</head>
<body style="background-color: rgb(186, 226, 238);">
    <%@include file="common/navbar.jsp"%>
    <hr>

    <h2 class="display-3 text-center m-1">Courses</h2>
    <hr>
    <%
    String courseAddedSuccessful = (String) session.getAttribute("course-added");
    if (courseAddedSuccessful != null) {
```

```
%>
<div class="alert alert-success text-center fs-3 mx-2"
role="alert"><%=courseAddedSuccess%></div>
<%
session.removeAttribute("course-added");
%
%>
<div class="courses">
    <div class="p-2 customFlexCard">
        <div class="card border-secondary mb-3 col" style="max-width: 18rem;">
            <div class="card-header">
                <b>View List of All courses</b>
            </div>
            <div class="card-body text-secondary">
                <h5 class="card-title">A list of all courses available</h5>
                <p class="card-text">
                    Will show a list of courses with details and 'Two
buttons' <br>&bull;
                    One to do editing to the course <br> &bull; Another to
                    delete
                    the course
                </p>
                <a class="btn btn-info" href="adminViewCourses.jsp">View all
                    courses</a>
            </div>
        </div>
    </div>
    <div class="p-2 customFlexCard">
        <div class="card border-secondary mb-3 col" style="max-width: 18rem;">
            <div class="card-header">
                <b>Add new course</b>
            </div>
            <div class="card-body text-secondary">
                <h5 class="card-title">Add more courses</h5>
                <p class="card-text">
                    Contains a form with input fields:- <br> &bull; Course
                    title
                    <br> &bull; Price of course<br> &bull; Description<br>
                    &bull; For which subject it is for<br>
                </p>
                <a class="btn btn-info" href="adminAddCourse.jsp">Add new
                    course</a>
            </div>
        </div>
    </div>
<hr>

<h2 class="display-3 text-center m-1">User</h2>
<hr>
<%
String added = (String) session.getAttribute("reg-sucess");
if (added != null) {
```

```
%>
<div class="alert alert-success text-center fs-2" role="alert"><%=added%></div>
<%
}
session.removeAttribute("reg-sucess");
%>
<div class="user">
    <div class="p-2 customFlexCard">
        <div class="card border-secondary mb-3 col" style="max-width: 18rem;">
            <div class="card-header">
                <b>All users</b>
            </div>
            <div class="card-body text-secondary">
                <h5 class="card-title">List of current user</h5>
                <p class="card-text">
                    Will show a list of users with details with 'Two
                    buttons' <br>&bull;
                    One to do editing of that user <br> &bull; Another to
                    delete
                    the course
                </p>
                <a class="btn btn-info"
                    href="adminViewUsers.jsp?whoUser=all">View
                    all users</a>
            </div>
        </div>
        <div class="p-2 customFlexCard">
            <div class="card border-secondary mb-3 col" style="max-width: 18rem;">
                <div class="card-header">
                    <b>All faculty</b>
                </div>
                <div class="card-body text-secondary">
                    <h5 class="card-title">List of Faculties</h5>
                    <p class="card-text">
                        Shows a list of faculty in details with 'Two buttons'
                        <br>&bull;
                        Edit that user <br>&bull; Delete the user
                    </p>
                    <a class="btn btn-info"
                        href="adminViewUsers.jsp?whoUser=faculty">View
                        all faculty</a>
                </div>
            </div>
            <div class="p-2 customFlexCard">
                <div class="card border-secondary mb-3 col" style="max-width: 18rem;">
                    <div class="card-header">
                        <b>All student</b>
                    </div>
                    <div class="card-body text-secondary">
                        <h5 class="card-title">List of Students</h5>
                        <p class="card-text">
```

```
<br>&bull; Shows a list of students in details with 'Two buttons'  
<br>&bull; Edit that user <br>&bull; Delete the user  
</p>  
<a class="btn btn-info" href="adminViewUsers.jsp?whoUser=student">View  
all student</a>  
</div>  
</div>  
</div>  
<div class="p-2 customFlexCard">  
<div class="card border-secondary mb-3 col" style="max-width: 18rem;">  
<div class="card-header">  
<b>Add User</b>  
</div>  
<div class="card-body text-secondary">  
<h5 class="card-title">Add a User</h5>  
<p class="card-text">  
To add a new user fill: <br>&bull; First and Last Name  
of  
user <br>&bull; Email, Password of user <br>&bull; Type  
of user  
</p>  
<a class="btn btn-info" href="addNewUser.jsp">Add new user</a>  
</div>  
</div>  
</div>  
<hr>  
  
<%@include file="common/footer.jsp"%>  
</body>  
</html>
```

### adminAddCourse.jsp

```
<%  
UserDetails userDetails1 = (UserDetails) session.getAttribute("loggedInUser");  
if ((userDetails1 == null) || (!userDetails1.getUserType().equals("admin"))){  
    session.setAttribute("login-failed",  
    "Only Admin can access this page. Login as admin");  
    response.sendRedirect("LogoutServlet");  
    //response.sendRedirect("login.jsp");  
}  
%>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link rel="stylesheet" type="text/css" href="stylesheet/register.css">
<style type="text/css">
.mustFill {
    color: red;
}
</style>
<title>Admin Add course | DevelopersPoint</title>
</head>
<body>
    <%@include file="common/navbar.jsp"%>

    <!-- Form to add new Course -->
    <div class="container">
        <!-- Form begins here -->
        <form method="post" action="AddCourseServlet">
            <h2 class="text-center">Add new Course with respective details</h2>
            <section>
                <label>Course Title:<span class="mustFill">*</span></label> <br>
                <input type="text" placeholder="Title to show on course" required
                       name="cTitle">
            </section>
            <br>
            <section>
                <label>Course Description:</label> <span
class="mustFill">*</span><br>
                <input type="text"
                       placeholder="Little explanation of Learning material" required
                       name="cExplanation">
            </section>
            <br>
            <section>
                <label>Price (in rs):</label><span class="mustFill">*</span> <br>
                <input type="number" placeholder="Course price without any discount"
                       required name="cPrice">
            </section>
            <br>
            <section>
                <label>Subject it is related to:</label><span
class="mustFill">*</span>
                <br> <select class="form-select" name="subjectImage" required
                           aria-label="Default select example">
                    <option selected>Select Course Subject</option>
                    <option value="java.png">Java</option>
                    <option value="mongoDB.png">MongoDB</option>
                    <option value="python.jpg">Python</option>
                    <option value="sql.jpg">SQL</option>
                    <option value="advancedjava.jpg">Advanced Java</option>
                    <option value="mysql.jpg">MySQL</option>
                    <option value="html.jpg">HTML</option>
                    <option value="css.jpg">CSS</option>
                </select>
            </section>
        </form>
    </div>
</body>
```

```
        <option value="javascript.png">JavaScript</option>
    </select>
</section>
<br> <br> <input type="submit"
    class="btn btn-pill text-center vw-20" value="Submit" id="submitMe">
</form>
</div>

<%@include file="common/footer.jsp"%>
</body>
</html>
```

### adminViewCourses.jsp

```
<%@page import="org.hibernate.internal.build.AllowSysOut"%>
<%@page import="org.hibernate.Query"%>
<%@page import="java.util.Iterator"%>
<%@page import="org.hibernate.Criteria"%>
<%@page import="java.util.List"%>
<%@page import="entities.CourseList"%>
<%@page import="org.hibernate.Session"%>
<%@page import="connection.DBConnection"%>
<%
UserDetails userDetailsService1 = (UserDetails) session.getAttribute("loggedInUser");
if ((userDetailsService1 == null) || (!userDetailsService1.getUserType().equals("admin"))) {
    session.setAttribute("login-failed",
        "Only Admins can access this page.\n Login as admin");
    response.sendRedirect("LogoutServlet");
    //response.sendRedirect("login.jsp");
}
%><%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Courses | DevelopersPoint</title>
</head>
<body>
    <%@include file="common/navbar.jsp"%>
    <%
Session sess = DBConnection.getFactory().openSession();
sess.beginTransaction();
Criteria criteria = sess.createCriteria(CourseList.class);
List<CourseList> col = criteria.list();
%>
```

```
<div class="container-fluid my-4 overflow-auto">
    <table class="table table-hover">
        <thead class="thead-light">
            <h2 class="display-4 text-center">Course</h2>
            <%
                String st = (String) session.getAttribute("course-delete-success");
                if (st != null) {
            %>
            <div class="alert alert-info text-center fade show"
                style="cursor: grab;">
                <%=st%>
                <span class="closebtn text-right float-right"
                    style="right: 10px; cursor: pointer; background-color:
white; padding: 2px 8px; border-radius: 3px; margin-left: 30px;">
                    onclick="this.parentElement.style.display='none';">x</span>
            </div>
            <%
                session.removeAttribute("course-delete-success");
            %}
            <tr>
                <th scope="col">Id</th>
                <th scope="col">Title</th>
                <th scope="col">Description</th>
                <th scope="col">Added on</th>
                <th scope="col">Price</th>
                <th scope="col">Edit</th>
                <th scope="col">Delete</th>
            </tr>
        </thead>
        <tbody>
            <%
                Iterator<CourseList> itr = col.iterator();
                while (itr.hasNext()) {
                    CourseList cl = itr.next();
            %>
            <tr>
                <th scope="row"><%=cl.getCourseID()%></th>
                <td style="word-break: break-
all;"><%=cl.getCourseName()%></td>
                <td style="word-break: break-
all;"><%=cl.getCourseDescription()%></td>
                <td><%=cl.getCourseAddedDate()%></td>
                <td><%=cl.getCoursePrice()%></td>
                <td><a type="button" class="btn btn-primary"
                    href="updateCourse.jsp?courseID=<%=cl.getCourseID()%>">Update</a></td>
                    <td><a type="button" class="btn btn-danger"
                        href="DeleteCourse?courseID=<%=cl.getCourseID()%>">Delete</a></td>
                </tr>
            <%
            }
        
```

```
%>
    </tbody>
  </table>
</div>
<!-- Closing the page data loading--&gt;
&lt;%
sess.getTransaction().commit();
sess.close();
%&gt;
&lt;%@include file="common/footer.jsp"%&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

### adminViewUsers.jsp

```
<%@page import="org.hibernate.internal.build.AllowSysOut"%>
<%@page import="java.util.List"%>
<%@page import="org.hibernate.Query"%>
<%@page import="entities.CourseList"%>
<%@page import="connection.DBConnection"%>
<%@page import="org.hibernate.Session"%>
<%
UserDetails userDetails1 = (UserDetails) session.getAttribute("logedInUser");
if ((userDetails1 == null) || (userDetails1.getUserType().equals("student"))) {
    session.setAttribute("login-failed",
    "Only Admins can access this page.\n Login as admin");
    response.sendRedirect("LogoutServlet");
}
%><%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link rel="stylesheet" type="text/css" href="stylesheet/register.css">
<title>Admin View users | DevelopersPoint</title>
</head>
<body>
<%@include file="common/navbar.jsp"%>
<%
String userType = request.getParameter("whoUser");
//System.out.println(userType);
Session sess = DBConnection.getFactory().openSession();
sess.beginTransaction();
String hql = null;
if (userType.equals("all")) {
    hql = "From UserDetails";
```

```
        } else if (userType.equals("faculty")) {
            hql = "From UserDetails where userType = 'faculty'";
        } else if (userType.equals("student")) {
            hql = "From UserDetails where userType = 'student'";
        }
    Query<UserDetails> query = sess.createQuery(hql);
    List<UserDetails> userList = query.list();
%>
<!-- Table -->
<div class="container-fluid my-4 overflow-auto">
    <%
        String succ = (String) session.getAttribute("Users-Success");
        if (succ != null) {
    %>
        <div class="alert alert-info text-center fade show"
            style="cursor: grab;">
            <%=succ%>
            <span class="closebtn text-right float-right"
                style="right: 10px; cursor: pointer; background-color: white;
padding: 2px 8px; border-radius: 3px; margin-left: 30px;">
                onclick="this.parentElement.style.display='none';">x</span>
        </div>
    <%
        session.removeAttribute("Users-Success");
    }
    %>
    <table class="table table-hover">
        <thead class="thead-light">
            <h2 class="display-4 text-center">Users</h2>
            <%
                String st = (String) session.getAttribute("user-change-success");
                if (st != null) {
            %>
                <div class="alert alert-info text-center fade show"
                    style="cursor: grab;">
                    <%=st%>
                    <span class="closebtn text-right float-right"
                        style="right: 10px; cursor: pointer; background-color: white;
padding: 2px 8px; border-radius: 3px; margin-left: 30px;">
                        onclick="this.parentElement.style.display='none';">x</span>
                </div>
            <%
                session.removeAttribute("course-delete-success");
            }
            %>
            <tr>
                <th scope="col">Name</th>
                <th scope="col">Email</th>
                <th scope="col">User Type</th>
                <th scope="col">Contact</th>
                <th scope="col">Address</th>
                <th scope="col">With us from</th>
            <%
```

```
        if (userDetails.getUserType().equals("admin")) {  
            %>  
            <th scope="col">Edit</th>  
            <th scope="col">Delete</th>  
  
            <%>  
            }  
            %>  
        </tr>  
    </thead>  
    <tbody>  
        <%>  
        for (UserDetails element : userList) {  
            %>  
            <tr>  
                <th scope="row"><%=element.getFullName()%></th>  
                <td style="word-break: break-all;"><%=element.getEmail()%></td>  
                <td style="word-break: break-all;"><%=element.getUserType()%></td>  
                <td><%=element.getContactNumber()%></td>  
                <td><%=element.toStringFullAddress()%></td>  
                <td><%=element.getRegistrationDate()%></td>  
  
                <%>  
                if (userDetails.getUserType().equals("admin")) {  
                    %>  
                    <td><a type="button" class="btn btn-primary"  
                        href="updateUser.jsp?userId=<%=element.getEmail()%>">Update</a></td>  
                    <td><a type="button" class="btn btn-danger"  
                        href="DeleteUserServlet?courseID=<%=element.getEmail()%>">Delete</a></td>  
                }  
                %>  
            </tr>  
        }  
    </tbody>  
    </table>  
    </div>  
    <%>  
    sess.getTransaction().commit();  
    sess.close();  
    %>  
    <%@include file="common/footer.jsp"%>  
</body>  
</html>
```

### contact.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Contact | Developers Point</title>
<style type="text/css">
.mustFill {
    color: red;
}
</style>
</head>
<body>

<%@include file="common/navbar.jsp"%>

<div class="jumbotron jumbotron-fluid">
    <div class="container my-3">

        <%
        String querySubmitted = (String) session.getAttribute("querySubmitted");
        if (querySubmitted != null) {
        %>
            <div class="alert alert-success" role="alert"><%=querySubmitted%></div>
        <%
        session.removeAttribute("querySubmitted");
        }
        %>

        <h2>Fill form no login required</h2>
        <form action="QuerySubmitServlet" method="post">
            <section class="my-3">
                <label for="name"> Name<span class="mustFill">*</span></label>
                <input type="text" class="form-control" name="name"
                    placeholder="Your name" required>
            </section>
            <section class="my-3">
                <label for="email"> Email-id<span
                    class="mustFill">*</span></label>
                <input type="email" class="form-control" name="email"
                    placeholder="Email id to contact you" required>
            </section>
            <section class="my-3">
```

```
<label for="contact"> Reachable number</label> <input  
type="text"  
number">  
</section>  
<section class="my-3">  
    <label for="contact"> Type of query</label><input type="text"  
class="form-control" name="queryHead"  
placeholder="A heading for your query">  
</section>  
<section class="my-3">  
    <label for="contact"> Query<span  
class="mustFill">*</span><sub>in  
details</sub>  
</label>  
    <textarea class="form-control" name="queryExplained" rows="7"  
placeholder="Explain properly in details we will reach  
as soon as possible to solve">  
        required</textarea>  
</section>  
    <br> <input type="submit" value="Submit">  
</form>  
<hr>  
<div class="text-center">  
    <h2>  
        <strong>Some other ways to contact us</strong>  
    </h2>  
    <b>Number:</b> <i>+91-9876-5432-89, +91-8375-9905-00</i> <br>  
    <i>support@developerspoint.com</i> <br> <b>Visit:</b> <i>36,  
    Ground floor, near bank of baroda, Sankat Vihar, New Delhi,  
Delhi  
110067</i>  
</div>  
</div>  
<%@include file="common/footer.jsp"%>  
</body>  
</html>
```

### coursesBought.jsp

```
<%@page import="entities.CourseList"%>  
<%@page import="org.hibernate.Criteria"%>  
<%@page import="connection.DBConnection"%>  
<%@page import="org.hibernate.Session"%>  
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"  
pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<style type="text/css">
.bg-cus {
    background-color: #f5eaf7;
}
</style>
<title>Course detail | DevelopersPoint</title>
</head>
<body>
<%@include file="common/navbar.jsp"%>
<%
int id = Integer.parseInt(request.getParameter("courseId"));
Session sess = DBConnection.getFactory().openSession();
sess.beginTransaction();
CourseList specific = (CourseList) sess.get(CourseList.class, id);
%>
<div class="row" style="min-height: 78vh; width: 100vw;">
    <div class="col-xs-12 p-5 col-md-6">
        <br> <br> <span class="fs-3"><strong>Topic:</strong><%=specific.getCourseName()%></span>
        <div>
            <strong>Launch Date:</strong>
<strong><%=specific.getCourseAddedDate()%></strong>
        </div>
        <div class="col-xs-12 p-5 col-md-6 bg-cus">
            <div class="shadow p-3 mb-5 bg-body rounded fs-4">
                <span class="text-center fw-bolder">Description: </span> In this
                course we
                <%=specific.getCourseDescription()%>
            <div class="shadow p-3 mb-5 bg-body rounded fs-3">
                <strong>Rs: </strong><%=specific.getCoursePrice()%>
            <div class="shadow p-3 mb-5 bg-body rounded fs-3">
                <a class="btn btn-info px-3 w-100" role="button"
                href="contact.jsp">Having
                <div style="display: flex; justify-content: space-between; align-items: center;">
                    <div style="flex-grow: 1; text-align: center;">
                        <strong>Problem? contact us</strong>
                    </div>
                    <img alt="Email icon" style="width: 2em; height: 2em; margin-left: 10px;">
                </div>
            </div>
        <%
        sess.getTransaction().commit();
        sess.close();
        %>
        <script type="text/javascript">
            function batchCourseEnquiry() {
                alert("Feel free to ask anything, and sit back freely we will reach you in
25min");
            }
            //onclick="batchCourseEnquiry()"
        </script>
    </div>
</div>
</body>
```

```
<%@include file="common/footer.jsp"%>
</body>
</html>
```

### courseDetail.jsp

```
<%@page import="entities.CourseList"%>
<%@page import="org.hibernate.Criteria"%>
<%@page import="connection.DBConnection"%>
<%@page import="org.hibernate.Session"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<style type="text/css">
.bg-cus {
    background-color: #f5eaf7;
}
</style>
<title>Course detail | DevelopersPoint</title>
</head>
<body>
<%@include file="common/navbar.jsp"%>
<%
int id = Integer.parseInt(request.getParameter("courseId"));
Session sess = DBConnection.getFactory().openSession();
sess.beginTransaction();
CourseList specific = (CourseList) sess.get(CourseList.class, id);
%>
<div class="row" style="min-height: 78vh; width: 100vw;">
    <div class="col-xs-12 p-5 col-md-6">
        <br> <br> <span class="fs-3"><strong>Topic:</strong><%=specific.getCourseName()%></span>
        <div>
            <strong>Launch Date:</strong>
<%=specific.getCourseAddedDate()%></div>
        <div class="col-xs-12 p-5 col-md-6 bg-cus">
            <div class="shadow p-3 mb-5 bg-body rounded fs-4">
                <span class="text-center fw-bolder">Description: </span> In this
                course we
                <%=specific.getCourseDescription()%></div>
            <div class="shadow p-3 mb-5 bg-body rounded fs-3">
                <strong>Rs: </strong><%=specific.getCoursePrice()%></div>

```

```
<div class="shadow p-3 mb-5 bg-body rounded fs-3">
    <a class="btn btn-info px-3 w-100" role="button"
        onclick="batchCourseEnquiry()" href="contact.jsp">More
Detail</a> <br>
<%
    if (userDetails != null) {
%
        <a class="btn btn-info px-3 w-100" role="button"
            href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>">Buy
        Course</a>
%
        session.setAttribute("courseDetails", specific);
    } else {
%
        <a class="btn btn-info px-3 w-100" role="button"
            href="Login.jsp">Login
            to buy Course</a>
%
    }
%
</div>
</div>
<%
sess.getTransaction().commit();
sess.close();
%>
<script type="text/javascript">
    function batchCourseEnquiry() {
        alert("Feel free to ask anything, and sit back freely we will reach you in
25min");
    }
    //onclick=batchCourseEnquiry()
</script>
<%@include file="common/footer.jsp"%>
</body>
</html>
```

courses.jsp

```
<%@page import="org.hibernate.internal.build.AllowSysOut"%>
<%@page import="org.hibernate.Query"%>
<%@page import="java.util.Iterator"%>
<%@page import="org.hibernate.Criteria"%>
<%@page import="java.util.List"%>
```

```
<%@page import="entities.CourseList"%>
<%@page import="org.hibernate.Session"%>
<%@page import="connection.DBConnection"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>

<html>
<head>
<meta charset="ISO-8859-1">
<title>Courses | DevelopersPoint</title>
<style>
* {
    box-sizing: border-box;
}

.bg-cus {
    background-color: #ebd3ed;
}

.x {
    /* margin: 3px; */
    margin: 13px 0px;
    padding: 10px;
    /*border: 7px solid white;*/
}
</style>
</head>
<body class="bg-cus">
<%@include file="common/navbar.jsp"%>
<%
Session sess = DBConnection.getFactory().openSession();
sess.beginTransaction();
Criteria criteria = sess.createCriteria(CourseList.class);
List<CourseList> col = criteria.list();
%>
<div class="container-fluid my-4 overflow-auto">
    <h2 class="display-4 text-center">Course</h2>
    <%
        String courseBoughtSuccessful = (String) session.getAttribute("CourseBought");
        if (courseBoughtSuccessful != null) {
    %>
        <div class="alert alert-success text-center fs-3 mx-2"
role="alert"><%=courseBoughtSuccessful%></div>
    <%
        session.removeAttribute("CourseBought");
    %>
    <!-- Create cards -->
    <div class="container-fluid">
        <div class="row text-center justify-content-center bg-cus">
            <%
                Iterator<CourseList> itr = col.iterator();
                while (itr.hasNext()) {
```

```
CourseList cl = itr.next();
%>
<div class="col-xs-12 col-sm-6 col-md-4 col-xl-3 x bg-cus">
    <div class="card mx-auto" style="width: 18rem;">
        
        <div class="card-body">
            <h5 class="card-title" style="word-break: break-all;"><%=cl.getCourseName()%></h5>
            <hr>
            <p class="card-text" style="word-break: break-all;"><%=cl.getCourseDescription()%></p>
            <a href="courseDetail.jsp?courseId=<%=cl.getCourseID()%>">
                class="btn btn-primary">More Detail</a>
        </div>
    </div>
<%
}
%>
</div>
</div>
<%
sess.getTransaction().commit();
sess.close();
%>
<%@include file="common/footer.jsp"%>
</body>
</html>
```

### faculty.jsp

```
<%@page import="java.util.Iterator"%>
<%@page import="java.util.List"%>
<%@page import="org.hibernate.Criteria"%>
<%@page import="entities.BoughtCourses"%>
<%@page import="connection.DBConnection"%>
```

```
<%@page import="org.hibernate.Session"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<style type="text/css">
/* Hide scrollbar for Chrome, Safari and Opera */
.example::-webkit-scrollbar {
    display: none;
}

/* Hide scrollbar for IE, Edge and Firefox */
.example {
    -ms-overflow-style: none; /* IE and Edge */
    scrollbar-width: none; /* Firefox */
    box-shadow: inset 0px -3.5PX 7px rgb(0 0 0/ 15%);
}

.courses, .user {
    background-color: #bfd4ec;
    margin: 10px;
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
    border-radius: 5px;
    text-align: center;
}

.customFlexCard {
    min-width: 200px;
    text-align: center;
    margin: 4px 7px;
}

.card {
    padding: 0px;
}
</style>
</style>
<title>Faculty | DevelopersPoint</title>
</head>
<body>
<%@include file="common/navbar.jsp"%>

<%
Session sess = DBConnection.getFactory().openSession();
sess.beginTransaction();
Criteria criteria = sess.createCriteria(BoughtCourses.class);
List<BoughtCourses> col = criteria.list();
%>
<div class="row" style="height: 78vh; width: 100vw;">
    <div class="col-xs-12 col-md-8" style="padding-left: 22px;">
```

```
<div class="example my-1" style="height: 38vh; overflow-y: scroll;">
    <div class="user">
        <div class="p-2 customFlexCard">
            <div class="card border-secondary mb-3 col"
                style="max-width: 18rem;">
                <div class="card-header">
                    <b>All users</b>
                </div>
                <div class="card-body text-secondary">
                    <h5 class="card-title">List of current
user</h5>

details with 'Two buttons' &bull;
&bull; Another to
View all users</p>
<a class="btn btn-info" href="adminViewUsers.jsp?whoUser=all">Edit that user &bull; Delete the
user</a>
</div>
</div>
<div class="p-2 customFlexCard">
    <div class="card border-secondary mb-3 col"
        style="max-width: 18rem;">
        <div class="card-header">
            <b>All faculty</b>
        </div>
        <div class="card-body text-secondary">
            <h5 class="card-title">List of
Faculties</h5>

with 'Two buttons' &bull;
user
View all faculty</p>
<a class="btn btn-info" href="adminViewUsers.jsp?whoUser=faculty">Edit that user &bull; Delete the
user</a>
</div>
</div>
<div class="p-2 customFlexCard">
    <div class="card border-secondary mb-3 col"
        style="max-width: 18rem;">
        <div class="card-header">
            <b>All student</b>
        </div>
        <div class="card-body text-secondary">
            <h5 class="card-title">List of
Students</h5>
```

```
<p class="card-text">
    Shows a list of students in details
with 'Two buttons' &br;&bull;
user
</p>
<a class="btn btn-info"
    href="adminViewUsers.jsp?whoUser=student">View all student</a>
</div>
</div>
<div class="container-fluid">
    <div style="height: 100%;"></div>
    <div class="row text-center justify-content-center">
        <%
            int count = 0;
            Iterator<BoughtCourses> itr = col.iterator();
            while (itr.hasNext()) {
                BoughtCourses cl = itr.next();
                if (cl.getUserEmail().equals(userDetails.getEmail())) {
                    count++;
                }
            }
        <%>
        <div class="col-xs-12 col-sm-6 x mt-4">
            <div class="card mx-auto" style="width: 18rem;">
                
                    alt="Card image cap">
                <div class="card-body">
                    <h5 class="card-title" style="word-break:
break-all;"><%=cl.getCourseName()%></h5>
                    <hr>
                    <p class="card-text" style="word-break:
break-all;">
                        <%=cl.getCouseDescription()%>
                    </p>
                    <a href="courseBought.jsp?courseId=<%=cl.getCourseID()%>">
                        Detail</a>
                        </div>
                    </div>
                <%
            }
        <%
        if (count < 1) {
    %>
    <div class="shadow p-3 mb-5 bg-body rounded fs-3 mt-5">
        <strong>No course bought yet</strong>
    </div>
</div>
</div>
```

```
<div class="shadow p-3 mb-5 bg-body rounded fs-3 mt-3">
    <a class="btn btn-info px-3 w-100" role="button"
        href="courses.jsp">Search Courses</a>
</div>
<%>
}
%>
</div>
</div>
<div class="col-xs-12 p-5 col-md-4 bg-cus">
    <h2 class="text-center">
        Profile details
    <hr>
    <div class="text-muted fs-5 lead">
        <div>
            Name:
            <%=userDetails.getFullName()%></div>
        <div>
            <br> Email id:
            <%=userDetails.getEmail()%></div>
        <br>
        <div>
            Contact number:
            <%=userDetails.getContactNumber()%></div>
        <div>
            <br> User Type:
            <%=userDetails.getUserType()%></div>
        <div>
            <br> User since:
            <%=userDetails.getRegistrationDate()%></div>
        <div>
            <br>
            <%=userDetails.toStringFullAddress()%></p>
            <a class="btn btn-info px-3 w-100" role="button"
                href="updateUser.jsp?userId=<%=userDetails.getEmail()%>">Edit</a>
        </div>
    </div>
</div>
<%>
sess.getTransaction().commit();
sess.close();
%>

<%@include file="common/footer.jsp"%>
</body>
</html>
```

### home.jsp

```
<%@page import="entities.UserDetails"%>
<%
UserDetails userDetails = (UserDetails) session.getAttribute("logedInUser");
if (userDetails == null) {
    session.setAttribute("Login-error",
    "Please login first to get at home page..");
    response.sendRedirect("login.jsp");
} else {
    HttpSession logedInDetails = request.getSession();
    logedInDetails.setAttribute("logedInUser", userDetails);
    if (userDetails.getUserType().equals("student")) {
        response.sendRedirect("student.jsp");
    } else if (userDetails.getUserType().equals("faculty"))
        response.sendRedirect("faculty.jsp");
    else if (userDetails.getUserType().equals("admin"))
        response.sendRedirect("admin.jsp");
}
%>
```

### index.jsp

```
<%@page import="connection.DBConnection"%>
<%@page import="org.hibernate.Session"%>
<%@page import="java.util.Iterator"%>
<%@page import="org.hibernate.Query"%>
<%@page import="entities.CourseList"%>
<%@page import="java.util.List"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link rel="stylesheet" type="text/css" href="stylesheet/index.css">
<title>Developers' Point | IGNOU PROJECT</title>
</head>
<body>
    <%@include file="common/navbar.jsp"%>
    <div class="jumbotron m-3 fs-4">
        <div class="primary-container">
            <div class="section section-1 text-center">
                <h1>
```

```
Our Latest Course
```

```
<% Session sess = DBConnection.getFactory().openSession();  
sess.beginTransaction();  
Query query = sess.createQuery("from CourseList ORDER BY  
courseAddedDate desc");  
  
to 10th number  
query.setFirstResult(0);  
query.setMaxResults(3);  
List<CourseList> col1 = query.list(); //will return the records from 5  
to 10th number  
Iterator<CourseList> itr1 = col1.iterator();  
CourseList cl = itr1.next();  
%>  
<hr>  
<div class="rows">  
    <div class="of2">  
        <div class="imagediv">  
              
        </div>  
        <div>  
            <p><b>Title:</b> <%=cl.getCourseName()%><br>  
            <%=cl.getCourseDescription()%><br> <a class="btn btn-info px-3 float-right" href="courseDetail.jsp?courseId=<%=cl.getCourseID()%>">More Detail</a>  
            </p>  
        </div>  
    </div>  
    <% cl = itr1.next(); %>  
    <div class="of2">  
        <div class="imagediv">  
              
        </div>  
        <div>  
            <p><b>Title:</b> <%=cl.getCourseName()%><br>  
            <%=cl.getCourseDescription()%><br> <a class="btn btn-info px-3 float-right" href="courseDetail.jsp?courseId=<%=cl.getCourseID()%>">More Detail</a>  
            </p>  
        </div>  
    </div>  
<div class="rows">  
    <div class="of2">  
        <div class="imagediv">  
              
        </div>  
        <div>  
            <p><b>Title:</b> <%=cl.getCourseName()%><br>  
            <%=cl.getCourseDescription()%><br> <a class="btn btn-info px-3 float-right" href="courseDetail.jsp?courseId=<%=cl.getCourseID()%>">More Detail</a>  
            </p>  
        </div>  
    </div>  
</div>
```

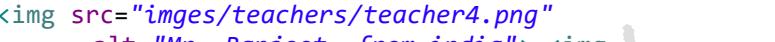


take your step and bend  
suitability</sub><br> <a  
right" href="contact.jsp"  
role="button">More Detail</a>

flexible before starting the batch,  
them according to your  
class="btn btn-info px-3 float-  
onclick="batchCourseEnquiry()">  
</p>  
</div>  
</div>  
</div>  
</div>  
<div class="section section-2 text-center w-100">  
    <h1 style="color: #fb8c00; margin: 20px auto; font-family: 'Belleza'; font-weight: bold;">Our  
        Exemplar</h1>  
    <div class="students">  
        <h2 class="glow">Some of our brilliant students</h2>  
        <div class="d-flex justify-content-center flex-wrap">  
            <div class="col w-30 p-2" style="min-width: 170px">  
                <img width="100%" class="pb-2" alt=""> <br>  
                <h4>Alina Gomes</h4>  
                <strong>Our Silver-ranked</strong> <br> Best AI  
                2107 with salary of 35 L.P.A.  
            </div>  
            <div class="col w-30 p-2" style="min-width: 170px">  
                <img width="100%" class="pb-2" alt=""> <br>  
                <h4>UJJWAL PANDEY</h4>  
                <strong>Our Golden student</strong> <br> Got  
                google with salary of 40 L.P.A.  
            </div>  
            <div class="col w-30 p-2" style="min-width: 170px">  
                <img width="100%" class="pb-2" alt=""> <br>  
                <h4>Madara Uchia</h4>  
                <strong>Our Bronze student</strong> <br> Best  
                of year 2019 with salary of 30 L.P.A.  
            </div>  
        </div>  
    </div>  
<div class="section teachers text-center">  
    <div class="mx-auto">

```
from india">





from usa">
Our world wide best & best faculty
```

Customer plus "

Customer circle "

```
</div>
<div class="section bg-white">
<div>
<div class="d-flex">
<a href="register.jsp" class="btn btn-custom btn-outline-dark mx-1 size-1" style="font-size: 2.1rem;"><i class="fa fa-user-plus" style="font-size: 2.1rem;"> Register</i> </a> <a href="Login.jsp" class="btn btn-custom btn-outline-dark mx-1 size-1" style="font-size: 2.1rem;"><i class="fa fa-user" style="font-size: 2.1rem;"> login</i> </a>
```

excellence to the faculty to teach them

**bold">Address**

of baroda, <br> 110067

```
</div>
<div class="section section-3">
<footer class="page-footer font-small blue pt-4">
<div class="container-fluid text-center text-md-left">
<div class="row">
<div class="col-md-5 mt-md-0 mt-3">
<h5 class="text-uppercase">Providing future</h5>
<p>We are those who provide the best & best student ensuring the best future for
```

</div>
<div class="col-md-3 mb-md-0 mb-3">
<h5 class="text-uppercase">

<p> 36,<br> Ground floor, <br> near bank Sankat Vihar,<br> Dew Delhi, Delhi

```
</div>
<div class="col-md-3 mb-md-0 mb-3">
    <h5 class="text-uppercase">Contact us</h5>
    <p>View Email:<br/>
        contactdp@Developerpoint.com<br/> <a href="mailto:contactdp@Developerpoint.com" class="unstyled">View form</a> from our<br/>
        site <br/> Contact no: <b> 9876-5432-89</b>
    </p>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<%@include file="common/footer.jsp"%>
<script type="text/javascript">
    function batchCourseEnquiry() {
        alert("Feel free to ask anything, and sit back freely we will reach you in 25min");
    }
</script>
</body>
</html>
```

## login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link rel="stylesheet" type="text/css" href="stylesheet/login.css">
<title>Login | Developers point</title>
</head>
<body>
    <%@include file="common/navbar.jsp"%>
    <div class="main-data d-flex justify-content-center align-items-center">
        <div class="form-outer">
            <form method="post" action="LoginServlet">
                <h2 class="my-1">
                    <i class="fa fa-user"></i> Login account
                </h2>
                <hr class="my-4">
```

```
<%
String invalidLogin = (String) session.getAttribute("login-failed");
if (invalidLogin != null) {
%>
<div class="alert alert-danger" role="alert"><%=invalidLogin%></div>
<%
session.removeAttribute("login-failed");
}
%>

<%
String logoutMsg = (String) session.getAttribute("logout-msg");
if (logoutMsg != null) {
%>
<div class="alert alert-success" role="alert"><%=logoutMsg%></div>
<%
session.removeAttribute("logout-msg");
}
%>
<%
String loginError = (String) session.getAttribute("Login-error");
if (loginError != null) {
%>
<div class="alert alert-danger" role="alert"><%=loginError%></div>
<%
}
session.removeAttribute("Login-error");
%>

<section>
<label>E-mail:</label> <br> <input type="text"
placeholder="E-mail Address" name="uEmail" required>
</section>
<br>
<section>
<label>Password:</label> <br> <input type="password" min="8"
max="16" placeholder="It was 8-16 character long"
required>
</section>
<br> <br> <input type="submit" value="Submit">
</form>
</div>
<%@include file="common/footer.jsp"%>
</body>
</html>
```

### register.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link rel="stylesheet" href="stylesheet/register.css">
<title>Register | Developers Point</title>
<style type="text/css">
.mustFill {
    color: red;
}
</style>
</head>
<body>
    <%@include file="common/navbar.jsp"%>

    <div class="main-data d-flex justify-content-center align-items-center">
        <div class="form-outer">
            <h2 class="my-2">
                <i class="fa fa-user-plus"></i> Register and Enjoy our Premium
                Content
            </h2>
            <hr class="my-3">
            <%
                String regMsg = (String) session.getAttribute("reg-sucess");
                if (regMsg != null) {
            %>
                <div class="alert alert-success" role="alert">
                    <%=regMsg%>
                    Login: <a href="Login.jsp">Click Here...</a>
                </div>
            <%
                session.removeAttribute("reg-sucess");
            }
                String FailedMsg = (String) session.getAttribute("reg-failed");
                if (FailedMsg != null) {
            %>
                <div class="alert alert-danger" role="alert">
                    <%=FailedMsg%>
                </div>
            <%
                session.removeAttribute("reg-failed");
            }
            <%
        </div>
        <form method="post" action="RegisterServlet">
            <section>
                <label>First Name:<span class="mustFill">*</span></label> <br>
                <input type="text" placeholder="First name" required

```

```
        name="uFirstName">
    </section>
    <br>
    <section>
        <label>Last Name:</label> <br> <input type="text"
            placeholder="Last name" name="uLastName">
    </section>
    <br>
    <section>
        <label>E-mail:</label><span class="mustFill">*</span> <br>
<input
name="uEmail">
            type="email" placeholder="E-mail Address" required
        </section>
        <br>
        <section>
            <label>Password:</label><span class="mustFill">*</span> <br>
<input
name="uPassword">
            type="password" min="8" max="16" required
            placeholder="8-16 character Long" name="uPassword"
            onkeyup="CheckCount()"> <label class="hidden"
            id="alert_custom" style="color: orange;">The password
            must
            be between 8-16 character long!!</label>
        </section>
        <br> <br> <input type="submit"
            class="btn btn-pill text-center vw-20" value="Submit"
            id="submitMe">
    </form>
    <span style="font-weight: lighter; font-size: small;"><span
        class="mustFill">*</span> Must fill to submit</span>
</div>
</div>
<script>
    function CheckCount() {
        var len = document.getElementsByName("uPassword")[0].value.length;
        console.log(len);
        if ((len < 8) || (len > 16)) {
            if (document.getElementById("alert_custom").classList
                .contains("hidden")) {
                document.getElementById("alert_custom").classList
                    .remove('hidden');
                document.getElementById("submitMe").setAttribute(
                    "disabled", "true");
            }
        } else {
            if (!document.getElementById("alert_custom").classList
                .contains('hidden')) {
                document.getElementById("alert_custom").classList
                    .add('hidden');
                document.getElementById("submitMe").removeAttributeNode(
                    document.getElementById("submitMe")
                        .getAttributeNode("disabled"));
            }
        }
    }
</script>
```

```
        }
    }
</script>

<%@include file="common/footer.jsp"%>
</body>
</html>
```

### student.jsp

```
<%@page import="java.util.Iterator"%>
<%@page import="entities.CourseList"%>
<%@page import="java.util.List"%>
<%@page import="org.hibernate.Criteria"%>
<%@page import="entities.BoughtCourses"%>
<%@page import="connection.DBConnection"%>
<%@page import="org.hibernate.Session"%>
<%@page import="entities.UserDetails"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<style type="text/css">
.bg-cus {
    background-color: #f5eaf7;
}
</style>
<title>Student | Developer Point</title>
</head>
<body>
    <%@include file="common/navbar.jsp"%>
    <%
Session sess = DBConnection.getFactory().openSession();
sess.beginTransaction();
Criteria criteria = sess.createCriteria(BoughtCourses.class);
List<BoughtCourses> col = criteria.list();
%>
    <div class="row" style="min-height: 78vh; width: 100vw;">
        <div class="col-xs-12 p-1 col-md-8">
            <div class="container-fluid">
                <div class="row text-center justify-content-center">
                    <%
```

```
int count = 0;
Iterator<BoughtCourses> itr = col.iterator();
while (itr.hasNext()) {
    BoughtCourses cl = itr.next();
    if (cl.getUserEmail().equals(userDetails.getEmail())) {
        count++;
    }
}
<div class="col-xs-12 col-sm-6 x mt-4">
    <div class="card mx-auto" style="width: 18rem;">
        
        <div class="card-body">
            <h5 class="card-title" style="word-break: break-all;"><%=cl.getCourseName()%></h5>
            <hr>
            <p class="card-text" style="word-break: break-all;">
                <%=cl.getCourseDescription()%>
            </p>
            <a href="courseBought.jsp?courseId=<%=cl.getCourseID()%>" class="btn btn-primary">More
                Detail</a>
            </div>
        </div>
    <%
    }
}
if (count < 1) {
%>
    <div class="shadow p-3 mb-5 bg-body rounded fs-3 mt-5">
        <strong>No course bought yet</strong>
        <div class="shadow p-3 mb-5 bg-body rounded fs-3 mt-3">
            <a class="btn btn-info px-3 w-100" role="button"
                href="courses.jsp">Search Courses</a>
        </div>
    <%
    }
%>
</div>
</div>
<div class="col-xs-12 p-5 col-md-4 bg-cus">
    <h2 class="text-center">
        Profile details
    <hr>
    <div class="text-muted fs-5 Lead">
        <div>
            Name:<br/>
            <%=userDetails.getFullName()%>
        </div>
    </div>
</div>
```

```
<div>
    <br> Email id:
    <%=userDetails.getEmail()%></div>
<br>
<div>
    Contact number:
    <%=userDetails.getContactNumber()%></div>
<div>
    <br> User Type:
    <%=userDetails.getUserType()%></div>
<div>
    <br> User since:
    <%=userDetails.getRegistrationDate()%></div>
<div>
    <br>
    <%=userDetails.toStringFullAddress()%></p>
    <a class="btn btn-info px-3 w-100" role="button"
        href="updateUser.jsp?userId=<%=userDetails.getEmail()%>">Edit</a>
    </div>
</div>

<%
sess.getTransaction().commit();
sess.close();
%>

<%@include file="common/footer.jsp"%>
</body>
</html>
```

## updateCourse.jsp

```
<%@page import="org.hibernate.Query"%>
<%@page import="entities.CourseList"%>
<%@page import="connection.DBConnection"%>
<%@page import="org.hibernate.Session"%>
<%
UserDetails userDetails1 = (UserDetails) session.getAttribute("loggedInUser");
if ((userDetails1 == null) || (!userDetails1.getUserType().equals("admin"))) {
    session.setAttribute("login-failed",
        "Only Admins can access this page.\n Login as admin");
    response.sendRedirect("LogoutServlet");
    //response.sendRedirect("login.jsp");
}
%><%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="ISO-8859-1">
<link rel="stylesheet" type="text/css" href="stylesheet/register.css">
<title>Update Course | DevelopersPoint</title>
</head>
<body>
    <%@include file="common/navbar.jsp"%>
    <%
        int id = Integer.parseInt(request.getParameter("courseID"));
        Session sess = DBConnection.getFactory().openSession();
        sess.beginTransaction();
        CourseList updateIt = (CourseList) sess.get(CourseList.class, id);
    %>
    <div class="container">
        <!-- Form begins here -->
        <form method="post" action="UpdateCourseServlet">
            <h2 class="text-center">Update Course</h2>
            <input type="hidden" name="courcesID"
                   value="<%updateIt.getCourseID()%>">
            <section>
                <label>Course Title:<span class="mustFill">*</span></label> <br>
                <input type="text" placeholder="Title to show on course" required
                       name="cTitle" value="<%updateIt.getCourseName()%>">
            </section>
            <br>
            <section>
                <label>Course Description:</label> <span
class="mustFill">*</span><br>
                <input type="text"
                       placeholder="Little explanation oF Learning material" required
                       name="cExplanation"
value="<%updateIt.getCouseDescription()%>">
            </section>
            <br>
            <section>
                <label>Price (in rs):</label><span class="mustFill">*</span> <br>
                <input type="number" placeholder="Course price without any discount"
                       required name="cPrice" value="<%updateIt.getCoursePrice()%>">
            </section>
            <br> <br> <input type="submit"
                           class="btn btn-pill text-center vw-20" value="Submit" id="submitMe">
        </form>
    </div>
    <%
        sess.getTransaction().commit();
        sess.close();
    %>
    <%@include file="common/footer.jsp"%>
</body>
</html>
```

### updateUser.jsp

```
<%@page import="org.hibernate.internal.build.AllowSysOut"%>
<%@page import="org.hibernate.Transaction"%>
<%@page import="org.hibernate.Session"%>
<%@page import="connection.DBConnection"%>
<%
UserDetails userDetailsService1 = (UserDetails) session.getAttribute("logedInUser");
if (userDetailsService1 == null) {
    /*session.setAttribute("login-failed",
    "Only Admin can access this page. Login as admin");
    response.sendRedirect("LogoutServlet");*/
    session.setAttribute("login-failed", "Login to access this page");
    response.sendRedirect("login.jsp");
}
%><%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<link rel="stylesheet" type="text/css" href="stylesheet/register.css">
<style type="text/css">
.mustFill {
    color: red;
}
</style>
<title>Admin update user details | DevelopersPoint</title>
</head>
<body>
<%@include file="common/navbar.jsp"%>
<div class="container" style="min-height: 78vh;">
    <%
        //Getting previous page.
        String url2 = request.getHeader("Referer");
        System.out.println(url2);
        String[] urlParts = url2.split("/");
        //System.out.println(urlParts[urlParts.length - 1]);
    <%
        String email = request.getParameter("userId");
        System.out.println(email);
        /*
            UserDetails updateIt = null;
            HttpSession session2 = null;
            Session sess = DBConnection.getFactory().openSession();
            Transaction tr = sess.beginTransaction();
            updateIt = (UserDetails) sess.load(UserDetails.class, email);
            session2 = request.getSession();
            session2.setAttribute("uid", updateIt);
            tr.commit();
            sess.close();
        */
        Session sess = DBConnection.getFactory().openSession();
    <%
}
```

```
sess.beginTransaction();
UserDetails updateIt = (UserDetails) sess.get(UserDetails.class, email);
if (updateIt != null) {
    //System.out.println(updateIt.toString());
%>
<!-- Form begins here -->
<form method="post" action="UpdateUserServlet">
    <h2 class="text-center display-5">Update user details with
        precautions</h2>
    add details in value and work in servlet to get them all and update
    in database. then work on courses page. then work on student page.
    <section>
        <label>First Name:<span class="mustFill">*</span></label> <br>
        <input type="text" placeholder="First name" required
               name="uFirstName" value="<%=updateIt.getFirstName()%>">
    </section>
    <br>
    <section>
        <label>Last Name:</label> <br> <input type="text"
            placeholder="Last name" name="uLastName"
            value="<%=updateIt.getLastName()%>">
    </section>
    <br>
    <section>
        <!-- Previous page url with parameter it gave -->
        <input type="hidden" name="previousPageUrl"
               value="<%=urlParts[urlParts.length - 1]%>"> <input
               type="hidden" placeholder="E-mail Address" required
               name="uEmail"
               value="<%=updateIt.getEmail()%>">
    </section>
    <br>
    <section>
        <label>Password:</label><span class="mustFill">*</span> <br> <input
            type="password" min="8" max="16" required
            placeholder="8-16 character long" name="uPassword"
            onkeyup="CheckCount()" value="<%=updateIt.getPassword()%>">
        <label class="hidden" id="alert_custom" style="color: orange;">The
            password must be between 8-16 character long!!</label>
    </section>
    <br>
    <section>
        <label>Type of user:- <span class="mustFill">(this
            user was <%=updateIt.getUserType()%>) </span>*</span>
        <br>
        <select
            class="form-select" name="uType"
            value="<%=updateIt.getUserType()%>"
            aria-label="Default select example" required>
            <option selected value="<%=updateIt.getUserType()%">Select
                user type</option>
            <option value="student">Student</option>
            <option value="faculty">Faculty</option>
            <option value="admin">Admin</option>
        </select>
    </section>
</form>
```

```
</section>
<br>
<section>
    <label>Contact no:</label><input type="text"
        placeholder="User Contact no" name="uContact"
        value="<%updateIt.getContactNumber()%>">
</section>
<br>
<section>
    <label>House-no:</label><input type="text"
        placeholder="House number" name="uHouse"
        value="<%updateIt.getHouseNo()%>">
</section>
<br>
<section>
    <label>Landmark:</label><input type="text"
        placeholder="Something easily visible to everyone"
        name="uLandmark"
        value="<%updateIt.getLandMark()%>">
</section>
<br>
<section>
    <label>City:</label><input type="text"
        placeholder="City user live in" name="uCity"
        value="<%updateIt.getCity()%>">
</section>
<br>
<section>
    <label>State:</label><input type="text"
        placeholder="State user live in" name="uState"
        value="<%updateIt.getState()%>">
</section>
<br>
    <section>
        <label>Pin Code:</label><input type="text"
            placeholder="Address full pin code" name="uPinCode"
            value="<%updateIt.getPinCode()%>">
    </section>
    <br> <br> <input type="submit"
        class="btn btn-pill text-center vw-20" value="Update" id="submitMe">
</form>
<%
sess.getTransaction().commit();
sess.close();
} else {
%>
<h2 class="text-center">User not found</h2>
<%
}
%>
</div>
<%@include file="common/footer.jsp"%>
</body>
</html>
```

### Forget.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<style type="text/css">
.main-data {
    display: flex;
    align-items: center;
    justify-content: center;
    min-height: 77vh;
    /*background-color: #c8ffff;*/

}

section>input {
    border: none;
    border-bottom: 1px solid red;
    padding: 2px 6px;
}

section>label {
    font-size: 14px;
}

section>input:focus {
    outline: none;
    border-bottom: 2.2px solid red;
}

input[type="submit"] {
    background: transparent;
    border: 1.5px solid #06dc06;
    width: 100%;
    margin-bottom: 2px;
}

input[type="submit"]:hover {
    margin-top: 2px;
    margin-bottom: 0px;
    background-color: #06dc06;
```

```
        box-shadow: 0px 0px 4px #06dc06, 2px 4px 2px 1px #2e822e;  
    }  
  
.forget-tip-form {  
background-color: white;  
width: 65%;  
min-width: 272px;  
margin: 45px 0px;  
box-shadow: 0px 1px 8px #ce93d8;  
transition: all 0.5s ease;  
box-shadow: 0px 1px 8px #ce93d8;  
}  
  
.forget-tip-form:hover {  
    box-shadow: 0px 4px 7px #ce93d8, inset 0px 1px 7px #d3e3e2b5;  
}  
  
.formsComboDiv {  
display: flex;  
min-height: 45vh;  
flex-direction: row;  
flex-wrap: wrap;  
justify-content: space-around;  
align-items: stretch;  
align-content: space-around;  
padding: 18px;  
margin: 1px;  
}  
  
.ResetForm, .hintForm {  
margin-bottom: 40px;  
}  
  
/>  
<title>Recover your password</title>  
</head>  
<body>  
    <%@include file="common/navbar.jsp"%>  
    <div class="form-outer main-data">  
        <div class="forget-tip-form ">  
            <h2 class="text-center my-4 ">  
                <i class="fa fa-question"></i> Forgot password  
            </h2>  
            <hr style="box-shadow: 0px 2px 3px black; color: white;">  
            <div class="formsComboDiv">  
                <div class="hintForm">  
                    <form method="get" action="ForgetPass" class="shadow"  
                        style="width: auto !important;">  
                        <sup style="color: blue; text-shadow: 1px 1px 1px  
green;">Get  
some hint for you</sup> <br> <br>  
                <section>  
                    <label>&nbsp;Enter E-mail:</label> <br> <input  
type="email"  
placeholder="Mail-id" name="uEmail">  
                </div>  
            </div>  
        </div>  
    </div>  
</body>
```

```
</section>
<br>
<section>
<%
String passi = request.getParameter("p");
%>
<label>&nbsp;Password hint:</label> <br> <input
type="text"
upper here"
null)%><%=passi%>">

session.getAttribute("userNotFound");

14px;"><%=userNotFound%></span>

<br> <sup> For more details contact us.</sup><br> <br>
<input type="submit" value="Submit">
</form>
</div>
<div class="ResetForm">
<form method="post" action="ResetPassword" class="shadow"
style="width: auto !important; max-width: 270px;">
<sup style="color: #f44a3d; text-shadow: 1px 1px 1px
password</sup> <br> <br>
<section>
<label>&nbsp;Enter Registered E-mail:</label>
<input type="email" placeholder="Mail-id"
>
</section>
<br>
<section>
<label>&nbsp;Registered Mobile no:</label> <br>
<input type="text" placeholder="Moblile no."
value="">
</section>
<%
String resetActionSituation = (String)
if (resetActionSituation != null) {
%>
```

```
<span style="color: orange; padding-top: 14px; padding-top: 10px; font-size: small;"><%=resetActionSituation%></span>
<%
}
session.removeAttribute("resetProbCause");
%>
<br> <sup> Never add fake detail</sup><br> <br> <input type="submit" value="Submit" onclick="waitmsg()">
</form>
</div>
</div>
</div>
<script type="text/javascript">
    function waitmsg() {
        alert("Click ok, and Wait few sec, we are fetching your details!!")
    }
</script>
<%@include file="common/footer.jsp"%>
</body>
</html>
```

### ForgetPass.java

```
package servlets;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.hibernate.Session;

import connection.DBConnection;
```

```
import entities.UserDetails;

public class ForgetPass extends HttpServlet {

    private static final long serialVersionUID = 1L;

    public ForgetPass() {
    }

    protected void doGet(HttpServletRequest request,
                         HttpServletResponse response) throws ServletException, IOException {
        String uEmail = request.getParameter("uEmail");
        System.out.println(uEmail);
        Session openSession = DBConnection.getFactory().openSession();
        UserDetails persist = openSession.get(UserDetails.class, uEmail);
        openSession.close();
        if (persist != null) {
            String mm = "";
            int len = persist.getPassword().length();
            for (int i = 0; i < len - 5; i++) {
                mm += '*';
            }
            String s = persist.getPassword().substring(0, 2) + mm
                    + persist.getPassword().substring(len - 3, len);
            response.sendRedirect("forget.jsp?p=" + s);
        } else {
            HttpSession sess = request.getSession();
            sess.setAttribute("userNotFound", "User Not Found");
        }
    }
}
```

```
        response.sendRedirect("forget.jsp");

    }

}

}
```

### ResetPassword.java

```
package servlets;

import java.io.IOException;
import java.util.List;
import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Message;
import javax.mail.MessagingException;
```

```
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import connection.DBConnection;
import entities.UserDetails;

public class ResetPassword extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public ResetPassword() {
    }

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {
        HttpSession sess = request.getSession();
        String email = request.getParameter("emailPassReset");
        String phone = request.getParameter("phonePassReset");
        if (!email.isBlank()) {
            UserDetails a = isMailInDatabase(email);
```

```
if (a != null) {  
    String emailFromDatabase = a.getEmail();  
  
    boolean validMail = Validate_code_source  
        .isAddressValid(emailFromDatabase);  
  
    if (validMail) {  
        try {  
            sendmail(email, a.getContactNumber(), a.getPassword());  
            sess.setAttribute("resetProbCause",  
                "Details has been sent to the "  
                + emailFromDatabase  
                + ", check for mail");  
            response.sendRedirect("forget.jsp");  
        } catch (MessagingException e) {  
            sess.setAttribute("resetProbCause",  
                "Server problem, try in 15 min.");  
            response.sendRedirect("forget.jsp");  
            e.printStackTrace();  
        }  
    } else {  
        sess.setAttribute("resetProbCause",  
            "User found attached to this E-mail id, but this  
            mail id is fake. Try with phone number.");  
        response.sendRedirect("forget.jsp");  
    }  
}  
  
} else if (a == null) {
```

```
sess.setAttribute("resetProbCause",
                  "No user found attached to this E-mail id.");
response.sendRedirect("forget.jsp");
}

} else if (!phone.isBlank()) {

    List<UserDetails> phoneInDatabase = isPhoneInDatabase(phone);
    System.out.println(phoneInDatabase);
    if (!phoneInDatabase.isEmpty())
        for (UserDetails userD : phoneInDatabase) {
            if (userD.getEmail() != null) {
                boolean validMail = Validate_code_source
                    .isAddressValid(userD.getEmail());
                if (validMail) {
                    try {
                        sendmail(userD.getEmail(),
                                userD.getContactNumber(),
                                userD.getPassword());
                        sess.setAttribute("resetProbCause",
                                          "Details has been sent to the
+ "
+ ", check for
attached mail: "
+ userD.getEmail()
details in it.");
                    }
                    response.sendRedirect("forget.jsp");
                } catch (MessagingException e) {

```

```
sess.setAttribute("resetProbCause",
                  "Server problem, try in 15
min.");
```

```
        response.sendRedirect("forget.jsp");
        e.printStackTrace();
    }
} else {
    sess.setAttribute("resetProbCause",
                      "User found attached to this number,
but the mail id is fake. Try contact us.");
    response.sendRedirect("forget.jsp");
}
} else if (userD.getEmail() == null) {
    sess.setAttribute("resetProbCause",
                      "User found with number, but has no
attached mail id.");
    response.sendRedirect("forget.jsp");
}
else {
    sess.setAttribute("resetProbCause",
                      "No user found attached to this phone number. User hint
or contact us through contact form");
    response.sendRedirect("forget.jsp");
}
} else {
```

```
        sess.setAttribute("resetProbCause", "Fill atleast one field.");
        response.sendRedirect("forget.jsp");
    }

}

UserDetails isMailInDatabase(String email) {
    org.hibernate.Session sess = DBConnection.getFactory().openSession();
    sess.beginTransaction();
    UserDetails usd = (UserDetails) sess.get(UserDetails.class, email);
    sess.close();
    return usd;
};

public void sendmail(String recipient) throws MessagingException {
    Properties prop = new Properties();
    prop.setProperty("mail.smtp.auth", "true");
    prop.setProperty("mail.smtp.starttls.enable", "true");
    prop.setProperty("mail.smtp.host", "smtp.gmail.com");
    prop.setProperty("mail.smtp.port", "587");

    // Sender
    final String myAddress = "ujjwalpandey.apps@gmail.com";
    final String pass = "Goldenstar@123";

    Session sess = Session.getInstance(prop, new Authenticator() {
        @Override
        protected PasswordAuthentication getPasswordAuthentication() {
```

```
        return new PasswordAuthentication(myAddress, pass);

    }

});

Message mess = prepareMessage(sess, myAddress, recipient);

Transport.send(mess);

}

public void sendmail(String recipient, String phone, String password)

throws MessagingException {

Properties prop = new Properties();

prop.setProperty("mail.smtp.auth", "true");

prop.setProperty("mail.smtp.starttls.enable", "true");

prop.setProperty("mail.smtp.host", "smtp.gmail.com");

prop.setProperty("mail.smtp.port", "587");



// Sender

final String myAddress = "ujjwalpandey.apps@gmail.com";

final String pass = "Goldenstar@123";



Session sess = Session.getInstance(prop, new Authenticator() {

    @Override

    protected PasswordAuthentication getPasswordAuthentication() {

        return new PasswordAuthentication(myAddress, pass);

    }

});

Message mess = prepareMessage(sess, myAddress, recipient, phone,

password);

Transport.send(mess);
```

}

```
private static Message prepareMessage(Session sess, String myAddress,
                                     String recipient) {
    try {
        Message mess = new MimeMessage(sess);
        mess.setFrom(new InternetAddress(myAddress));
        mess.setRecipient(Message.RecipientType.TO,
                          new InternetAddress(recipient));
        mess.setSubject("Developers' Point Details");
        // mess.setText("This is going to be the body.");
        String mailBody = "<center> The number registered with " + recipient
                         + "</center>";
        mess.setContent(mailBody, "text/html");
        return mess;
    } catch (MessagingException e) {
        e.printStackTrace();
        System.out.println("Some extra");
    }
    return null;
};

private static Message prepareMessage(Session sess, String myAddress,
                                     String recipient, String phone, String pass) {
    try {
        Message mess = new MimeMessage(sess);
        mess.setFrom(new InternetAddress(myAddress));
        mess.setRecipient(Message.RecipientType.TO,
```

```
        new InternetAddress(recipient));

    mess.setSubject("Developers' Point Details");

    // mess.setText("This is going to be the body.");

    String mailBody = "<center> Details attached to mail id: "
        + recipient + " are: <br> Phone number: <b> " + phone
        + "</b>, <br> Password: <b>" + pass + "</b>"
        + "<br> <a href='http://localhost:8081/User_Faculty_Institute_Workspace/login.jsp'>click here</a> "
        + "to login.<br>"
        + "<b>Tip: Change password every 4 month to stay secure.</b>"
        + "</center>";

    mess.setContent(mailBody, "text/html");

    return mess;

} catch (MessagingException e) {
    e.printStackTrace();
    System.out.println("Some extra");
}

return null;
}

List<UserDetails> isPhoneInDatabase(String phone) {
    org.hibernate.Session sess1 = DBConnection.getFactory().openSession();
    List<UserDetails> resultList = sess1
        .createQuery("FROM UserDetails where contactNumber = " + phone,
                    UserDetails.class)
        .getResultList();

    sess1.close();

    return resultList;
}
```

```
};  
  
}
```

### Validate\_code\_source.jsp

```
package servlets;  
  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
import java.net.Socket;  
import java.util.ArrayList;  
import java.util.Hashtable;  
  
import javax.naming.NamingEnumeration;  
import javax.naming.NamingException;  
import javax.naming.directory.Attribute;  
import javax.naming.directory.Attributes;  
import javax.naming.directory.DirContext;
```

```
import javax.naming.directory.InitialDirContext;

public class Validate_code_source {

    public static void main(String args[]) {

        String testData[] = {"tvf@tvf.cz", "ujjwalpandey.apps@gmail.com",
                            "ujjwalpandey.apps@gmail.com", "thisIsFake@gmail.com"};

        for (int i = 0; i < testData.length; i++) {

            System.out.println(
                testData[i] + " is valid? " + isAddressValid(testData[i]));
        }
        return;
    }

    private static int hear(BufferedReader in) throws IOException {

        String line = null;
        int res = 0;
        while ((line = in.readLine()) != null) {

            String pfx = line.substring(0, 3);
            try {
                res = Integer.parseInt(pfx);
            } catch (Exception ex) {
                res = -1;
            }
            if (line.charAt(3) != '-')
                break;
        }
        return res;
    }
}
```

```
private static void say(BufferedWriter wr, String text) throws IOException {
    wr.write(text + "\r\n");
    wr.flush();
    return;
}

private static ArrayList getMX(String hostName) throws NamingException {
    // Perform a DNS lookup for MX records in the domain
    Hashtable env = new Hashtable();
    env.put("java.naming.factory.initial",
            "com.sun.jndi.dns.DnsContextFactory");

    DirContext ictx = new InitialDirContext(env);
    Attributes attrs = ictx.getAttributes(hostName, new String[]{"MX"});
    Attribute attr = attrs.get("MX");
    // if we don't have an MX record, try the machine itself
    if ((attr == null) || (attr.size() == 0)) {
        attrs = ictx.getAttributes(hostName, new String[]{"A"});
        attr = attrs.get("A");
        if (attr == null)
            throw new NamingException(
                "No match for name '" + hostName + "'");
    }
    // Huzzah! we have machines to try. Return them as an array list
    // NOTE: We SHOULD take the preference into account to be absolutely
    // correct. This is left as an exercise for anyone who cares.
    ArrayList res = new ArrayList();
    NamingEnumeration en = attr.getAll();
    while (en.hasMore()) {
```

```
        String x = (String) en.next();

        String f[] = x.split(" ");

        if (f[1].endsWith("."))

            f[1] = f[1].substring(0, (f[1].length() - 1));

        res.add(f[1]);

    }

    return res;
}

public static boolean isAddressValid(String address) {

    // Find the separator for the domain name

    int pos = address.indexOf('@');

    // If the address does not contain an '@', it's not valid

    if (pos == -1)

        return false;

    // Isolate the domain/machine name and get a list of mail exchangers

    String domain = address.substring(++pos);

    ArrayList mxList = null;

    try {

        mxList = getMX(domain);

    } catch (NamingException ex) {

        return false;
    }

    // Just because we can send mail to the domain, doesn't mean that the
    // address is valid, but if we can't, it's a sure sign that it isn't

    if (mxList.size() == 0)

        return false;

    // Now, do the SMTP validation, try each mail exchanger until we get
```

```
// a positive acceptance. It *MAY* be possible for one MX to allow
// a message [store and forwarder for example] and another [like
// the actual mail server] to reject it. This is why we REALLY ought
// to take the preference into account.

for (int mx = 0; mx < mxList.size(); mx++) {

    boolean valid = false;

    try {

        int res;

        Socket skt = new Socket((String) mxList.get(mx), 25);

        BufferedReader rdr = new BufferedReader(
            new InputStreamReader(skt.getInputStream()));

        BufferedWriter wtr = new BufferedWriter(
            new OutputStreamWriter(skt.getOutputStream()));

        res = hear(rdr);

        if (res != 220)
            throw new Exception("Invalid header");

        say(wtr, "EHLO orbaker.com");
        res = hear(rdr);

        if (res != 250)
            throw new Exception("Not ESMTP");

        // validate the sender address

        say(wtr, "MAIL FROM: <tim@orbaker.com>");
        res = hear(rdr);

        if (res != 250)
            throw new Exception("Sender rejected");

        say(wtr, "RCPT TO: <" + address + ">");
        res = hear(rdr);
```

```
// be polite
say(wtr, "RSET");
hear(rdr);
say(wtr, "QUIT");
hear(rdr);
if (res != 250)
    throw new Exception("Address is not valid!");
valid = true;
rdr.close();
wtr.close();
skt.close();
} catch (Exception ex) {
    // Do nothing but try next host
} finally {
    if (valid)
        return true;
}
}
return false;
}
public String call_this_to_validate(String email) {
String testData[] = {email};
String return_string = "";
for (int ctr = 0; ctr < testData.length; ctr++) {
    return_string = (testData[ctr] + " is valid? "
        + isAddressValid(testData[ctr]));
}
}
```

```
        return return_string;  
  
    }  
  
}
```

### payment.jsp

```
<%@page import="entities.CourseList"%>  
<%@page import="org.hibernate.Criteria"%>  
<%@page import="connection.DBConnection"%>  
<%@page import="org.hibernate.Session"%>  
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"  
    pageEncoding="ISO-8859-1"%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="ISO-8859-1">  
<link rel="stylesheet" type="text/css" href="stylesheet/payment.css">  
<style type="text/css">  
@import  
    url('https://fonts.googleapis.com/css2?family=KoHo&family=Varela+Round&display=swap')  
    ;  
  
.bg-cus {  
    background-color: #f5eaf7;  
  
.main {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    flex-direction: column;  
    flex-wrap: nowrap;  
    min-height: 70vh;  
    margin: 0;  
}  
  
.main-in {  
    min-width: 75%;
```

```
max-width: 80%;  
height: 100%;  
border-bottom-right-radius: 30px;  
border-top-left-radius: 18px;  
box-shadow: 11px -11px 14px 2px #9ac1ff, -9px 9px 10px 0px #f2c8ff;  
margin: 80px auto;  
padding: 12px 15px;  
position: relative;  
transition: all .7s ease;  
height: 100%  
}  
  
.tag {  
top: 100px;  
visibility: hidden;  
background: transparent;  
right: -5px;  
position: absolute;  
padding: 3px 15px 3px 21px;  
border-radius: 59px 0 0 10px;  
box-shadow: 0px 0px 3px #39ecf8;  
z-index: -1;  
background: transparent;  
transition-delay: 0.91s;  
transition: all .6s cubic-bezier(0, 0.34, 0.33, 1.06), visibility 0.7s  
ease 0.001s;  
}  
  
.main-in:hover {  
border-top-right-radius: 30px;  
border-bottom-left-radius: 30px;  
box-shadow: 0px 0px 8px 1px #2141d5, 13px -15px 14px 0px #9ac1ff, -13px  
13px 15px 0px #d869f9;  
/*border-top-left-radius: 0px;*/  
}  
  
.main-in:hover .tag {  
z-index: 1;  
visibility: visible;  
}  
  
.main-in:hover .payment-heading {  
transition: text-shadow .7s ease;  
text-shadow: 0px 0px 2.7px blue, 4px 8px 7px #9de0f0;  
}  
  
.tag:hover {  
border-radius: 59px 0 0 10px;  
box-shadow: -14px 0px 2px 2px #6b41fc;  
font-size: larger;  
transition-delay: 0.001s;  
}  
  
.payment-description {
```

```
font-family: 'KoHo', sans-serif;
line-height: 3.3rem;

}

.payment-description span {
    font-family: 'Varela Round', sans-serif;
}

.button {
    color: blue;
    text-decoration: blink;
    margin-top: 10px;
    margin-right: 5px;
    padding: 5px 10px;
    display: inline-block;
}

.button:hover, button:hover {
    color: blue;
    text-shadow: 0px 0px 4px #5df0f7;
    margin-top: 5px;
    box-shadow: 1px 3px 2px blue;
    border-radius: 20px;
    border: 1px solid blue;
    margin-top: 5px;
    box-shadow: 1px 3px 2px blue;
    border-radius: 20px;
    box-shadow: 1px 3px 2px blue;
    border-radius: 20px;
    border-radius: 20px;
}

.bg-linear-grad {
    background: linear-gradient(180deg, #9458ca, #BE63FF, #b977f4, #BE63FF, #9978db);
}

.flex-modal {
    display: flex;
    justify-content: center;
    align-items: center;
    align-content: center;
    flex-direction: column;
    flex-wrap: nowrap;
    flex-direction: column;
}

.flex-modal-in {
    min-height: 10vh;
    display: flex;
    flex-direction: column;
    justify-content: center;
    flex-wrap: nowrap;
    align-items: center;
    width: 90%;
```

```
}

.flex-modal-in>div {
  width: 100%;
  margin: 10px 0;
  background: linear-gradient(180deg, #7706d8, #ae67ec);
  /*box-shadow: -1px 2px 6px #88b6a5;*/
  border-radius: 6px;
  padding: 15px;
  font-size: 45px;
  margin: 10px 0;
}

.cardDetails {
  background: white;
  display: flex;
  color: black;
  border-radius: 0px;
  font-size: 15px;
  flex-direction: column;
  flex-wrap: nowrap;
  justify-content: center;
  align-items: flex-start;
}

.cardDetails>div {
  padding: 6px;
  width: 100%;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
}

.cardDetails>div>input[type="text"]
width: 100%;

.modal-footer>a {
  color: white;
}

.payment-heading {
  margin-top: 30px;
  font-family: cursive;
  font-size: 2.5rem;
  font-weight: bold;
  text-shadow: 0px 0px 2.7px blue;
}

.main-in:hover>.payment-heading:hover {
  text-shadow: 0px 0px 2.7px blue, -9px 6px 7px #9de0f0
}

</style>
<title>Course detail | DevelopersPoint</title>
```





```
placeholder="yyyy" maxlength="4">
</div>
<div style="text-align: left;">
  <small>CVC</small><br> <input type="text"
    placeholder="mm" maxlength="3">
  <input type="text"
    placeholder="Full Name">
  style="width: 100%;"
  href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>">
  Pay Now</a>
  <a href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>" class="button rounded text-center">Pay Now</a>
  <a href="#" class="button" data-dismiss="modal">Close</a> <a href="#" class="button" data-dismiss="modal" style="width: 75%;"
  href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>" class="button rounded text-center">Pay Now</a>
</div>
</div>
<!-- Modal Debit card-->
<div class="modal fade" id="debitCardPay" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    #f4f8c3">
    !Important,>
    <div class="modal-content bg-linear-grad" style="color: #f4f8c3">
      <div class="modal-header" style="display: block">
        <h3 class="modal-title mt-2 display-7" id="exampleModalLongTitle">Checkout</h3>
      </div>
```



```
</div>
</div>
</div>
</div>
<div class="modal-footer">
    <a type="button" class="button" data-
dismiss="modal ">Close</a> <a
        href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>" type="button" style="width: 75%;"
        class="button rounded text-
center">Pay Now</a>
</div>
</div>
</div>
<!-- Modal Pay pal -->
<div class="modal-fade" id="paypalPay" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content bg-linear-grad" style="color: #f4f8c3;">
            <div class="modal-header" style="display: block">
                <h3 class="modal-title mt-2 display-7" id="exampleModalLongTitle">Checkout</h3>
            </div>
            <div class="modal-body flex-modal" style="">
                <div class="flex-modal-in">
                    <div>
                        <%=specific.getCoursePrice() - (specific.getCoursePrice() * specific.getDiscount() / 100)%>
                        <sup><i class="fas fa-rupee-
sign"></i></sup> <br> <span
color: black; border-radius: 0px; font-size: 15px;">Pay to Developers' Point pvt.
15px;">Pay Using PayPal</div>
<input type="text"
placeholder="XXXX@paypal.com" maxlength="30">
</div>
</div>
<div style="background: white; class="cardDetails">
<div style="font-size: .text-muted">
    Enter Pay pal id: <br>
</div>
</div>
```

```
Pay Now  
Close  
Pay Now  


Important:



Checkout



- (specific.getCoursePrice() * specific.getDiscount() / 100)%  
sign"></i></sup> <br> <span style="color: black; border-radius: 0px; font-size: 15px;">15px; >Pay to Developers' Point pvt.



18px;">Pay using PayPal</div>



<%=specific.getcoursePrice() - (specific.getcoursePrice() * specific.getdiscount() / 100)%>  
<sup><i class="fas fa-rupee-sign"></i></sup> <br> <span style="color: black; border-radius: 0px; font-size: 15px;">15px; >Pay to Developers' Point pvt.

18px;">Pay using PayPal</div>



<div class=".text-muted">


```

Enter upi id: <br>

```
<input type="text"
       placeholder="XXXX@upiProvider.com" maxlength="30">
</div>

<div>
    <a type="button" href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>" class="button rounded text-center">Pay Now</a>
    <a type="button" class="button fs-6" data-dismiss="modal">Close</a>
</div>
<div class="modal-footer ">
    <a type="button" class="button fs-6" href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>">Pay Now</a>
    <a type="button" style="width: 75%; border-radius: 10px; background-color: #007bff; color: white; padding: 5px 15px; font-weight: bold; margin-left: 10px;" href="#">Bank Transfer</a>
</div>
</div>

<div class="modal fade" id="netBankingPay" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content bg-linear-grad" style="color: #f4f8c3;">
            <div class="modal-header" style="display: block; text-align: center; margin-bottom: 10px;">
                <h3 class="modal-title mt-2 display-7" id="exampleModalLongTitle">Checkout</h3>
            </div>
            <div class="modal-body flex-modal" style="">
                <div class="flex-modal-in">
                    <div style="text-align: center; margin-bottom: 10px;">
                        <math>\text{Rs. } \frac{\text{specific.getCoursePrice()}}{100} \times (\text{specific.getDiscount()} / 100)\%</math>
                    </div>
                    <sup><i class="fas fa-rupee-sign"></i></sup> <br> <span style="font-size: 15px;">Pay to Developers' Point pvt. ltd.</span>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
color: black; border-radius: 0px; font-size: 15px;" style="background: white; class="cardDetails"><div style="font-size: 18px;">Pay using Net Banking</div><br> <input type="text" placeholder="XXXX@bank.com" maxlength="30" style="width: 100%; "><a href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>" class="button rounded text-center">Pay Now</a><a href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>" class="button rounded text-center fs-6" data-dismiss="modal ">Close</a><a href="BuyCourseServlet?userId=<%=userDetails.getEmail()%>" class="button rounded text-center fs-6" data-dismiss="modal ">Pay Now</a><div class="modal-footer "><a type="button" class="button fs-6" data-dismiss="modal ">Close</a><a type="button" style="width: 75%; ">Pay Now</a></div><!-- Models begin here --><div><%> sess.getTransaction().commit(); sess.close(); %><script type="text/javascript"> function batchCourseEnquiry() { alert("Feel free to ask anything, and sit back freely we will reach you in 25min"); } //onclick="batchCourseEnquiry()" $(document).ready(function() { $('[data-toggle="tooltip"]').tooltip();}); </script>
```

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></script>
>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<%@include file="common/footer.jsp"%>
</body>
</html>
```

## 6. Tables used (Screenshots of real table)

UserDetails

userdetails courselist courselist userdetails x

1 • SELECT \* FROM institute\_workspace.userdetails;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

email	city	contactNumber	firstName	houseNo	landMark	lastName	password	pinCode	regstrationDate	state	userType
admin@gmail.com	NULL	0	Admin	NULL	NULL		admin4567	NULL	15-04-2021, 18:32:53	NULL	admin
facman3@gmail.com	cp	7777777777	Fac 3	789	fruit shop	man	fackman	45	26-04-2021, 20:16:15	New Delhi	Faculty
facman4@gmail.com	cp	9999999999	hi si	3	bus stand	mumu	mumumumu	891231	22-03-2021, 20:16:15	New Delhi	student
faculty@g.com	delhi	997789863	faculty	any	any	man	12345678	55	18-04-2021, 23:46:19	rsg	faculty
maanvi@mail.com	delhi	04545454	Maanvi	ramadiv	fasdf	Pandey	maanvi1234	45	19-04-2021, 19:50:50	new delhi	student
manishaPande@mani.com	sangam	92929299229	manisha	G-12	near, lakshman medicose	kumari	123123123	45	29-03-2021, 7:16:15	New Delhi	student
rahulpal@gmail.com	delhi	1119	Rahul	45	neem chwak	Pal	12345678	110060	20-04-2021, 16:53:42	new delhi	student
s@s.s	delhi	837599050	Maanvi	45	sbi bank, pati gagli	Pandey	123456789	110040	19-04-2021, 18:31:42	new delhi	faculty
ujjwalpandey.aps@gmail.com	delhi	9876543210	Ujjwal	54	monday market	Pandey ji	12345678	110060	14-04-2021, 18:21:59	new Delhi	student
ujjwalpandey.aps3@gmail....	NULL	NULL	Ujjwal 3	NULL	NULL	Pandey3	12345678	110060	24-04-2021, 20:20:15	NULL	student
user10@gmail.com	Rohini	999999999	vikas	2	neat, masjid	pandey	yownnnn	110088	25-04-2021, 20:16:15	New Delhi	student
user9@gmail.com	Tokyo	123504158	hi	34	chomin shop	ha	888888888	220001	25-04-2021, 20:15:10	China	student
veenadeci@mail.com	hongk...	3928347	veena	98	5 star hotel	pandey	2324	123	02-03-2021, 12:16:15	handff	faculty
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

userdetails 1 x

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:24:13	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	8 row(s) returned	0.000 sec / 0.000 sec
2	15:38:03	SELECT * FROM institute_workspace.courselist LIMIT 0, 100	6 row(s) returned	0.000 sec / 0.000 sec
3	15:38:28	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	13 row(s) returned	0.000 sec / 0.000 sec

### Couselist

userdetails    courselist    **couselist**    userdetails

File    Edit    View    Insert    Delete    Refresh    Limit to 100 rows    Favorites    Help

1 • `SELECT * FROM institute_workspace.couselist;`

Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap Cell Content: [ ]

courseID	courseAddedDate	courseImage	courseName	coursePrice	courseDescription
4	2021-04-16	pythone.jpg	Learn Python from basics	7899	This course will introduce to the power of python...
9	2021-04-16	mysql.jpg	lear MySQL Beginner to Pro	3999	We will learn all the concepts of MySql and work...
10	2021-04-18	javascript.png	Learn JavaScript Beginner to Pro in less than 9hrs.	5999	Here we will learn all the important basic of java...
11	2021-04-18	mysql.jpg	MySQL for analysis	3669	We will learn how to use MySQL to do 'Data Ana...
12	2021-04-18	html.jpg	Learn HTML-5 with Developers point.	790	You will learn html5 basic and get to know best ...
13	2021-04-19	mongoDB.png	Database MongoDB	2998	Learn mongoDB from top educator of Developer...
NULL	NULL	NULL	NULL	NULL	NULL

courselist1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:24:13	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	8 row(s) returned	0.000 sec / 0.000 sec
2	15:38:03	SELECT * FROM institute_workspace.couselist LIMIT 0, 100	6 row(s) returned	0.000 sec / 0.000 sec
3	15:38:28	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	13 row(s) returned	0.000 sec / 0.000 sec

BY

boughtcourses

userdetails    counselist    counselist   .userdetails    boughtcourses x

File Edit View Insert Tools Window Help

1 • SELECT \* FROM institute\_workspace.boughtcourses;

Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap Cell Content: [ ]

buyingID	courseID	courseImage	courseName	coursePrice	courseDescription	userEmail
1	13	mongoDB.png	Database MongoDB	2998	Learn mongoDB from top educator of Developer...	admin@gmail.com
2	9	mysql.jpg	lear MYSQL Beginner to Pro	3999	We will learn all the concepts of MySql and work...	admin@gmail.com
3	11	mysql.jpg	MySQL for analysis	3669	We will learn how to use MySql to do data analysis	admin@gmail.com
4	11	mysql.jpg	MySQL for analysis	3669	We will learn how to use MySql to do data analysis	admin@gmail.com
5	4	pythone.jpg	4 updste 2	5555	A try to edit 5	maanvi@mail.com
6	10	javascript.png	Learn JavaScript Beginer to Pro in less than 16hrs.	5999	Here we will learn all the important basic of java...	ujjwalpandey.aps@gmail.com
7	13	mongoDB.png	Database MongoDB	2998	Learn mongoDB from top educator of Developer...	ujjwalpandey.aps@gmail.com
8	12	html.jpg	tyr cour edit	790	sdfdasdfh	rahulpal@gmail.com
9	12	html.jpg	Learn HTML-5 with Developers point.	790	You will learn html5 basic and get to know best ...	rahulpal@gmail.com
HULL	HULL	HULL	HULL	HULL	HULL	HULL

boughtcourses 1 x

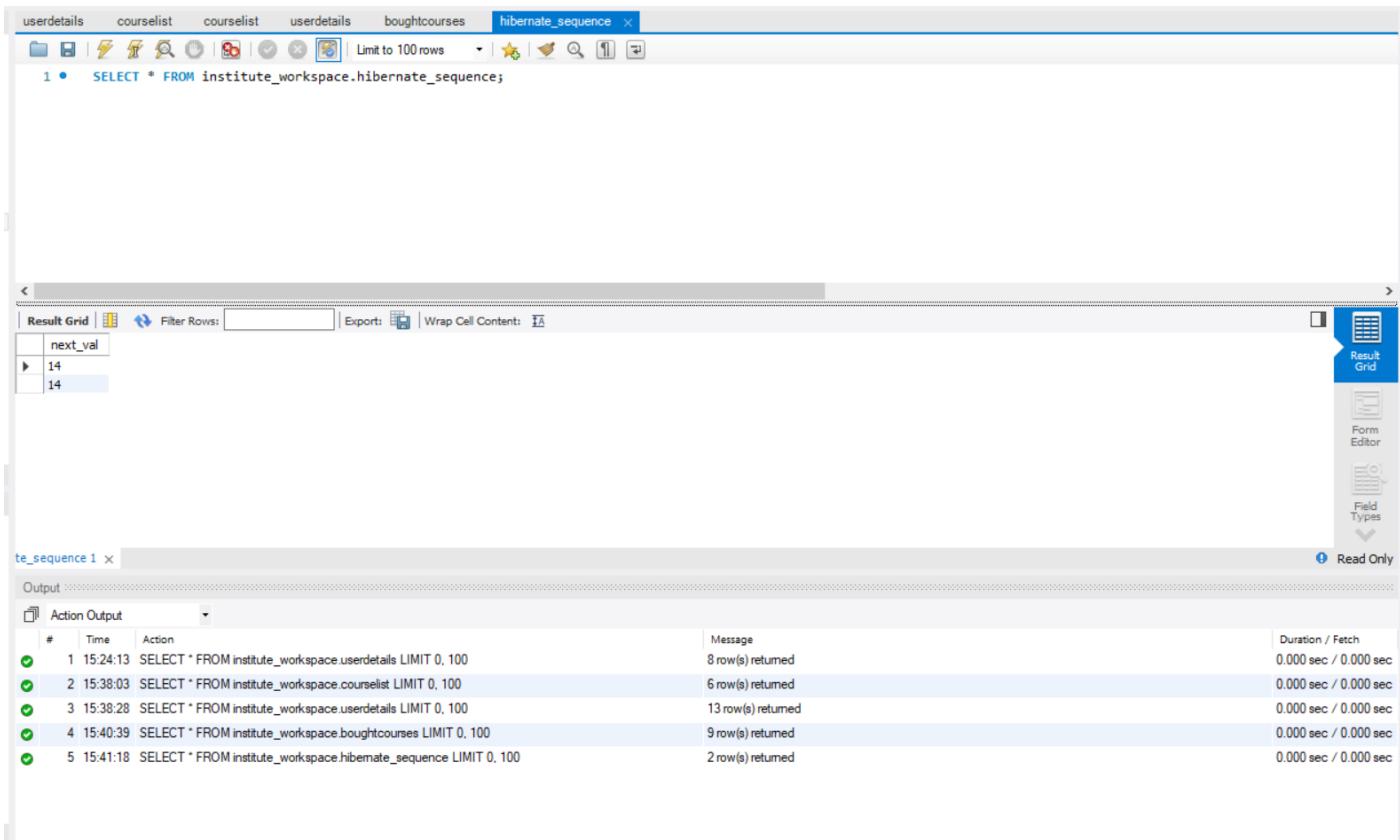
Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	15:24:13	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	8 row(s) returned	0.000 sec / 0.000 sec
2	15:38:03	SELECT * FROM institute_workspace.counselist LIMIT 0, 100	6 row(s) returned	0.000 sec / 0.000 sec
3	15:38:28	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	13 row(s) returned	0.000 sec / 0.000 sec
4	15:40:39	SELECT * FROM institute_workspace.boughtcourses LIMIT 0, 100	9 row(s) returned	0.000 sec / 0.000 sec

BYUJ

### hibernate\_sequence



The screenshot shows a database interface with the following details:

- Table:** hibernate\_sequence
- Query:** SELECT \* FROM institute\_workspace.hibernate\_sequence;
- Result Grid:** Shows one row with the column "next\_val" containing the value 14.
- Action Output:** Displays the history of database actions with the following log:

#	Time	Action	Message	Duration / Fetch
1	15:24:13	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	8 row(s) returned	0.000 sec / 0.000 sec
2	15:38:03	SELECT * FROM institute_workspace.couselist LIMIT 0, 100	6 row(s) returned	0.000 sec / 0.000 sec
3	15:38:28	SELECT * FROM institute_workspace.userdetails LIMIT 0, 100	13 row(s) returned	0.000 sec / 0.000 sec
4	15:40:39	SELECT * FROM institute_workspace.boughtcourses LIMIT 0, 100	9 row(s) returned	0.000 sec / 0.000 sec
5	15:41:18	SELECT * FROM institute_workspace.hibernate_sequence LIMIT 0, 100	2 row(s) returned	0.000 sec / 0.000 sec

queryfromform

userdetails    counsellist    counsellist   .userdetails    boughtcourses    hibernate\_sequence    queryfromform x

1 • SELECT \* FROM institute\_workspace.queryfromform;

Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap Cell Content: [ ]

qSoNo	qContact	qEmail	qName	qQueryExplained	qQueryHead
3	3456578	ujjwalpandey.aps@gmail.com	Ujjwal Pandey	adfsd2	fa
4	3456578	ujjwalpandey.aps@gmail.com	Ujjwal Pandey	adfsd2d.MetamodelImpl.entityPersister(MetamodelImpl.java:704) org.hibernate.internal.SessionImpl.getEntityPersister(SessionImpl.java:1705) org.hibernate.event.internal.AbstractSaveEventListener.saveWithGeneratedId(AbstractSaveEventListener.java:122)  org.hibernate.event.internal.DefaultSaveOrUpdateEventListener.saveWithGeneratedOrRequestedId(DefaultSaveOrUpdateEventListener.java:38) org.hibernate.event.internal.DefaultSaveOrUpdateEventListener.entityIsTransient(DefaultSaveOrUpdateEventListener.java:177) org.hibernate.event.internal.DefaultSaveEventListener.performSaveOrUpdate(DefaultSaveEventListener.java:32) org.hibernate.event.internal.DefaultDeserive( HttpServlet.java:648) javax.servlet.http.HttpServlet.service( HttpServlet.java:729) org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52) note The full stack trace of the root cause is available in the Apache Tomcat/8.0.27 logs.  .MetamodelImpl. org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52) note The full stack trace of the root cause is available in the Apache Tomcat/8.0.27 logs.	fa
5		ujjwalpandey.aps@gmail.com	Ujjwal Pandey	try 3 Adding notification that the query has submitted.	notification of adding
6	fa	ujjwalpandey.aps@gmail.com	Ujjwal Pandey	aise hi	HULL
*	HULL	HULL	HULL	HULL	HULL

queryfromform 5 x

Output:

Action Output

#	Time	Action	Message	Duration / Fetch
5	15:41:18	SELECT * FROM institute_workspace.hibernate_sequence LIMIT 0, 100	2 row(s) returned	0.000 sec / 0.000 sec
6	15:42:02	SELECT * FROM institute_workspace.queryfromform LIMIT 0, 100	6 row(s) returned	0.015 sec / 0.000 sec
7	15:43:29	SELECT * FROM institute_workspace.queryfromform LIMIT 0, 100	6 row(s) returned	0.000 sec / 0.000 sec
8	15:43:52	SPLIT * FROM institute_workspace.queryfromform LIMIT 0, 100	6 row(s) returned	0.000 sec / 0.000 sec

BY UJWAL PANDAY

## 7. Outputs

### Index page

anyone can visit  
(No login required)

Our latest course section is dynamic it shows latest added course.



Developers' Point Courses Contact Register Login

### Our Latest Course

**Title:** Database MongoDB  
**Description:** Learn mongoDB from top educator of Developers' point, MongoDB no need to care about datatypes.  
[More Detail](#)

**Title:** Learn JavaScript Beginner to Pro in less than 16hrs.  
**Description:** Here we will learn all the important basic of javascript and go upto the pro level to learn the concepts in deep.  
[More Detail](#)

### New Batch

**Teaching:**Front-end Web Development  
**Description:** Learn how to develop the web with the help of HTML, CSS, JavaScript, React, Bootstrap, etc  
**Batch Days:** MWF  
**Things:** Things are flexible before starting the batch, take your step and bend them according to your suitability  
[More Detail](#)

**Subject:** Full Stack Development  
**Description:** How to use HTML, CSS, JavaScript, Java, Advanced Java, SQL, etc...  
**Batch Days:** TTS  
**Things:** Things are flexible before starting the batch, take your step and bend them according to your suitability  
[More Detail](#)

### Our Exemplar

**Alina Gomes**

**Our Silver-ranked**  
Best AI developer of 2107 with salary of 35 L.P.A.

**UJJWAL PANDEY**

**Our Golden student**  
Got placement in google with salary of 40 L.P.A.

**Madara Uchia**

**Our Bronze student**  
Best Data Scientist of year 2019 with salary of 30 L.P.A.

Our world wide best in best faculty

Register Login

PROVIDING EXCELLENCE TO THE FUTURE  
We are those who provide the best in best faculty to teach student ensuring the best future for them

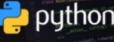
ADDRESS  
36,  
Ground floor,  
near bank of baroda,  
Sankat Vihar,  
Dew Delhi, Delhi 110067

CONTACT US  
View Email:  
[contactdp@Developerpoint.com](mailto:contactdp@Developerpoint.com)  
[View form](#) from our site  
Contact no: 9876-5432-89

© 2020 Copyright: Developers Point

Developer's Point Home Courses Contact Admin logout

## Course



### Learn Python from basics

This course will introduce to the power of python and do clear your all doubts.

[More Detail](#)



### Learn MySQL Beginner to Pro

We will learn all the concepts of MySQL and work on 3 projects. update

[More Detail](#)



### Introduction To JavaScript

Learn JavaScript Beginner to Pro

Here we will learn all the important basic of javascript and go upto the pro level to learn the concepts in depth.

[More Detail](#)



### MySQL for analysis

We will learn how to use MySQL to do 'Data Analysis'

[More Detail](#)



### Learn HTML-5 with Developers point.

You will learn html5 basic and get to know best practice tools, with some other cool sources to learn anything for free.

[More Detail](#)



### Database MongoDB

Learn mongoDB from top educator of Developers' point, MongoDB no need to care about datatypes.

[More Detail](#)



### Compute Basic with Microsoft products

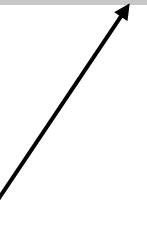
Learn the essential of todays world's starting with compute and Microsoft basics

[More Detail](#)

© 2020 Copyright: Developers Point



BYU



**Courses Page**  
Anyone can come and search for courses

 Developers' Point  Courses  Contact

+ Register

 login

Fill form no login required

Name\*

Your name

Email-id\*

Email id to contact you

Reachable number

Contact number

Type of query

A heading for your query

Query\*in details

Explain properly in details we will reach as soon as possible to solve

Submit

### Some other ways to contact us

Number: +91-9876-5432-89, +91-8375-9905-00

Mail-name: support@developerspoint.com

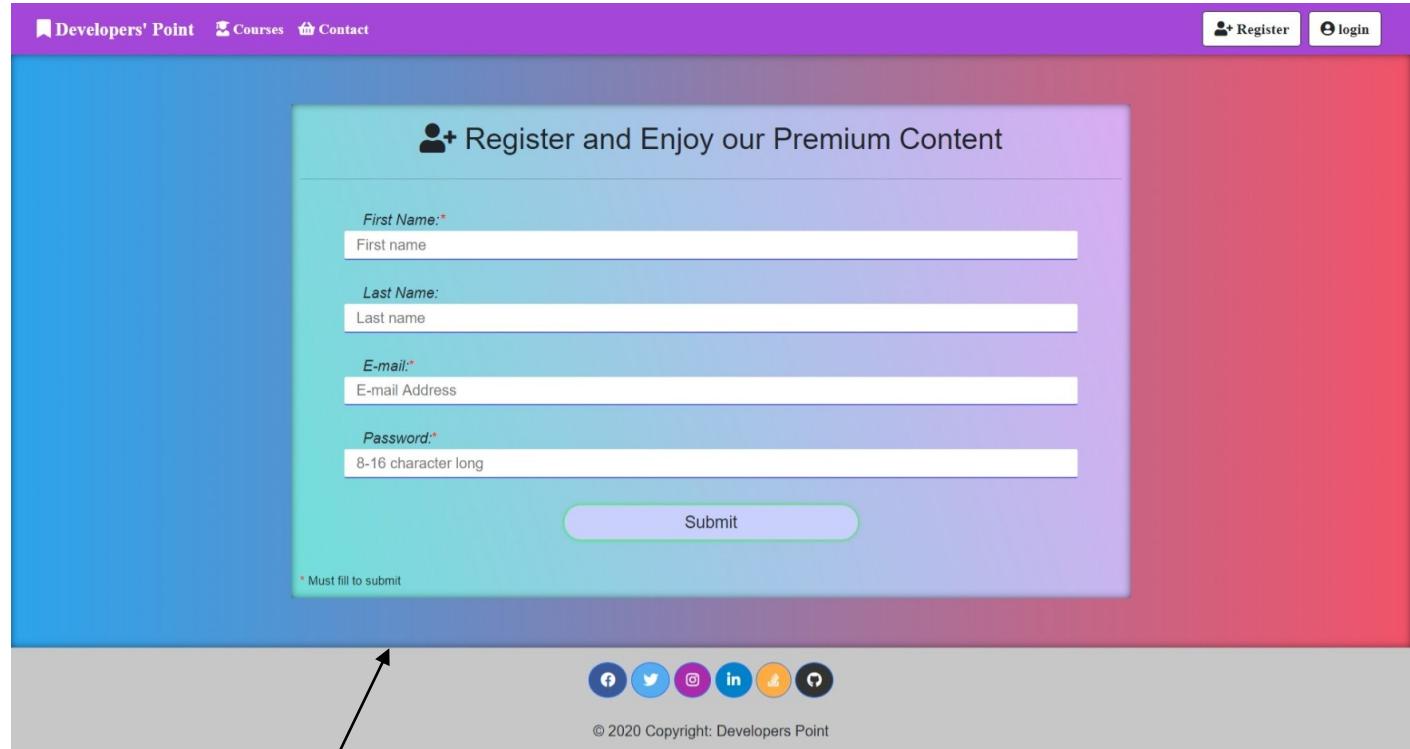
Visit: 36, Ground floor, near bank of baroda, Sankat Vihar, New Delhi, Delhi 110067



© 2020 Copyright: Developers Point

**Contact Page**  
Anyone can come  
and contact  
institute on any  
query

BYUW

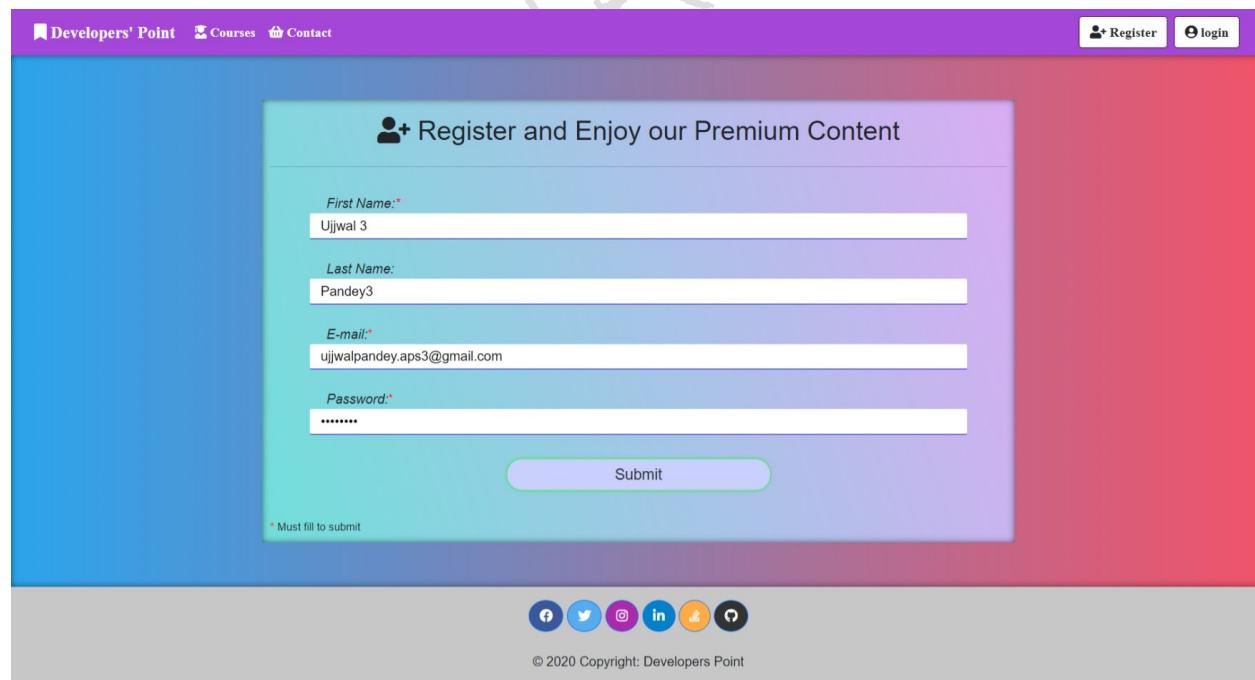


### Registration Page/Form

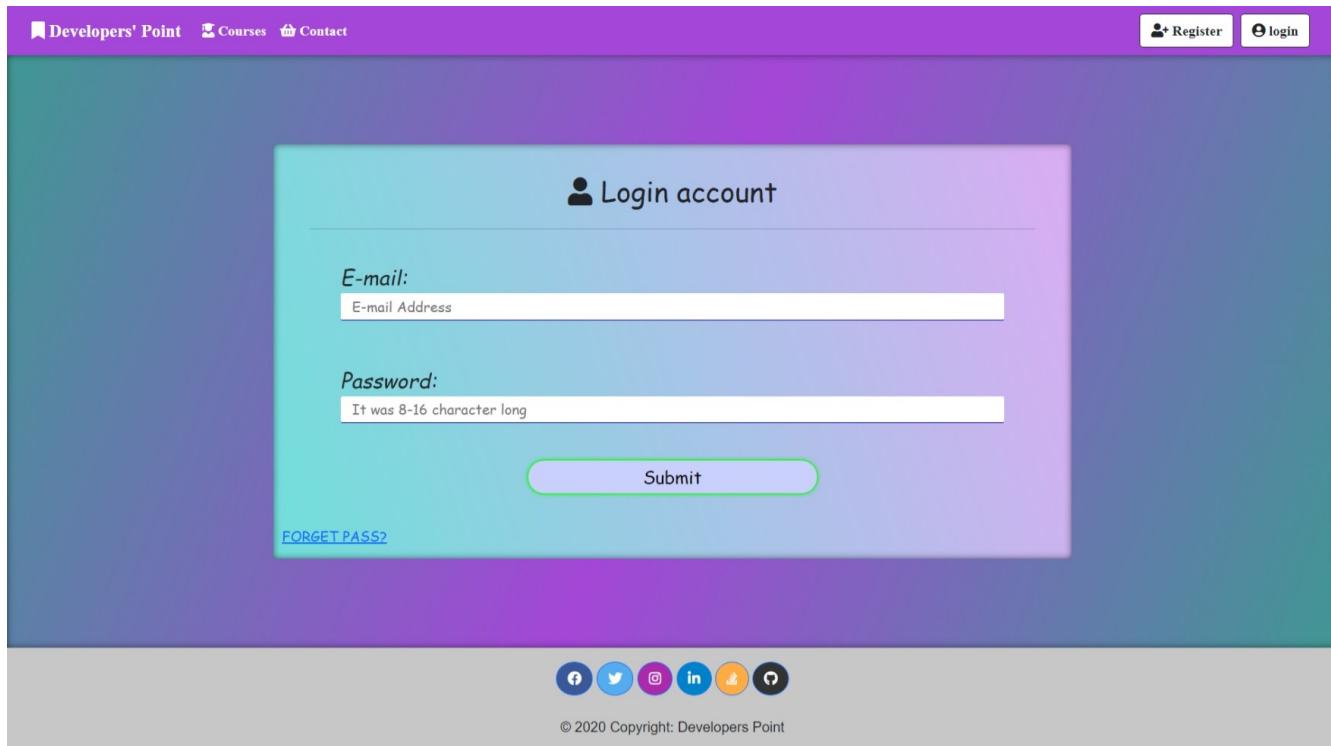
To access main features a person should register once

### Registration Page/Form demo

Registering myself  
On registration user gets a welcome mail, with initial details

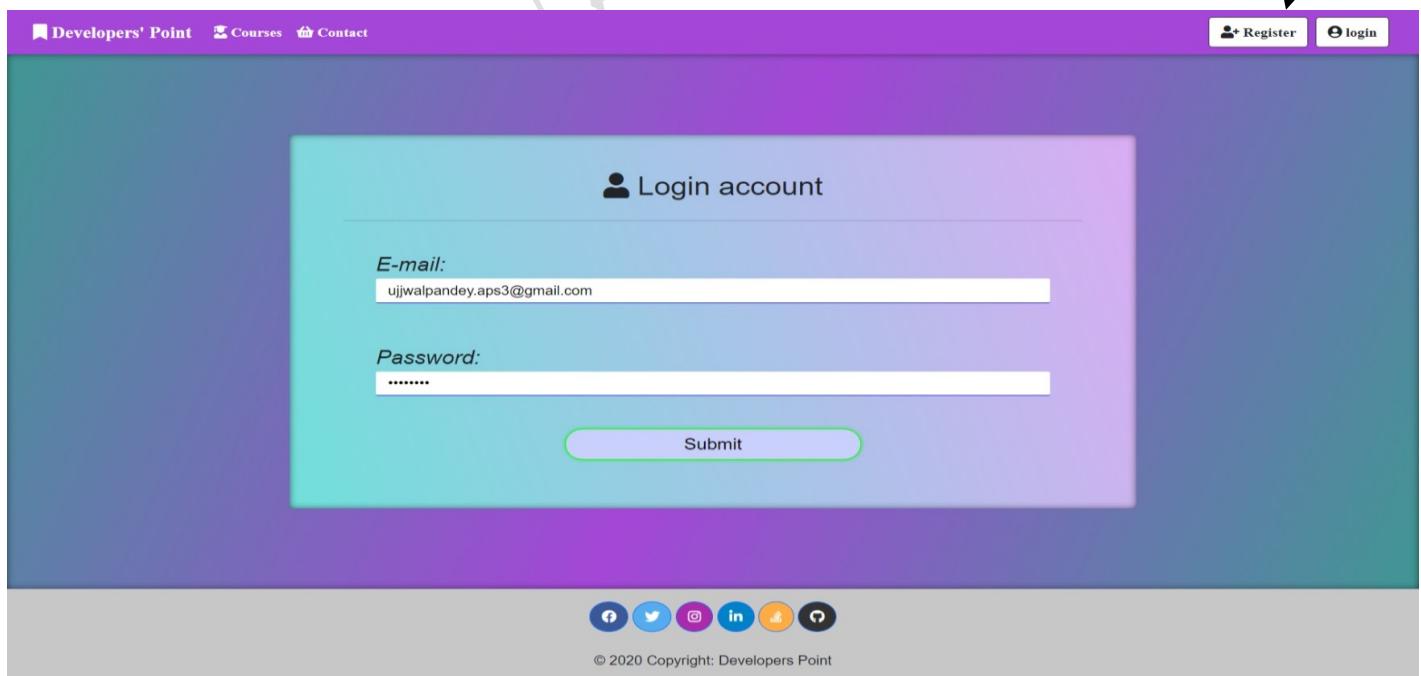


On successful registration user gets a mail.



**Login Page/Form**  
To access main features a person should login.  
Everyone (Student, faculty and admin)

**Login Page/Form demo**  
Logging as a student



The screenshot shows the Institute Management System's home page for a student. At the top, there is a purple header bar with navigation links: 'Developers' Point', 'Home' (which is highlighted with a red box), 'Courses', and 'Contact'. On the right side of the header are 'Ujjwal 3 Pandey3' and 'logout' buttons. The main content area has a white background. In the center, there is a message 'No course bought yet' above a blue search bar labeled 'Search Courses'. To the right, there is a 'Profile details' section containing user information: Name: Ujjwal 3 Pandey3, Email id: ujjwalpandey.aps3@gmail.com, Contact number: null, User Type: student, User since: 24-04-2021, 20:20:15, Address: null, null, null null, and an 'Edit' button. Below the profile section, there is a grey footer bar with social media icons (Facebook, Twitter, Instagram, LinkedIn, YouTube, GitHub) and the text '© 2020 Copyright: Developers Point'. A large watermark 'BY UJJWAL' is visible across the bottom of the page.

This will appear only when user don't have a single course bought.  
On click refer to the Courses Page

'Home Page' of 'Student'  
Logged in as student  
When they haven't bought any course

When logged in this button appears

Developer's Point Home Courses Contact

Ujjwal Pandey Ji logout

**Introduction To JavaScript**



Learn JavaScript Beginner to Pro in less than 16hrs.

Here we will learn all the important basic of javascript and go upto the pro level to learn the concepts in depth.

[More Detail](#)

**mongoDB**



Database MongoDB

Learn mongoDB from top educator of Developers' point, MongoDB no need to care about datatypes.

[More Detail](#)

**Profile details**

Name: Ujjwal Pandey Ji  
Email id: ujjwalpandey.aps@gmail.com  
Contact number: 9876543210  
User Type: student  
User since: 14-04-2021, 18:21:59  
Address:- m-54, monday market, Delhi, New Delhi 110060

[Edit](#)

**MySQL**



Learn MySQL Beginner to Pro

We will learn all the concepts of MySQL and work on 3 projects. update

[More Detail](#)

© 2020 Copyright: Developers Point

The page after the user (student/faculty) has bought at-least one course

BY UJJWAL

The screenshot shows a course details page. At the top, there are navigation links: 'Developers Point', 'Home', 'Courses', 'Contact', 'Ujjwal 3 Pandey3', and 'logout'. Below the navigation is a large image of two computer monitors displaying data visualizations. The main content area has a light purple background. It starts with a section titled 'Compute Basic with Microsoft products' with a description: 'In this course we Learn the essential of todays world's starting with compute and Microsoft basics'. A 'Requirements' section lists prerequisites. Below that is a price section showing 'Rs 270 - Rs 300 10% off.' At the bottom right are 'More Detail' and 'Buy Course' buttons. The footer features social media icons and a copyright notice: '© 2020 Copyright: Developers Point'. A large, stylized signature 'BB' is visible on the left side of the page.

**Topic:** Compute Basic with Microsoft products

**Launch Date:** 2021-05-14

**Last Update Date:** 2021-05-14 16:26:57.0

**Description:** In this course we Learn the essential of todays world's starting with compute and Microsoft basics

**Requirements:**

- Before starting this course student should know switch on computer/laptop
- Specially created for 'Beginner' level individuals
- Providing all the necessary content in just '45' hrs
- By our Ujjwal Pandey ,specialist in Microsoft's products like MS Office, Word, Excel, etc.
- On demand in Hindi
- 15-Day Money-Back Guarantee

**Price:** Rs 270 - ~~Rs 300~~ 10% off.

**More Detail**

**Buy Course**

**Course Details**

This page appears when a user clicks on view more details button of a courses card from all courses page

This button will take you to contact form. From where you can ask for any required detail and institute management team will refer you back

With this button you can buy the course by redirecting to payment gateway and pay they're with different means

**Developers' Point** **Home** **Courses** **Contact** **Ujjwal 3 Pandey3** **logout**

Buying Microsoft's products like MS Office, Word, Excel, etc. Course

**₹ 270**

This is the best course for the people who are at Beginner level

Price: ₹ 300

Discount: 10%

₹-30

Total price: ₹270

**Credit card** **Debit card** **Pay pal** **UPI** **Net Banking**

**Payment Gateway**  
This page is designed to show you the details a buyer wants to see without any interference.  
This page have-five means of payment

**Payment Ways**  
All the payments proceeding are done in the same page, they opens' separate modals showing area to fill details

The screenshot shows a user profile page with a modal overlay titled "Profile Details". The modal contains a section titled "DETAILS" with three fields: "Email ID: ujjwalpandey.aps3@gmail.com", "Name: Ujjwal 3", and "User since: 24-04-2021, 20:20:15". Below these fields are two buttons: "Close" and "Edit". A yellow arrow points from the top-right corner of the main profile page towards the "Edit" button in the modal. A grey watermark "BY UJJWAL" is visible across the bottom left.

**Profile Details**

**DETAILS**

Email ID: ujjwalpandey.aps3@gmail.com

Name: Ujjwal 3

User since: 24-04-2021, 20:20:15

Close

Edit

**Profile details**

Name: Ujjwal 3 Pandey3

Email id: ujjwalpandey.aps3@gmail.com

Contact number: null

User Type: student

User since: 24-04-2021, 20:20:15

Address:- null, null, null, null null

With this button on top-right corner,  
On click it will open a model showing basic  
detail of a user and giving a button to  
update the profile details  
Like the detail which is showing null fill it  
with real data

Developer's Point Home Courses Contact

Ujjwal Pandey logout

## Update details with precautions

add details in value and work in servlet to get them all and update in database. then work on courses page. then work on student page.

First Name:<sup>\*</sup>

Ujjwal

Last Name:

Pandey ji

Password:<sup>\*</sup>

\*\*\*\*\*

Type of user:- *(this user was student)\**

Student

Contact no:

9876543210

House-no:

54

Landmark:

monday market

City:

delhi

State:

new Delhi

Pin Code:

110060

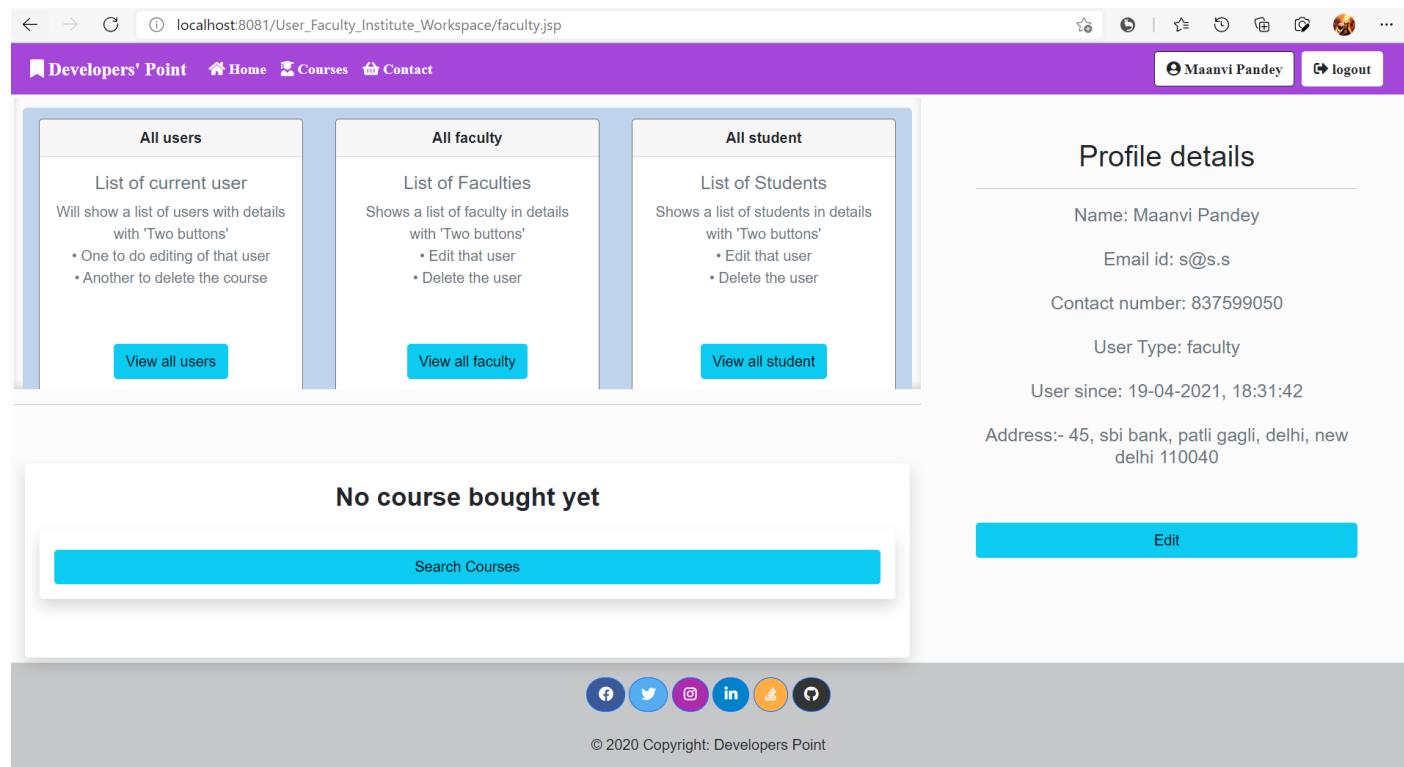
Update



© 2020 Copyright: Developers Point

BB

**Updating profile page  
(Dynamic page)**  
User can update its own profile here  
It is an example too



The screenshot shows the Faculty Home page with a purple header bar. The header includes links for 'Developers' Point', 'Home', 'Courses', 'Contact', and a sign-in box for 'Maanvi Pandey' with a 'logout' button. Below the header are three cards: 'All users' (List of current user), 'All faculty' (List of Faculties), and 'All student' (List of Students). Each card has a 'View all [category]' button. A central message box says 'No course bought yet' with a 'Search Courses' input field and an 'Edit' button. At the bottom, there's a footer with social media icons and the text '© 2020 Copyright: Developers Point'. A large, faint, handwritten-style watermark 'BYUWA' is visible across the page.

**Faculty Home page**  
Logged in as Maanvi pandey

It is same as student page with extra scrollable frame of view others details.

The diagram illustrates the Institute Management System interface, showing various components and their interactions:

- User Management:** Three cards labeled "All users", "All faculty", and "All student". Each card has a "View all [category]" button.
- Courses:** Three course cards:
  - Introduction To JavaScript:** Describes learning Microsoft products. Includes a "More Detail" button.
  - MySQL for analysis:** Describes learning MySQL for Data Analysis. Includes a "More Detail" button.
  - Learn Python from basics:** Describes learning Python basics. Includes a "More Detail" button.
- Profile Details:** A card showing profile information for a user named "hi sii nuu".
  - Name: hi sii nuu
  - Email id: facman4@gmail.com
  - Contact number: 9999999999
  - User Type: faculty
  - User since: 22-03-2021, 20:16:15
  - Address: - 3, bus stand, cp12, New Delhi 891231A blue "Edit" button is located at the bottom right of this card.
- Footer:** A grey bar with social media icons (Facebook, Twitter, Instagram, LinkedIn, YouTube, GitHub) and the text "© 2020 Copyright: Developers Point".
- Annotations:**
  - A large yellow curved arrow points from the "All users" card to the "MySQL for analysis" card.
  - A black arrow points from the "Learn Python from basics" card to the "MySQL for analysis" card.
  - An orange arrow points from the "MySQL for analysis" card to the "Edit" button.
  - A black arrow points from the "Faculty Home page after courses has been bought" box to the "Learn Python from basics" card.
  - An orange arrow points from the "List of bought courses" box to the "Edit" button.
- Text Boxes:**
  - Faculty Home page after courses has been bought
  - List of bought courses

Developers' Point Home Courses Contact Admin logout

## Courses

**View List of All courses**

A list of all courses available  
Will show a list of courses with details and 'Two buttons'

- One to do editing to the course
- Another to delete the course

[View all courses](#)

**Add new course**

Add more courses  
Contains a form with input fields:-

- Course title
- Price of course
- Description
- For which subject it is for

[Add new course](#)

## User

**All users**

List of current user  
Will show a list of users with details with 'Two buttons'

- One to do editing of that user
- Another to delete the user

[View all users](#)

**All faculty**

List of Faculties  
Shows a list of faculty in details with 'Two buttons'

- Edit that user
- Delete the user

[View all faculty](#)

**All student**

List of Students  
Shows a list of students in details with 'Two buttons'

- Edit that user
- Delete the user

[View all student](#)

**Add User**

Add a User  
To add a new user fill:

- First and Last Name of user
- Email, Password of user
- Type of user

[Add new user](#)

© 2020 Copyright: Developers Point

An admin can do all the things specified in synopsis

**Admin Home page**  
Logged in as Admin

Completely different than students and faculties

# Course

Course with id: 11 updated successfully.

ID	Title	Description	Added on	Price	Edit	Delete
4	Learn Python from basics	This course will introduce to the power of python and do clear your all doubts.	2021-04-16	7899	<a href="#">Update</a>	<a href="#">Delete</a>
9	lear MySQL Beginner to Pro	We will learn all the concepts of MySQL and work on 3 projects. update	2021-04-16	3999	<a href="#">Update</a>	<a href="#">Delete</a>
10	Learn JavaScript Beginner to Pro in less than 9hrs.	Here we will learn all the important basic of javascript and go upto the pro level to learn the concepts in deep.	2021-04-18	5999	<a href="#">Update</a>	<a href="#">Delete</a>
11	MySQL for analysis	We will learn how to use MySQL to do 'Data Analysis'	2021-04-18	3669	<a href="#">Update</a>	<a href="#">Delete</a>
12	Learn HTML-5 with Developers point.	You will learn html5 basic and get to know best practice tools, with some other cool sources to learn anytime for free.	2021-04-18	790	<a href="#">Update</a>	<a href="#">Delete</a>
13	Database MongoDB	Learn mongoDB from top educator of Developers' point, MongoDB no need to care about datatypes.	2021-04-19	2998	<a href="#">Update</a>	<a href="#">Delete</a>



© 2020 Copyright: Developers Point

## View all Courses Page

Admin can View the list of all the courses.  
Admin can 'delete' and 'update' them also.

Blue line under word course will appear when admin has updated any course Disappear after one time show

## Add new Course with respective details

*Course Title:*\*

Title to show on course

*Course Description:*\*

Little explanation oF learning material

*Price (in rs):*\*

Course price without any discount

*Subject it is related to:*\*

Select Course Subject

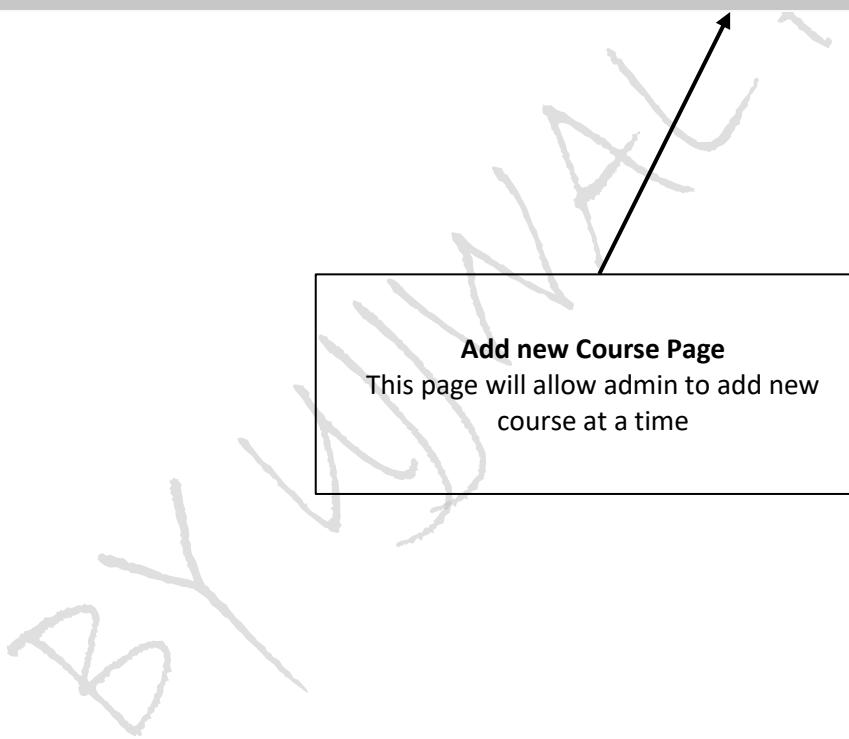
Submit



© 2020 Copyright: Developers Point

### Add new Course Page

This page will allow admin to add new course at a time



Developer's Point Home Courses Contact Admin logout

### Update Course

*Course Title:*\*  
MySQL for analysis

*Course Description:*\*  
We will learn how to use MySQL to do 'Data Analysis'

*Subject it is related to:*\*  
MySQL

*Price (in rs):*\*  
3669

*Discount:*\*  
18

*Which level of user for:*\*  
Intermediate

*Total time this will course take to cover:*\*  
12

*Creator name:*\*  
Sonam

*What the student must know before this course:*\*  
how to use computer

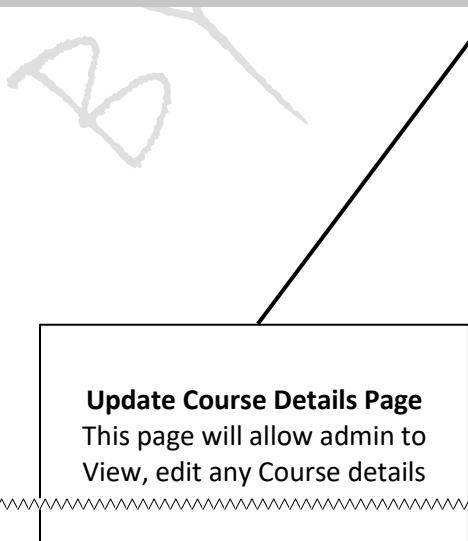
*Days of money back guarantee:*\*  
13

*Course teaching language:*\*  
MySQL

*Which spoken language used to teach by teacher:*\*  
English

**Submit**

© 2020 Copyright: Developers Point



**Developers' Point** [Home](#) [Courses](#) [Contact](#) [Admin](#) [logout](#)

# Users

Name	Email	User Type	Contact	Address	With us from	Edit	Delete
Admin	admin@gmail.com	admin	0	Address:- null, null, null, null null	15-04-2021, 18:32:53	<a href="#">Update</a>	<a href="#">Delete</a>
faculty man	faculty@g.com	faculty	997789863	Address:- any, any, delhi, rasg 55	18-04-2021, 23:46:19	<a href="#">Update</a>	<a href="#">Delete</a>
Maanvi Pandey	maanvi@mail.com	student	04545454	Address:- ramadiv, fasdf, delhi, new delhi 45	19-04-2021, 19:50:50	<a href="#">Update</a>	<a href="#">Delete</a>
Rahul Pal	rahulpal@gmail.com	student	1119	Address:- 45, neem chwak, delhi, new delhi 110060	20-04-2021, 16:53:42	<a href="#">Update</a>	<a href="#">Delete</a>
Maanvi Pandey	s@s.s	faculty	837599050	Address:- 45, sbi bank, patli gagli, delhi, new delhi 110040	19-04-2021, 18:31:42	<a href="#">Update</a>	<a href="#">Delete</a>
Ujjwal Pandey ji	ujjwalpandey.aps@gmail.com	student	9876543210	Address:- 54, monday market, delhi, new Delhi 110060	14-04-2021, 18:21:59	<a href="#">Update</a>	<a href="#">Delete</a>
Ujjwal 3 Pandey3	ujjwalpandey.aps3@gmail.com	student	null	Address:- null, null, null, null null	24-04-2021, 20:20:15	<a href="#">Update</a>	<a href="#">Delete</a>

© 2020 Copyright: Developers Point

All type of users

**View all Users Page**  
This page will allow admin to View, edit and delete any user dynamically

**Users**

Name	Email	User Type	Contact	Address	With us from	Edit	Delete
faculty man	faculty@g.com	faculty	997789863	Address:- any, any, delhi, rasg 55	18-04-2021, 23:46:19	<a href="#">Update</a>	<a href="#">Delete</a>
Maanvi Pandey	s@s.s	faculty	837599050	Address:- 45, sbi bank, patli gagli, delhi, new delhi 110040	19-04-2021, 18:31:42	<a href="#">Update</a>	<a href="#">Delete</a>

© 2020 Copyright: Developers Point

Name	Email	User Type	Contact	Address	With us from	Edit	Delete
Maanvi Pandey	maanvi@mail.com	student	04545454	Address:- ramadiv, fasdf, delhi, new delhi 45	19-04-2021, 19:50:50	<a href="#">Update</a>	<a href="#">Delete</a>
Rahul Pal	rahulpal@gmail.com	student	1119	Address:- 45, neem chwak, delhi, new delhi 110060	20-04-2021, 16:53:42	<a href="#">Update</a>	<a href="#">Delete</a>
Ujjwal Pandey ji	ujjalipandey.aps@gmail.com	student	9876543210	Address:- 54, monday market, delhi, new Delhi 110060	14-04-2021, 18:21:59	<a href="#">Update</a>	<a href="#">Delete</a>
Ujjwal 3 Pandey3	ujjalipandey.aps3@gmail.com	student	null	Address:- null, null, null, null	24-04-2021, 20:20:15	<a href="#">Update</a>	<a href="#">Delete</a>

Only faculty

Only students

[Developers' Point](#) [Home](#) [Courses](#) [Contact](#) [Admin](#) [Logout](#)

### Add new user with respective details

*First Name:*\*  
First name

*Last Name:*  
Last name

*E-mail:*\*  
E-mail Address

*Password:*\*  
8-16 character long

*Type of user:*\*  
Select user type

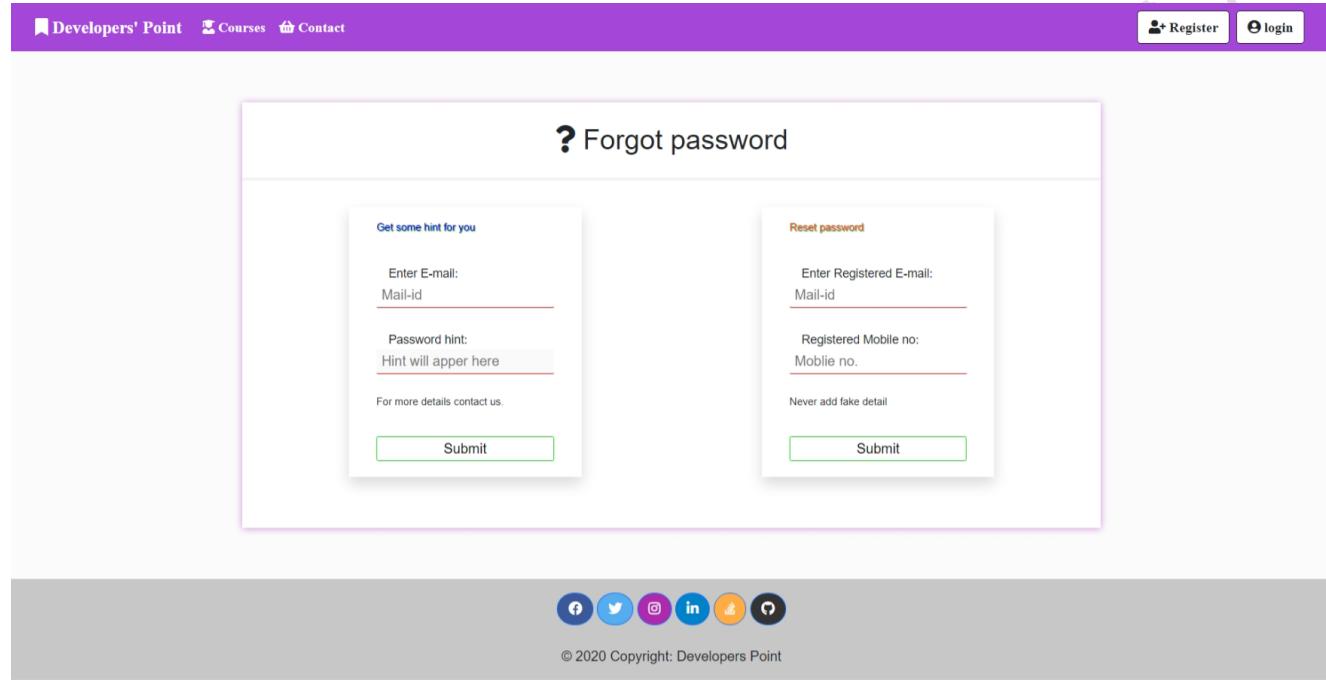
[Submit](#)



© 2020 Copyright: Developers Point

#### Add new User (by Admin) Page

This page will allow admin add new user and assign their role of student, admin or faculty  
194



The screenshot shows a purple header bar with navigation links: 'Developers' Point', 'Courses', 'Contact', 'Register', and 'login'. Below the header is a white form titled 'Forgot password'. The form is divided into two sections: 'Get some hint for you' on the left and 'Reset password' on the right. The 'Get some hint for you' section contains fields for 'Enter E-mail:' and 'Mail-id', and a note 'Hint will appear here'. The 'Reset password' section contains fields for 'Enter Registered E-mail:' and 'Mail-id', and 'Registered Mobile no:' and 'Mobile no.'. Both sections have a note 'Never add fake detail' and a green 'Submit' button.



**Forget Password page**  
This page has 2 functionalities,  
1. Get some hint about password,  
2. Receive a mail with the details

© 2020 Copyright: Developers Point

**In Footer**

In footer (available on each and every page) all these icons visible are not only icons they are click and referred to actual my profiles

## 8. Testing

### Unit Testing

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its action as required.

### Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the Server module and Client module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

### System Testing:

After completion of integration testing. I used system testing to test my system. It is the testing of the system its initial objectives.

### Alpha Testing:

The alpha testing is conduct at the developer sitein the presence of the customer. The software is used in a natural setting by all the errors and me and usage problems are recorded for maintenance. This test is conduct in a controlled environment to uncover errors.

### Beta Testing:

The beta test is conduct br the end user of the software. Thereafter the beta test is a live application of the software in an environment that cannot be controlled by the developer. They report all the problems to me. As a result of problems reported during beta test, I made modification leading to completion of my software project.

### Debugging:

After successful coding successful coding the individual modules, all the modules were crosschecked. The debugging is performed to ensure that no software bugs existed. Debugging is recursively performed until the newly designed system resembles the proposed system. It was ensured that all the modules were logically and syntactically correctly coded.

Debugging occurs as a sequence of successful testing. That is, when a test cases uncovers error, Debugging can add should be orderly process; it is still much an art.

The debugging process will always have one of the two outcomes:

1. The cause will be found, corrected, removed, or
2. The cause will be found.

### THE STEPS IN THE SOFTWARE TESTING:

- The steps involved during Unit testing are as follows:
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Complete code review of the module.
- Actual testing done manually.
- Modifications done for the errors found during testing.
- Prepared the test result scripts.

### THE UNIT TESTING DONE INCLUDED THE TESTING OF THE FOLLOWING ITEMS:

- Functionality of the entire module/forms.
- Validations for user input.
- Checking of the Coding standards to be maintained during coding.
- Testing the module with all the possible test data.
- Testing of the functionality involving all type of calculations etc.
- Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, we integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

- The steps involved during System testing are as follows:
- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.

The System Testing done included the testing of the following items:

- Functionality of the entire system as a whole.
- User Interface of the system.
- Testing the dependent modules together with all the possible test data scripts.
- Verification and Validation testing.
- Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

## 9. Input and Output

### Input Design for

The system input design is divided in to 4 sections.

1. The Administration section,
2. The Faculty section,
3. The Student section, and
4. The visitor section.

### Administrator

Input design is an important part of development process since inaccurate input data are the most common cause of errors in data processing. Erroneous entries can be controlled by input design. It consists of developing specifications and procedures for entering data into a system and must be in simple format. The goal of input data design is to make data entry as easy, logical and free from errors as possible. In input data design, we design the source document that capture the data and then select the media used to enter them into the computer.

There are three major approaches for entering data in to the computer.

They are

- Menus,
- Dialog Boxes, and
- Form.

### Output Design

Designing computer output should proceed in an organized, well throughout manner; the right output element is designed so that people will find the system whether or executed. When we design an output, we must identify the specific output that is needed to meet the system. The usefulness of the new system is evaluated on the basis of their output.

**Existing System of User-faculty workspace:** In the existing system the admission, retrieval, course buying, updating data and other works are done only manually but in proposed system we have to computerize the processes using this application.

- Lack of security of data.
  - Reduce the more man power.
  - Reduce the time consumed in process.
  - Reduce the consumption of large volume of pare work.
  - Remove manual calculations.
  - No direct role of the management team in small works.
- 

## **10. Implementation of the security for the software development**

System security mechanism, software security has become increasingly important in the age of hackers and virus. This attribute measures a system's ability to withstand attack on the security. Attack can be made on all three components of the software: Programs, Data and documents.

### **Creating of user profiles & Access Rights:**

This system administration can create the user and delete user. The admin can change the setting user based the requirement of the system. An authorized user cannot log in on the system. This checked during login of every user, before opening a page the web servers check the type of user before referring the user to that page, only the authorized and permitted type of user can access the page not any one, not all type of users to open all type of pages if tried he/she will be redirected to the login page and if they are already logged in, they will be logged out.

---

## **11. Limitations of the Project**

- Excel export has not been developed for course bought.
  - The transactions are executed in offline mode, hence online data for course buying and batch admission, and modification is not possible.
  - Offline reports of Course bought, Batch Registration, transaction receipt cannot be generated due to offline mode execution.
  - Cannot choose batch online, to do that we have to use contact form or other means.
- 

## **12. Future Application on the Project**

- We can add printer in future,
  - We can give more advance software for App including more facilities,
  - We can host online courses on the platform so that users can buy, access and learn online worldwide without coming to any course,
  - Integrate multiple load balancers to distribute the loads of the system as the web app goes worldwide,
  - Create the master and slave database structure to reduce the overload of the database queries,
  - Implement the backup mechanism for taking backup of codebase and database on regular basis on, different servers,
  - Implement own transaction system, and so on.
-

## 13. Bibliography

1. www.w3schools.com
2. tutorialpoint.com
3. DZone.com
4. en.wikipedia.org
5. Youtube.com (Several tutorial channels of YouTube)
6. www.hotscripts.com/category/java/
7. [www.apache.org/](http://www.apache.org/)
8. Java original documentations
9. www.mysql.com/click.php?e=35050
10. [bootstrap.com](http://bootstrap.com) for CSS(.CSS) and JavaScript(.jsp)
11. mdbBootstrap.com

**Thank you**

