# **B. Tech. project tracking dashboard – G 11**

PROJECT REPORT - II

*Submitted by*

| S. No | Name | Roll Number |
| --- | --- | --- |
| 1. | A SANDEEP REDDY | CB.EN.U4CSE18007 |
| 2. | B KISHORE REDDY | CB.EN.U4CSE18009 |
| 3. | J SAIRAM SAMPATH | CB.EN.U4CSE18024 |
| 4. | K KOUSIK | CB.EN.U4CSE18033 |
| 5. | HRIDHI SETHI | CB.EN.U4CSE18502 |

*is fulfilment of the requirements for the Course – 15CSE376 (Net Centric Programming)*

## BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMRITA SCHOOL OF ENGINEERING AMRITA VISHWA VIDYAPEETHAM COIMBATORE -641112**

**OCTOBER 2021**

Table of Contents

| **S.NO** | **CONTENT** | **PAGE NUMBERS** |
| --- | --- | --- |
| 1 | JDBC | 3 |
| 2 | SERVLET | 3 |
| 3 | ER DIAGRAM | 5 |
| 4 | ARCHITECTURE DIAGRAM | 6 |
| 5 | TABLE SPECIFICATIONS | 7 |
| 6 | FUNCTIONALITIES USING SERVLETS | 9 |
| 7 | DATABASE CONNECTIVITY | 13 |
| 8 | VALIDATION | 16 |

# JDBC

JDBC is an application programming interface (JDBC API) that defines a set of standard operations for interacting with relational database management systems (DBMSs). The DBMSs may be located on a remote machine connected to the Internet. In order to access a database under a specific DBMS, for example, PostgreSQL, one must

have a driver for that DBMS and the driver must implement JDBC API. JDBC is a trademark name and not an acronym. JDBC or Java Database Connectivity is a specification from Sun microsystems that provides a standard abstraction (that is API or Protocol) for java applications to communicate with various databases. It provides the language with java database connectivity standard. It is used to write programs required to access databases. JDBC along with the database driver is capable of accessing databases and spreadsheets. The enterprise data stored in a relational database (RDB) can be accessed with the help of JDBC APIs. JDBC is an API (Application programming interface) which is used in java programming to interact with databases. The classes and interfaces of JDBC allows applications to send requests made by users to the specified database. Enterprise applications that are created using the JAVA EE technology need to interact with databases to store application-specific information. So, interacting with a database requires efficient database connectivity which can be achieved by using the

ODBC (Open database connectivity) driver. This driver is used with JDBC to interact or communicate with various kinds of databases such as Oracle, MS Access, MySQL and SQL server databases.

# Servlets

Java Servlets, or simply servlets are a set of Java classes that can be used and extended for Web server-side programming provided the Web server supports Java servlets. Basically, the programmer specifies which servlet is to be used to process which request or which type of requests from client. Thus, when a request is received by the Web server, the Web server finds the proper servlet for the request. For example, in an HTML form, its action clause can explicitly specify which servlet will be invoked to process the data submitted through the form. Servlets are the Java programs that run on the Java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, then send a

response back to the web server. The server-side extensions are nothing but the technologies that are used to create dynamic Web pages. Actually, to provide the facility of dynamic Web pages, Web pages need a container or Web server. To meet this requirement, independent Web server providers offer some proprietary solutions in the form of APIs(Application Programming Interface).

These APIs allow us to build programs that can run with a Web server.

In this case, Java Servlet is also one of the component APIs of Java Platform Enterprise Edition which sets standards for creating dynamic Web applications in Java. Servlets are built from two packages:

1. javax.servlet(Basic)
2. javax.servlet.http(Advanced)

An attribute in a servlet is an object that can be set, get or removed from one of the following scopes (i.e.) request scope, session scope, application scope.The servlet programmer can pass information from one servlet to another using attributes. It is just like passing objects from one class to another so that we can reuse the same object again and again.

The following 4 are attribute specific methods. They are as follows:

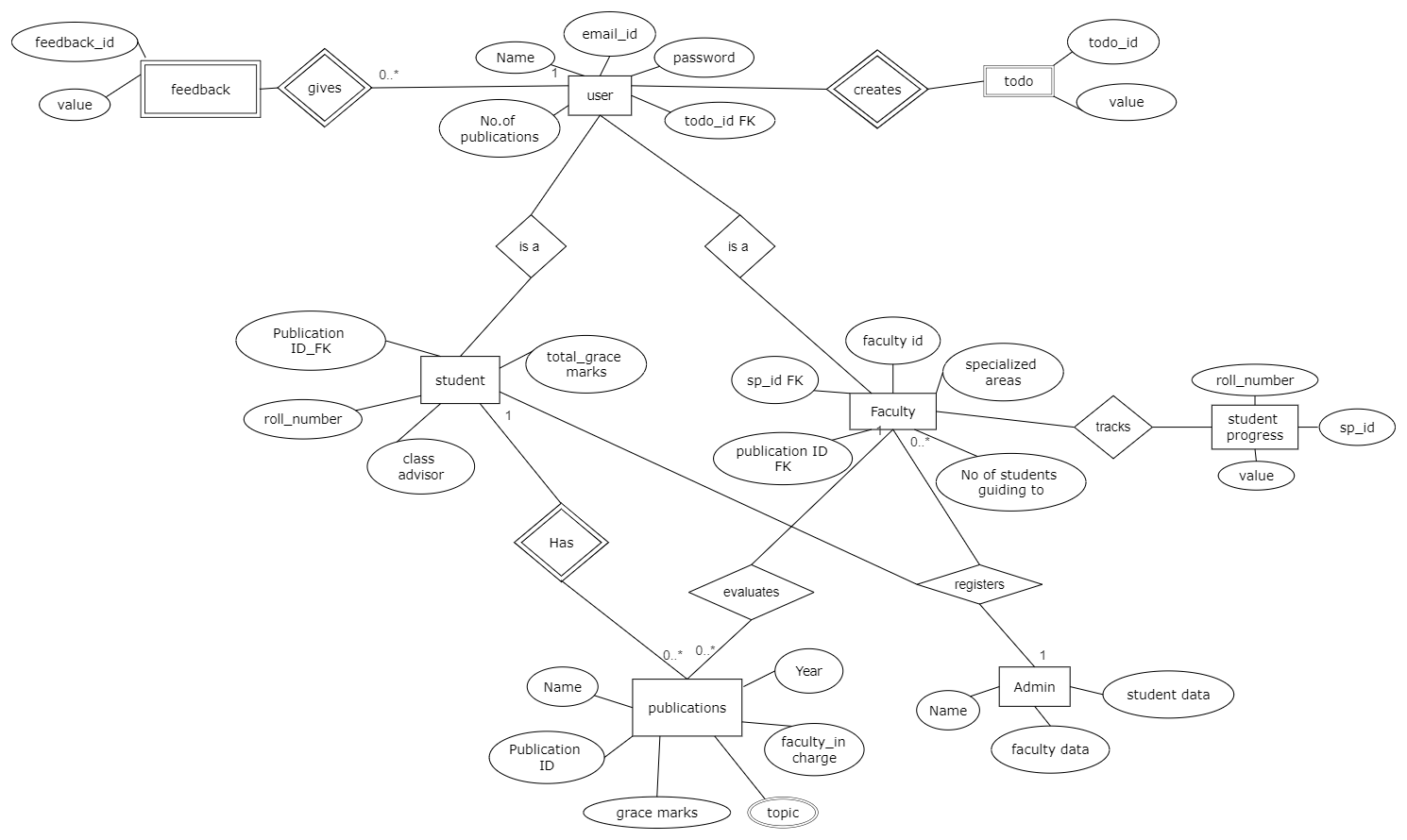
* public void setAttribute (String name, Object object): sets the given object in the application scope.
* public Object getAttribute (String name): Returns the attribute for the specified name.
* public Enumeration getInitParameterNames(): Returns the names of the context's initialization parameters as an Enumeration of String objects.
* public void removeAttribute (String name): Removes the attribute with the given name from the servlet context.

The web container maintains the life cycle of a servlet instance. Let's see the life cycle of the servlet:

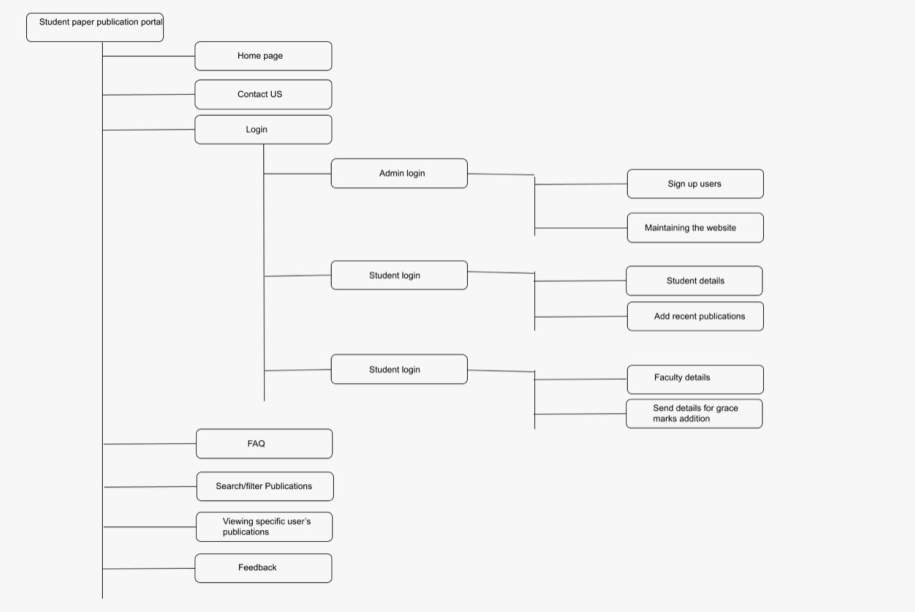
* Servlet class is loaded.
* Servlet instance is created.
* Init method is invoked.
* Service method is invoked.
* Destroy method is invoked.

There are 2 types of cookies in servlets namely non-persistent cookie and persistent cookie. Non-persistent cookies are valid for a single session only. It is removed each time the user closes the browser. Persistent cookie is valid for multiple sessions. It is not removed each time the user closes the browser. It is removed only if user logouts.

# ER DIAGRAM:

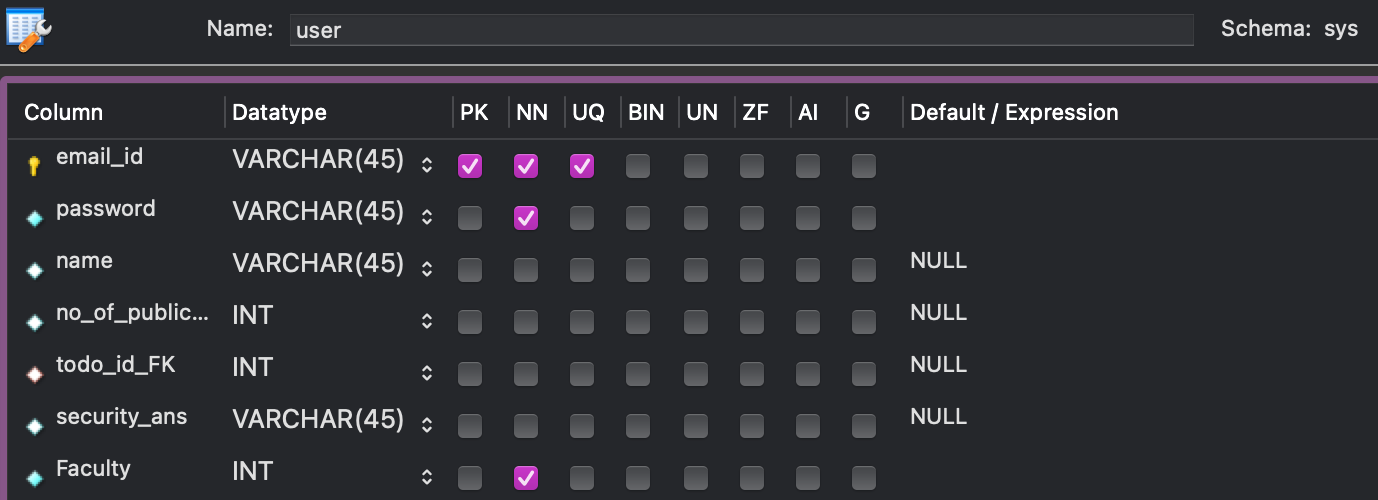
****

# ARCHITECTURE DIAGRAM:

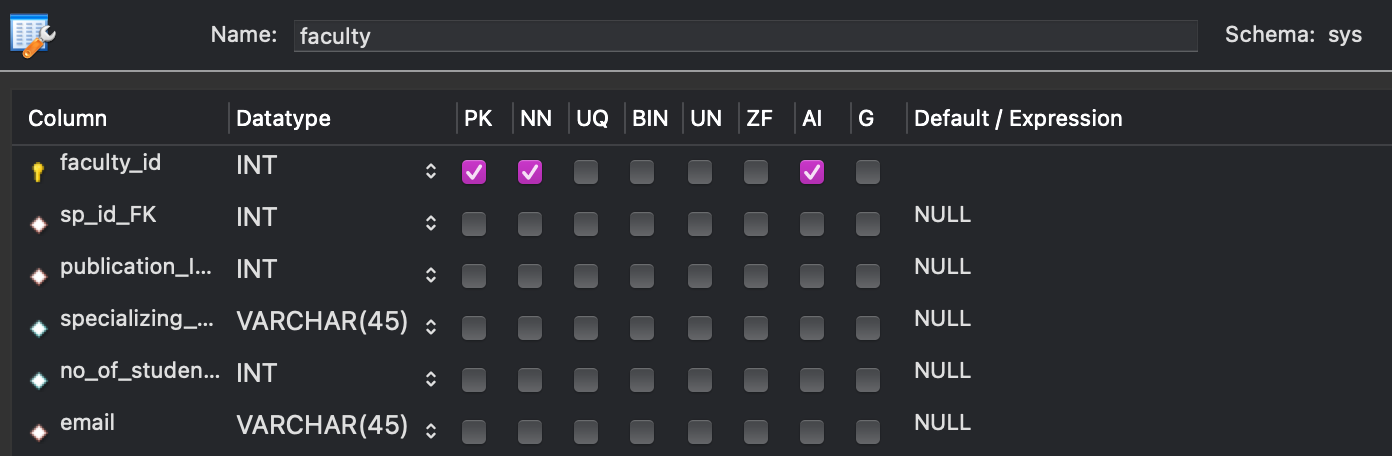
****

# Table specifications

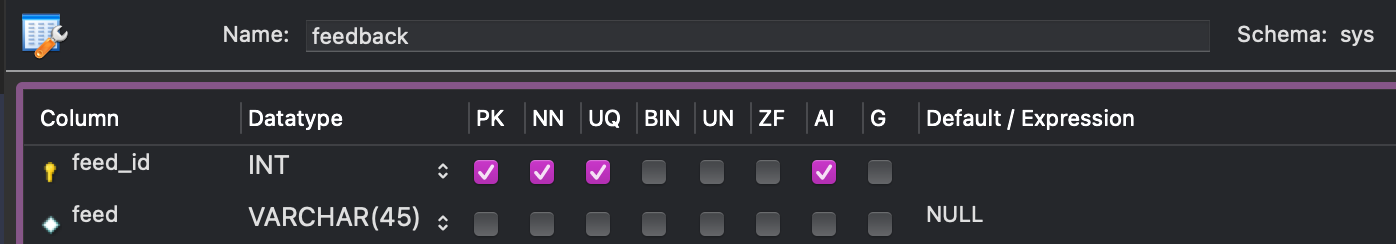
## User table



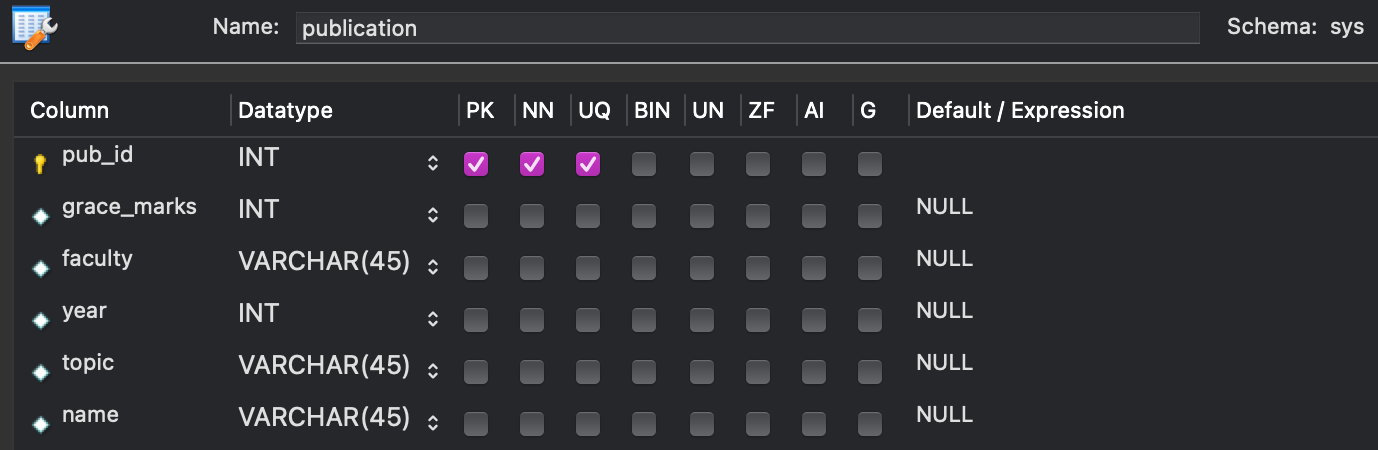
## Faculty table



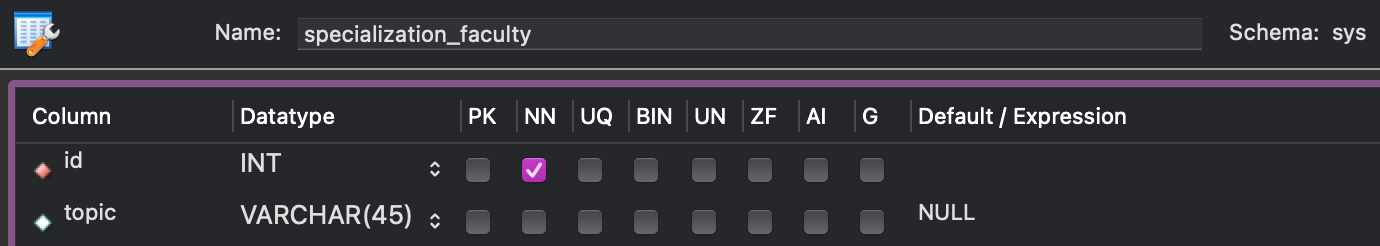
## Feedback table



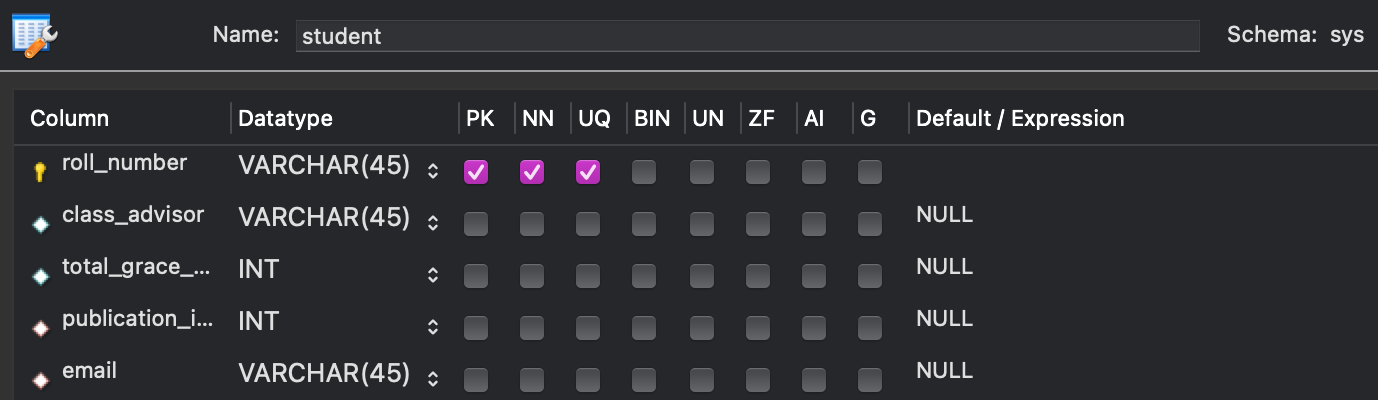
## Publications table



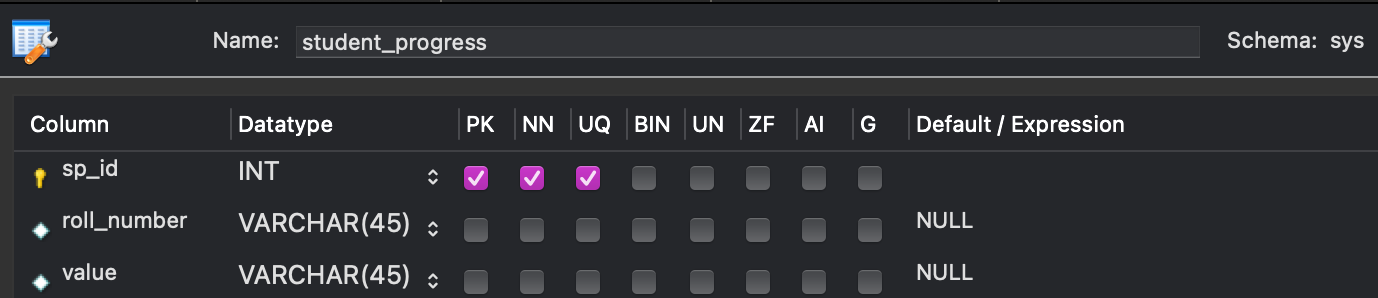
## Specializations table



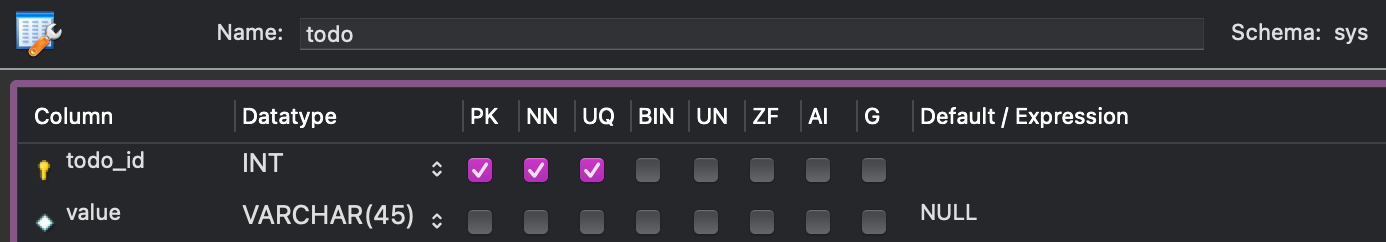
## Student table



## Student Progress table



## TO-DO list table



# Functionalities using servlets

## Faculty

| **S.No.** | **Servlet name** | **DoGet()** | **DoPost()** |
| --- | --- | --- | --- |
| 1. | Faculty signup | Renders signup page | 1. Verifies if the email id is already present or not 2. Verifies the strength of password   If verification is passed then redirects to faculty login page else throws an error and then redirects back to signup page |
| 2. | Faculty Login | Renders to login page | Verifies if the email is present in the database and whether the password matches with corresponding email. If verification is passed then redirects to the faculty dashboard, else throws the corresponding error and redirects to the login page. |
| 3. | Faculty dashboard | Renders Dashboard page | If logout then clear cookies of the user and redirects to the home page |
| 4. | Forget password | Renders forget password page | Email and security question verification is done first and then if correct redirects to a page to fill in the new password else throws an exception and redirects back.  If verification is correct then strength of password and weather password and reentered password match is validated and if true redirects to login page else throws error and redirects back. |
| 5. | Student progress | Renders student progress page | Adds the student progress details entered to the database. |
| 6. | Previous publications | Renders previous publication page | On search displays the corresponding publications. |
| 7. | Grace marks allocation | Renders grace marks allocation page | Adds the grace marks information entered to the database. |
| 8. | Faculty profile | Renders faculty profile page | If home button redirects to corresponding login page. |
| 9. | To-do list | Renders todo-list page | Adds/deletes/updates the todo entered to the database. |
| 10. | Reset password | Renders reset password page | Security question verification is done and then if the true password will be reseted by checking strength of password and similarity between password and re entered one. If verification is false then error is thrown and redirects. |

## Student

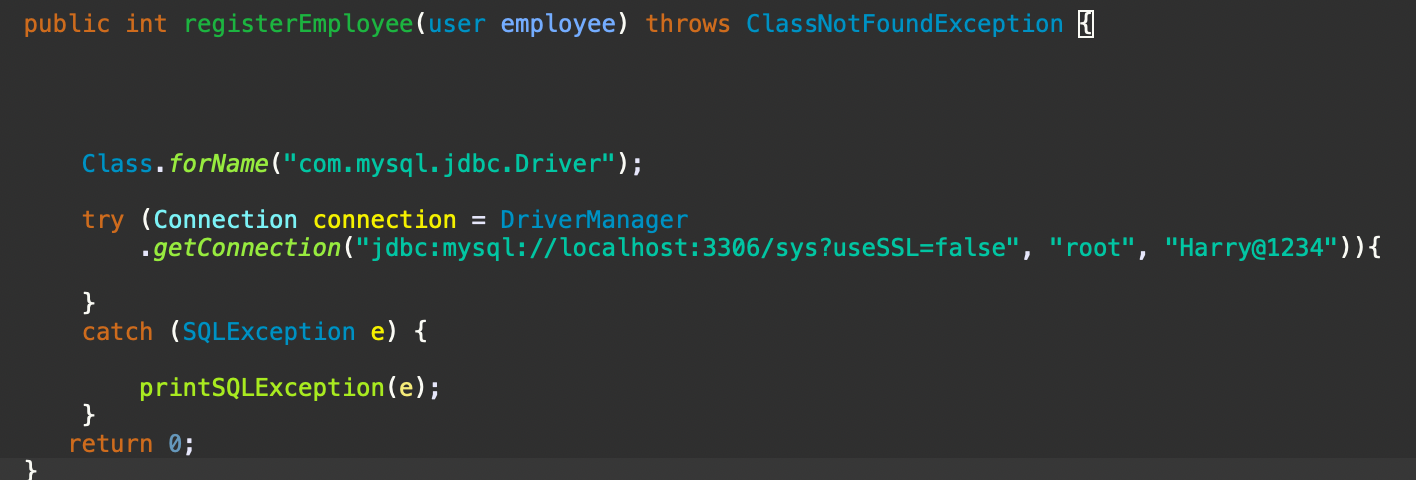
| **S.No** | **Servlet name** | **DoGet()** | **DoPost()** |
| --- | --- | --- | --- |
| 1. | Student signup | Renders signup page | 1. Verifies if the email id is already present or not 2. Verifies the strength of password   If verification is passed then redirects to student login page else throws an error and then redirects back to signup page |
| 2. | Student Login | Renders to login page | Verifies if the email is present in the database and whether the password matches with corresponding email. If verification is passed then redirects to the faculty dashboard, else throws the corresponding error and redirects to the login page. |
| 3. | Student dashboard | Renders Dashboard page | If logout then clear cookies of the user and redirects to the home page |
| 4. | Forget password | Renders forget password page | Email and security question verification is done first and then if correct redirects to a page to fill in the new password else throws an exception and redirects back.  If verification is correct then strength of password and weather password and reentered password match is validated and if true redirects to login page else throws error and redirects back. |
| 5. | Add publication | Renders add publication page | Checks the details weather all mandatory fields are entered and accordingly adds publication details to the database. |
| 6. | Previous publications | Renders previous publication page | On search displays the corresponding publications. |
| 7. | Connect with a faculty | Renders connect with a faculty page | Based on the search entry displays corresponding faculty details. |
| 8. | Student profile | Renders Student profile page | If the home button redirects to the corresponding login page. |
| 9. | To-do list | Renders todo-list page | Adds/deletes/updates the todo entered to the database. |
| 10. | Reset password | Renders reset password page | Security question verification is done and then if the true password will be reseted by checking strength of password and similarity between password and re entered one. If verification is false then error is thrown and redirects. |

## Misc

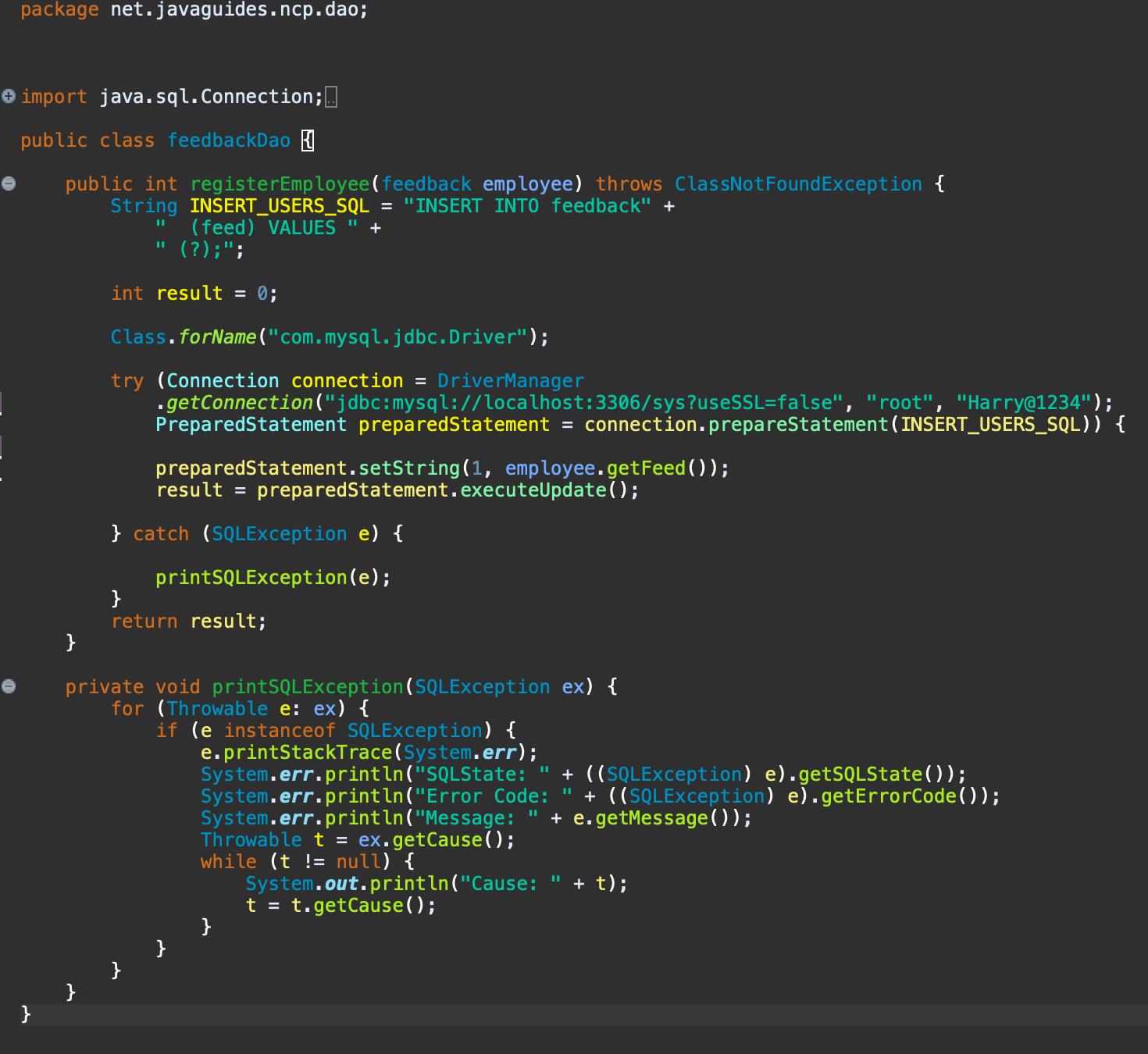
| **S.No.** | **Servlet name** | **DoGet()** | **DoPost()** |
| --- | --- | --- | --- |
| 1. | Home | Renders home page | None. |
| 2. | Filter publications | Renders filter publications page | On search entry displays the corresponding publications. |
| 3. | Feedback | Renders feedback page | Adds the feedback to the database |
| 4. | About publication | Renders about publication page | None |

# Database connectivity

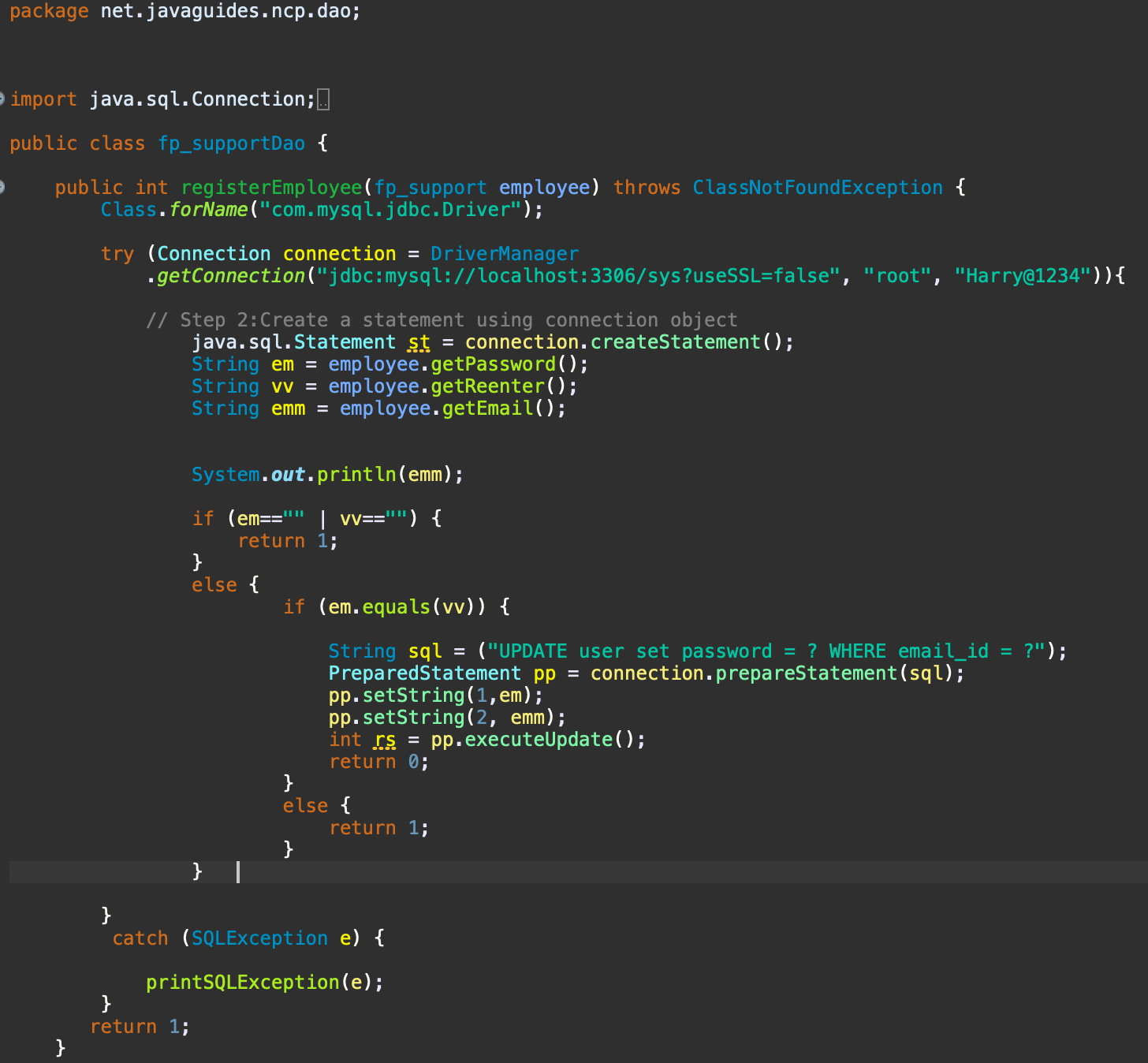
## Connectivity



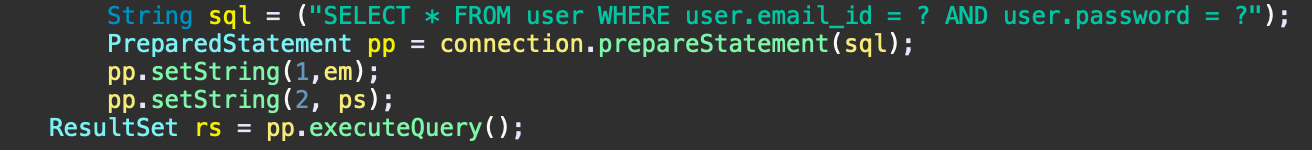
## Insertion of data



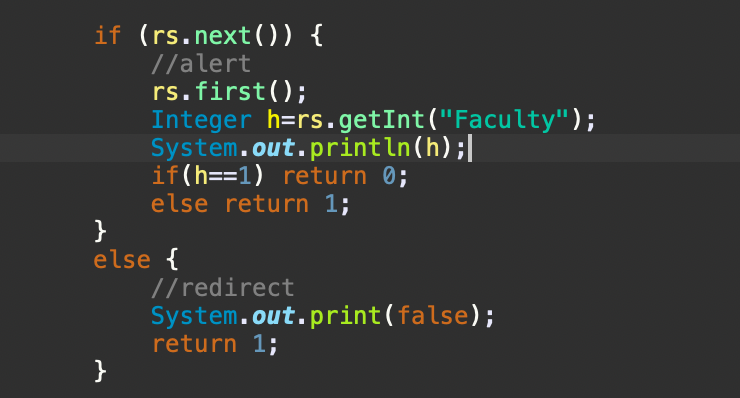
## Updation of data

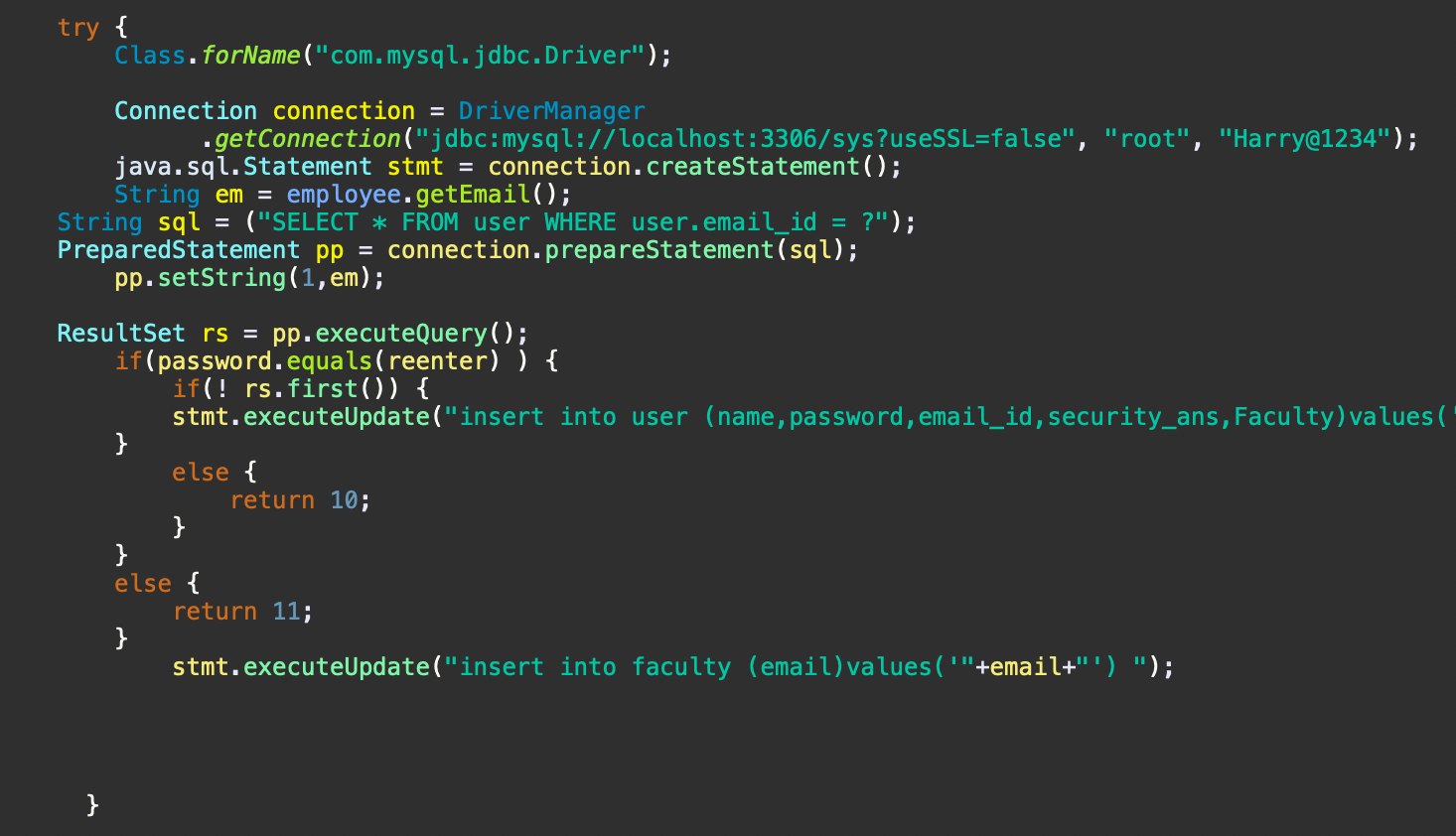


## Retrieving of data



## Validation of data





# Evaluation sheet

| Roll number | Technology | Max marks | Marks awarded | Total (30) |
| --- | --- | --- | --- | --- |
| 18007 | Servlet  JDBC | 10  10 |  |  |
| 18009 | Servlet  JDBC | 10  10 |  |  |
| 18024 | Servlet  JDBC | 10  10 |  |  |
| 18033 | Servlet  JDBC | 10  10 |  |  |
| 18502 | Servlet  JDBC | 10  10 |  |  |
|  | Project and Documentation | 10 |  |  |