

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
BELAGAVI-590018**



**A DBMS Mini-Project Report  
On**

***“Internships 101 – An Internship Agency”***

*Submitted in partial fulfillment of the requirements for the 5<sup>th</sup> semester of **Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi*

Submitted by:

**NAMRATHA J**

**1RN19CS084**

**MAHIMA N SHAH**

**1RN19CS072**

Under the Guidance of:

**Mrs.Soumya N G**  
Assistant Professor  
Dept. of CSE

**Mr.Sanjay P K**  
Assistant Professor  
Dept. of CSE



**Department of Computer Science and Engineering  
RNS Institute of Technology  
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098  
2021-2022**

**RNS Institute of Technology**  
Channasandra, Dr.Vishnuvardhan Road,  
Bengaluru-560 098

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

Certified that the DBMS mini-project work entitled “**Internships 101 – An Internship agency**” has been successfully carried out by **Namratha J** bearing USN **1RN19CS084** and **Mahima N Shah** bearing USN **1RN19CS072**, bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the **5th semester Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the mini-project requirements of the DBMS lab of 5<sup>th</sup> semester BE in CSE.

**Mrs.Soumya N G**  
Assistant Professor  
Dept. of CSE

**Mr.Sanjay P K**  
Assistant Professor  
Dept. of CSE

**Dr. Kiran P**  
Prof. and Head  
Dept. of CSE

**External Viva:**

**Name of the Examiners**

**Signature with Date**

- 1.**
- 2.**

# ACKNOWLEDGMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We are grateful to **Dr. M K Venkatesha**, Principal, RNSIT, Bangalore, for his support towards completing this mini project.

We would like to thank **Dr. Kiran P** Prof. &Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Mrs. Soumya N G**, Assistant Professor, **Mr. Sanjay P K**, Assistant Professor, Department of CSE, RNSIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of department of Computer Science & Engineering, RNSIT, Bengaluru for their constant support and encouragement.

Date :

Namratha J

1RN19CS084

Place : Bengaluru

Mahima N Shah

1RN19CS072

# **ABSTRACT**

Internships 101- An internship agency provides a platform for Internships and provide students a chance to collaborate without them having to visit various websites to find a suitable internship. Keeping that in mind, we have collected details of students that are, the basic details, marks, extracurricular activities and previous internship experience. The list of students who have entered their basic details can view it. It provides both sign up and login option.

# CONTENTS

Chapter No.	Title	Page No
	Acknowledgment	i
	Abstract	ii
1.	Introduction	
1.1	Database technologies	1
1.2	Characteristics of database approach	2
1.3	Applications of DBMS	3
1.4	Problem description/ statement	5
2.	Requirements Analysis	
2.1	Hardware Requirements	6
2.2	Software Requirements	6
2.3	Functional Requirements	6
2.3.1	Major Entities	6
2.3.2	End User Requirements	7
2.3.2.1	Netbeans IDE	7
2.3.2.2	PHP	7
2.3.2.3	MySQL	8
2.3.2.4	XAMPP Server	10
3.	Database Design	
3.1	Entities, Attributes and Relationships	11
3.2	ER Schema	12
3.3	Relational Schema	13
4.	Implementation	
4.1	Database connectivity	14
4.2	Pseudo code For Major Functionalities	16
5.	Results , snapshots and discussions	23
6.	Conclusion and Future Enhancements	28
	Bibliography	29

# CHAPTER 1

## INTRODUCTION

### 1.1 DATABASE TECHNOLOGIES

The essential feature of database technology is that it provides an internal representation (model) of the external world of interest. Examples are the representation of a particular date/time/flight/aircraft in an airline reservation or of the item code/item description/quantity on hand/reorder level/reorder quantity in a stock control system.

The technology involved is concerned primarily with maintaining the internal representation consistent with external reality; this involves the results of extensive R&D over the past 30 years in areas such as user requirements analysis, data modelling, process modelling, data integrity, concurrency, transactions, file organisation, indexing, rollback and recovery, persistent programming, object-orientation, logic programming, deductive database systems, active database systems...and in all these (and other) are as there remains much more to be done. The essential point is that database technology is a CORE TECHNOLOGY which has links to:

- Information management/processing
- Data analysis/statistics
- Data visualization/presentation
- Multimedia and hypermedia
- Office and document systems
- Business processes, workflow, CSCW(computer-supportedcooperativework)

Relational DBMS is the modern base technology for many business applications. It offers flexibility and easy-to-use tools at the expense of ultimate performance. More recently relational systems have started extending their facilities indirections like information retrieval, object-orientation and deductive/active systems which lead to the so-called 'Extended Relational Systems'.

Information Retrieval Systems began with handling library catalogues and then extended to full free-text by utilizing inverted index technology with a lexicon or thesaurus. Modern systems utilize some KBS(knowledge-basedsystems) techniques to improve the retrieval.

Object-Oriented DBMS started for engineering applications in which objects are complex, have versions and need to be treated as a complete entity. OODBMSs share many of the OOPL features such as identity, inheritance, late binding, overloading and overriding. OODBMSs have found favours in engineering and office systems but haven't been successful yet in traditional application areas.

Deductive / Active DBMS have evolved over the last 20 years and combines logic programming technology with database technology. This allows the database itself to react to the external events and also to maintain its integrity dynamically with respect to the real world.

## **1.2 CHARACTERISTICS OF DATABASE APPROACH**

Traditional form included organising the data in file format. DBMS was a new concept then, and all kinds of research were done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics—

- Real-world entity – A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses behaviour and attribute too. For example, a school database may use students as an entity and the image as an attribute.
- Relation-based tables – DBMS allows entities and relations to form tables.  
A user can understand the architecture of a database by just looking at the table names.
- Isolation of data and application – A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.
- Less redundancy – DBMS follows the rules of normalization, which splits a relation when any of its attributes has redundancy in its values. Normalization is a mathematically rich and scientific process that will reduce the data redundancy.
- Consistency – Consistency is a state where every relation in a database remains consistent. There exists methods and techniques, that can detect an attempt of leaving database in an inconsistent state. DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.

- Query Language – DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and the filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.
- ACID Properties – DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database to stay healthy in multi-transactional environments and also in case of failure.
- Multiuser and Concurrent Access – DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.
- Multiple views – DBMS offers multiple views for different users. A user in the Sales department will have a different view of the database from the person working in the Production department. This feature enables the users to have a concentrate view of the database according to the requirements.
- Security – Features like multiple views offer security to certain extent when users are unable to access the data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. It can also be helpful in deciding how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

### 1.3 APPLICATIONS OF DBMS

Applications of Database Management Systems:

- **Telecom:** There is a database to keep track of the information regarding the calls made, network usage, customer details etc. Without the database system it is hard to maintain such huge amounts of data which gets updated every millisecond.



- **Industry:** Whether it is a manufacturing unit, a warehouse or a distribution centre, each one needs a database to keep the records of the ins and outs. For example, a distribution centre should keep a track of the product units that were supplied to the centre as well as the products that got delivered from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System:** For storing information regarding a customer, keeping a track of his/her day to day credit and debit transactions, generating bank statements etc is done with through Database management systems.
- **Education sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding the student, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is lots of inter-related data that needs to be stored and retrieved in an efficient manner.
- **Online shopping:** You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

## **1.4 PROBLEM DESCRIPTION/STATEMENT**

The purpose of Internships 101- An internship agency is to provide a platform for Internships and provide students a chance to collaborate without them having to visit various websites to find a suitable internship. Keeping that in mind, we have collected details of students that are: their basic details, marks, extracurricular activities and previous internship experience. The list of students who have entered their basic details can view it. It provides both sign up and login option.

## CHAPTER 2

### REQUIREMENT ANALYSIS

#### 2.1 HARDWARE REQUIREMENTS

The Hardware requirements are very minimal and the program can be run on most of Machines.

Processor	:	Pentium 4 processor	
Processor	:	2.4GHz	
Speed			
RAM	:	1GB	
Storage	:	40GB	
Space			
Monitor	:	1024*768	or
Resolution		1336*768	or
		1280*1024	

#### 2.2 SOFTWARE REQUIREMENTS

1. Operating System used : Windows10
2. Language : Java
3. WAMPPServer : MySQL, PhpMyAdmin
4. IDEused : NetBeans8.2

#### 2.3 FUNCTIONAL REQUIREMENTS

##### 2.3.1 Major Entities

Users: Users is the entity that signs up as a student and a match with the potential internship company is made. Few attributes are name, email, phno, address, pw.

Students: Students is the entity that collects students basic details. Few attributes are id, name, age, fees, and photo.

Marks: Marks is the entity that collects marks details. Few attributes are id, sem 1, sem 2, sem 3, sem 4, sem 5, sem 6, sem 7, sem 8, cgpa.

Internships: Internships is the entity that collects student's previous internship details. Few attributes are role, company, loc, start, end, type, and certificate.

Extracurricular: Extracurricular is the entity that collects Extracurricular details. Few attributes are actname, level, exp, img.

## **2.3.2 End User Requirements**

The technical requirements for the project are mentioned below.

### **2.3.2.1 NETBEANS IDE**

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems.

The IDE provides comprehensive support for JDK 7 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the GlassFish Server, the reference implementation of Java EE.

### **2.3.2.2 PHP**

PHP is a server-side scripting language designed primarily for web development but is also used as a general-purpose programming language. Originally created by RasmusLerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP:Hypertext Pre-processor.

PHP code can be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code can also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is a free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers, on almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone into creating a formal PHP specification. HP development began in 1995 when RasmusLerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used in order to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could help to build simple, dynamic web applications. To accelerate bug reporting and to improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group on June 8, 1995. This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler, more limited and less consistent.

### **2.3.2.3 MySQL**

MySQL is a Relational Database Management System (RDBMS). MySQL server can manage many databases at the same time. In fact, many people might have different databases managed by a single MySQL server. Each database consists of a structure to hold onto the data itself. A data-base can exist without data, only a structure, be totally empty, twiddling its thumbs and waiting for data to be stored in it.

Data in a database is stored in one or more tables. You must create the data-base and the tables before you can add any data to the database. First you create the empty database. Then you add empty tables to the database. Database tables are organized in rows and columns. Each row represents an entity in the database, such as a customer, a book, or a project. Each column contains an item of information about the entity, such as a customer name, a book name, or a project start date. The place where a particular row and column intersect, the individual cell of the table, is called a field. Tables in databases can be related. Often a row in one table is related to several rows in another table. For instance, you might have a database containing data about

books you own. You would have a book table and an author table. One row in the author table might contain information about the author of several books in the book table. When tables are related, you include a column in one table to hold data that matches data in the column of another table. MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

MySQL is a database management system. A database is a structured collection of data. It can be anything from a simple shopping list to a picture gallery or the vast amount of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to refer to the current version of the SQL Standard.

MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations.

MySQL Database Server is very fast, reliable, and easy to use.

MySQL Server was originally developed to handle large databases and has been successfully used in highly demanding production environments for several years. MySQL Server today

offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

MySQL Server works in a client/server or embedded systems. The MySQL Database Software is a client/server system which consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

#### **2.3.2.4 XAMPP Server**

Xampp server installs a complete, ready-to-use development environment. Xampp server allows you to fit your needs and allows you to setup a local server with the same characteristics as your production.

While setting up the server and PHP on your own, you have two choices for the method of connecting PHP to the server. For many servers, PHP has a direct module interface (also called SAPI). These servers include Apache, Microsoft Internet Information Server, Netscape and iPlanet servers. Many other servers support ISAPI, the Microsoft module interface (Omni HTTPd for example). If PHP has no module support for your web server, you can always use it as a CGI or FastCGI processor. This means you set up your server to use the CGI executable of PHP to process all PHP file requests on the server.

## CHAPTER 3

### DATABASE DESIGN

#### 3.1 Entities, attributes and relationships

From the above list of entities, we can conclude that Users, Students, Marks, Internship, Extracurricular are the 5 core entities in our Internships 101 – Internship agency. Their attributes are as follows:

Users: Email,

Phno

Address

Pw

Students :

Id

Name

Age

Fees

Photo

Marks :

Id

Sem 1

Sem 2

Sem 3

Sem 4

Sem 5

Sem 6

Sem 7

Sem 8

Cgpa



Internships:

Role  
Company  
Loc  
Start  
End  
Type  
Certificate

Extracurricular:

Actname  
Level  
Exp  
Img

### 3.2 ER and schema

Figure 3.1 shows the relationship between entities.

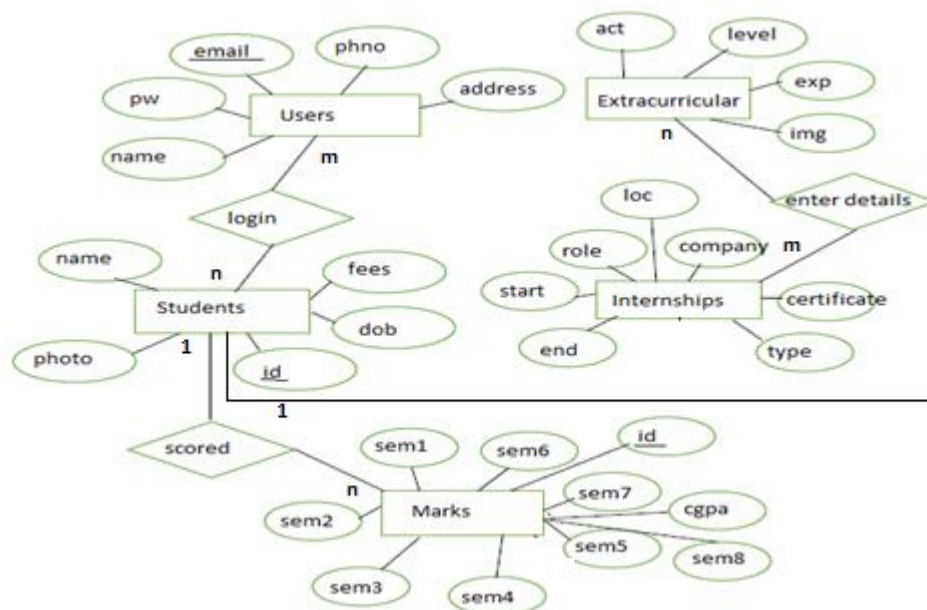


Fig.3.1 ER Diagram for Internships 101 – Internship agency

### 3.3 Relational Schema

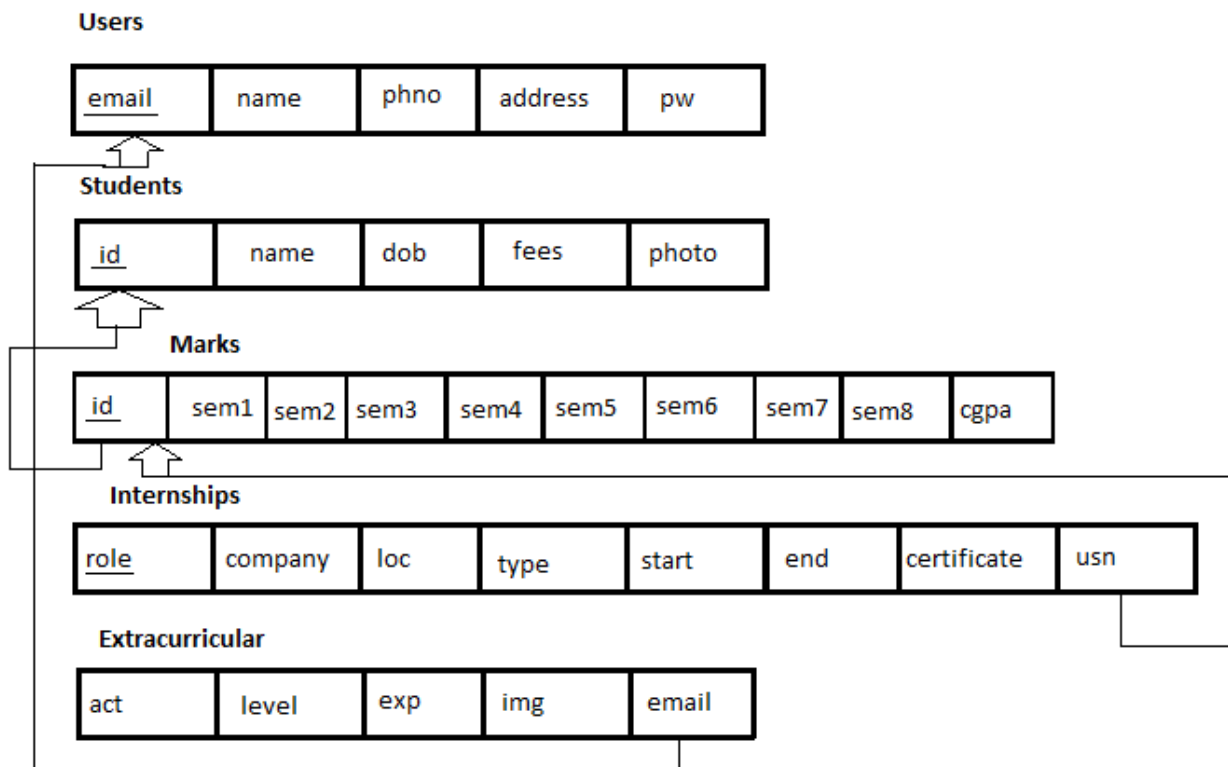


Fig.3.2. Schema for Internships 101 – An Internship Agency

## CHAPTER 4

### IMPLEMENTATION

#### 4.1 Creating Database Connection

1. PHP provides built-in database connectivity for a wide range of databases – MySQL, PostgreSQL, Oracle, Berkeley DB, Informix, Lotus Notes, and more.

```
2. public Connection MySqlConnection(){
    Connection conn=null;
    try {
        conn=DriverManager.getConnection("jdbc:mysql://localhost:3307/project","root","");
        //JOptionPane.showMessageDialog(null, "Mysql DB Connection Successful...");
        return conn;
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Mysql Connection Failed....");
        return null;
    }
}
```

Architecture used (4-TIER architecture)

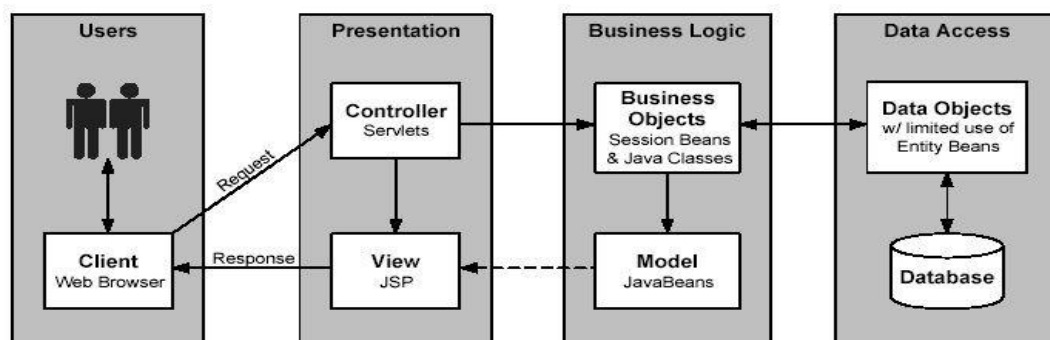


Figure 4.1 The 4-TIER architecture

Four Tier architecture is a client-server architecture in which presentation, application processing, and data management functions are physically separated. Four-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application.

## **Presentation layer**

This is the topmost level of the application. The presentation tier displays information related to services such as browsing merchandise, purchasing and shopping cart contents. It also communicates with other tiers and puts out the results to the browser/client tier and to all other tiers in the network. In simple terms, it is a layer which users can access directly (such as a web page, or an operating system's GUI).

## **Business layer**

Business layer or domain logic is the part of the program that encodes the real-world business rules which determine how data can be created, stored, and changed. It is contrasted with the remainder of the software that might be concerned with lower-level details of managing a database or displaying the user interface, system infrastructure, or generally connecting various parts of the program.

## **Data access layer**

A Data Access Layer (DAL) in computer software is a layer of computer program which provides simplified access to data stored in persistent storage.

For example, the DAL might return a reference to an object (in terms of object-oriented programming) with its attributes instead of a row of fields from a database table. This allows the client (or user) modules to be created with a higher level of abstraction. This kind of model could be implemented by creating a class of data access methods that directly reference a corresponding set of database stored procedures. Another implementation could potentially retrieve or write records to or from a file system. The DAL hides the complexity of the underlying data store from the external world.

**Control layer** The control layer is responsible for the communication between business and presentation layer. It connects logic and data with each other and provides a better connectivity and separation between layers.

## 4.2 Pseudo Code for Major Functionalities

**Login page:** It is used for login purposes. When we enter the correct email and password it will go to the next page. We can use signup to create an account.

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Connection conn=MySqlConnection();  
    PreparedStatement ps;  
    ResultSet rs;  
    String name=jTextField1.getText();  
    String repass = String.valueOf(jPasswordField1.getText());  
    String query= "SELECT * FROM `users` WHERE `name`=? AND `pw`=?";  
    try{  
        ps= conn.prepareStatement(query);  
        ps.setString(1, name);  
        ps.setString(2, repass);  
        rs= ps.executeQuery();  
        if(rs.next())  
        {  
            JOptionPane.showMessageDialog(null, "Login Successful!");  
            StudentsForm Info = new StudentsForm();  
            Info.setVisible(true); }  
        else{  
            JOptionPane.showMessageDialog(null, "Incorrect password");  
        }  
    }  
    catch(SQLException ex){  
        Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    }  
}
```

**Sign up page:**

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    if ((jTextField1.getText()!=null || jTextField2!=null  
        || jTextField3!=null || jTextField4!=null || jPasswordField1!=null)) {  
        try {
```

```

        Connection conn=MySQLConnection();
        PreparedStatement ps=conn.prepareStatement("insert into users"
            + "(name,phno,email,address,pw) values (?, ?, ?, ?, ?)");
        ps.setString(1, jTextField1.getText());
        ps.setString(2,jTextField2.getText());
        ps.setString(4,jTextField3.getText());
        ps.setString(3,jTextField4.getText());
        ps.setString(5,jPasswordField1.getText());
        String repass = String.valueOf(jPasswordField2.getText());
        if((jPasswordField1.getText().equals(repass))) {
            int res=ps.executeUpdate();
            if (res>=1) {
                JOptionPane.showMessageDialog(null,"Sign up successful!");
            }
        }else
            JOptionPane.showMessageDialog(null, "Enter the same password");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jPasswordField2.setText("");
    jPasswordField1.setText("");
}
else{
    JOptionPane.showMessageDialog(null, "Enter all details...");
}
}

```

**Home page:**

```

public void fillTable(){
    ArrayList<StudentBean> al=retrieveData();
    DefaultTableModel model=(DefaultTableModel)jTable1.getModel();
    model.setRowCount(0); // Empty/clear the table
    Object[] row=new Object[4];

```

```
for (int i = 0; i < al.size(); i++) {
    row[0]=al.get(i).getId();
    row[1]=al.get(i).getName();
    row[2]=al.get(i).getFees();
    row[3]=al.get(i).getDob();
    model.addRow(row);
}
//model.setRowCount(0);
}
//Show item/data to fields:
public void showItemToFields(int index){
    jTextField_id.setText(Integer.toString(retrieveData().get(index).getId()));
    jTextField_name.setText(retrieveData().get(index).getName());
    jTextField_fees.setText(Float.toString(retrieveData().get(index).getFees()));
    try {
        java.util.Date dob=null;
        dob=new SimpleDateFormat("dd-MM-yyyy").parse((String)retrieveData().get(index).getDob());
        jDateChooser1_dob.setDate(dob);
    } catch (Exception e) {
        System.out.println("Error at showItemToFields method "+e);
    }
    label_photo.setIcon(resetImageSize(null, retrieveData().get(index).getPhoto()));
}
```

**Insert:** This is for insertion into the tables

```
private void jButton_saveActionPerformed(java.awt.event.ActionEvent evt) {
    if ((jTextField_id.getText()!=null || jTextField_name!=null
    || jTextField_fees!=null || jDateChooser1_dob!=null)&& photopath!=null) {
        try {
            Connection conn=MySQLConnection();
            PreparedStatement ps=conn.prepareStatement("insert into students"
            + "(id,name,fees,dob,photo) values (?, ?, ?, ?, ?)");
            ps.setInt(1, Integer.parseInt(jTextField_id.getText()));
            ps.setString(2,jTextField_name.getText());
```

```

ps.setFloat(3, Float.parseFloat(jTextField_fees.getText()));
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
String dob1=sdf.format(jDateChooser1_dob.getDate());
ps.setString(4,dob1);
InputStream is=new FileInputStream(new File(photopath));
ps.setBlob(5, is);
int res=ps.executeUpdate();
fillTable();
if (res>=1) {
    JOptionPane.showMessageDialog(null, res+" Number of Student"
        + " inserted into database .....");
}else
    JOptionPane.showMessageDialog(null, "Student Insertion Failed ....");
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
jTextField_id.setText("");
jTextField_name.setText("");
jTextField_fees.setText("");
}else{
    JOptionPane.showMessageDialog(null, "All fields are compulsory ....");
}
}

```

**Update:** This is for updating the table entries

```

private void jButton_updateActionPerformed(java.awt.event.ActionEvent evt) {
    if (jTextField_id!=null || jTextField_name!=null
        || jTextField_fees!=null || jDateChooser1_dob!=null) {
        String qry=null;
        PreparedStatement ps=null;
        Connection conn=MySQLConnection();
        if (photopath==null) {
            try {
                qry="update students set name=?, fees=?,dob=? where id=?";
            }

```



```
ps=conn.prepareStatement(qry);
ps.setString(1,jTextField_name.getText());
ps.setFloat(2, Float.parseFloat(jTextField_fees.getText()));
SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
String dob1=sdf.format(jDateChooser1_dob.getDate());
ps.setString(3,dob1);
ps.setInt(4, Integer.parseInt(jTextField_id.getText()));
int res=ps.executeUpdate();
fillTable();
if (res>=1) {
    JOptionPane.showMessageDialog(null, res+" Number of Student"
        + " inserted into database .....");
}else
    JOptionPane.showMessageDialog(null, "Student Insertion Failed ....");
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
}
jTextField_id.setText("");
jTextField_name.setText("");
jTextField_fees.setText("");
}else{
    try {
        InputStream is=new FileInputStream(new File(photopath));
        qry="update students set name=?, fees=?,dob=?, photo=? where id=?";
        ps=conn.prepareStatement(qry);
        ps.setString(1,jTextField_name.getText());
        ps.setFloat(2, Float.parseFloat(jTextField_fees.getText()));
        SimpleDateFormat sdf=new SimpleDateFormat("dd-MM-yyyy");
        String dob1=sdf.format(jDateChooser1_dob.getDate());
        ps.setString(3,dob1);
        ps.setBlob(4, is);
        ps.setInt(5, Integer.parseInt(jTextField_id.getText()));
        int res=ps.executeUpdate();
        fillTable();
```

```
        if (res>=1) {
            JOptionPane.showMessageDialog(null, res+" Student"
                + " Updated into database .....");
        }else
            JOptionPane.showMessageDialog(null, "Student Updation Failed ....");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
    jTextField_id.setText("");
    jTextField_name.setText("");
    jTextField_fees.setText("");
}
} else
    JOptionPane.showMessageDialog(null, "All fields are mandatory.....");
}
}
```

**Delete:** This is for deleting the table entries

```
private void jButton_deleteActionPerformed(java.awt.event.ActionEvent evt) {
    if (jTextField_id.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Please Enter the student id");
    } else {
        try {
            String qry="delete from students where id=?";
            Connection conn=MySQLConnection();
            PreparedStatementps=conn.prepareStatement(qry);
            ps.setInt(1, Integer.parseInt(jTextField_id.getText().toString()));
            int res=ps.executeUpdate();
            fillTable();
            if (res>=1) {
                JOptionPane.showMessageDialog(null, "Student Deleted Successful ....");
            } else
                JOptionPane.showMessageDialog(null, "Student Deletion failed ....");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}
```

---

**Trigger:** Trigger is added to marks table (fig. 4.2) which calculates the percentage of marks automatically upon entering the cgpa.

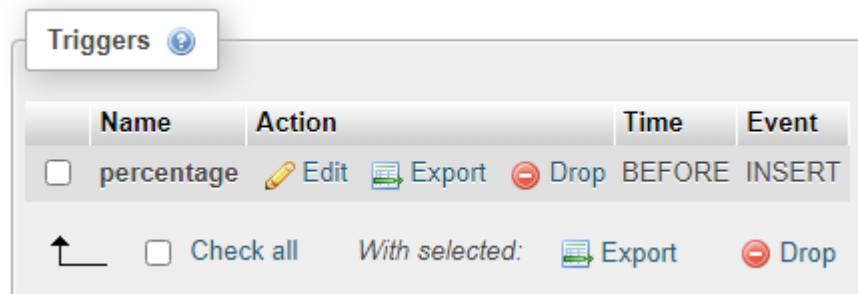


Figure 4.2 Trigger

## CHAPTER 5

### RESULTS , SNAPSHOTS AND DISCUSSIONS

**Login page:** This page allows the users to login into the application as shown in fig. 5.1.



The screenshot shows a login window with a light blue background. At the top, the word "LOGIN" is centered in bold black text. Below it, there are two input fields: "Username" with the text "namratha" and "Password" with masked characters "\*\*\*\*\*". At the bottom, there are three buttons: "Login", "Sign up", and "Close".

Fig. 5.1 Login page

**Signup page:** This page allows users to create a new account if the account does not exist as shown in fig. 5.2.



The screenshot shows a signup window with a light blue background. At the top, the words "SIGN UP" are centered in bold black text. Below it, there are six input fields: "Name" with "namratha", "Phone number" with "9108357779", "Email" with "19cs084@gmail.com", "Address" with "Vijayanagar", "Password" with "\*\*\*\*\*", and "Confirm password" with "\*\*\*\*\*". At the bottom, there are three buttons: "Save", "clear", "exit", and "Login".

Fig. 5.2 Signup page

**Homepage:** Allows the students to enter their basic details and also to view their entries as shown in fig. 5.3.

**INTERNSHIPS 101**  
RR Nagar, Bangalore  
Student Information

**Id:**  **New**

**Student Name:**  **Insert/Save**

**Fees:**  **Update**

**Date of Birth:**  **Delete**

**Photo:**  **Proceed to enter marks**

**Search Student By Name:**

Id	Student Name	Fees	Date of Birth
1	Namratha	10000.0	06-01-2017
2	mahima n shah	9999.0	06-01-2022

Fig. 5.3 Homepage

**Marks entry:** Allows the students to enter their marks and cgpa as shown in fig. 5.4.

**ACADEMICS** \* Fields are compulsory  
Enter SGPA (Enter till current semester, enter as 0 for other s...

**USN\*:**

**Semester ...**

**Semester 2 :**

**Semester 3 :**

**Semester 4 :**

**Semester 5 :**

**Semester 6 :**

**Semester 7 :**

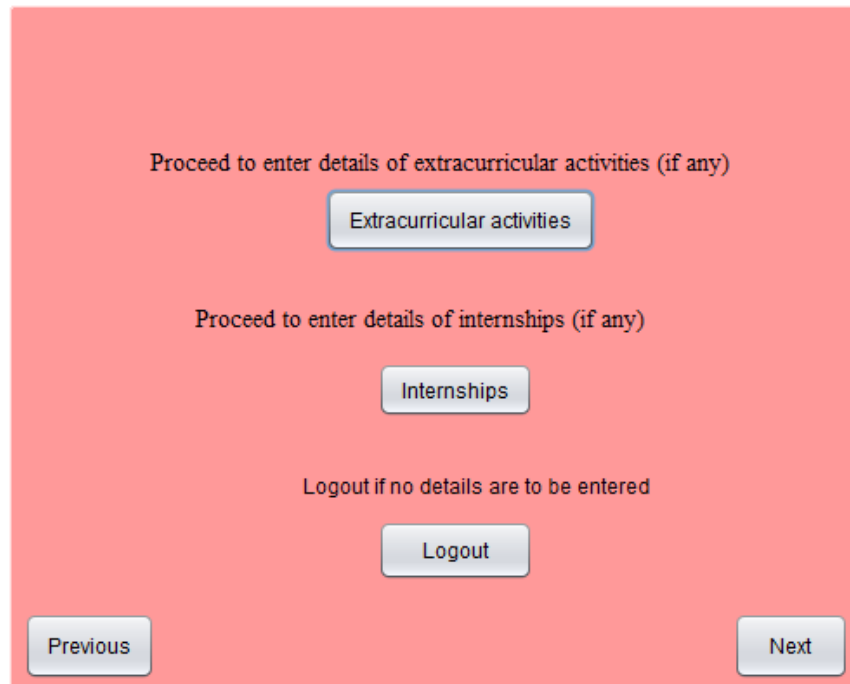
**Semester 8 :**

**CGPA\*:**

**Save Back Next**

Fig. 5.4 Marks entry

**Extracurricular activities and previous experience landing page:** Students can either proceed to enter internships details or extracurricular activities details as shown in fig. 5.5.



Proceed to enter details of extracurricular activities (if any)

Extracurricular activities

Proceed to enter details of internships (if any)

Internships

Logout if no details are to be entered

Logout

Previous Next

Fig. 5.5 Extracurricular activities and previous experience landing page

**Extracurricular activities:** Students can enter their extracurricular activity details and logout else he can proceed to enter the internship details as shown in fig. 5.6.



**EXTRACURRICULAR ACTIVITIES**

Activity nam... Hockey

Level : Beginner

Years of experience : 5

Upload certificate :

Certificate of Participation

Namratha J

Select certificate

Add another activity detail

Save back Logout

Fig. 5.6 Extracurricular activities

**Previous internships:** Students can enter their work experience details and logout else he can proceed to enter the extracurricular details as shown in fig. 5.7.

**WORK EXPERIENCE**

Role : Tech intern Start date : Jan, 2022

Compa... Amazon End date Feb, 2022

Locatio... Toronto Type : paid

Upload certificate :

Select certificate

Add another experience

Save back Logout

Fig. 5.7 Previous internships

### Table entries:

**Users table:** Fig 5.8 shows entries of Users table.

				name	phno	email	address	pw
<input type="checkbox"/>	Edit	Copy	Delete	namratha	9108357779	1rn19cs084@gmail.com	Vijayanagar	namratha
<input type="checkbox"/>	Edit	Copy	Delete	anagha	8979678756	anagha@gmail.com	Bangaladesh	qwerty
<input type="checkbox"/>	Edit	Copy	Delete	Ananya	7878787878	anaya@gmail.com	Jayanagar	password
<input type="checkbox"/>	Edit	Copy	Delete	divya	8877777777	divya@gmail.com	RR Nagar	divya
<input type="checkbox"/>	Edit	Copy	Delete	mahima	8888888888	kkk@gmail.com	tt nagar	qwerty
<input type="checkbox"/>	Edit	Copy	Delete	harshitha	999999	ww@gmail.com	RR Nagar	lkjhg
<input type="checkbox"/>	Check all	With selected:		Edit	Copy	Delete	Export	

Fig. 5.8 Users table

**Students table:** Fig 5.9 shows entries of Students table.

<div><div></div><div></div></div>			id	name	fees	dob	photo	
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	1	Namratha	10000	06-01-2017	[BLOB - 36.4 KiB]
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	2	mahima n shah	9999	06-01-2022	[BLOB - 36.4 KiB]
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	3	Trisha	77777	08-02-2001	[BLOB - 5.4 KiB]
<div><div></div><div></div></div>	<input type="checkbox"/> Check all	With selected:		<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	<div><div></div><div>Export</div></div>	

Fig. 5.9 Students table

**Marks table:** Fig 5.10 shows entries of Marks table.

<div><div></div><div>←T→</div></div>			<div>▼</div>	id	sem1	sem2	sem3	sem4	sem5	sem6	sem7	sem8	cgpa	percent
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	1RN19CS009	9	9	8	9	10	9	9	0	9.5	87.5
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	1rn19cs056	9	9	9	9	9	9	9	9	9	1.28571
<input type="checkbox"/>	<div><div></div><div>Edit</div></div>	<div><div></div><div>Copy</div></div>	<div><div></div><div>Delete</div></div>	1rn19cs091	9	9	9	9	9	9	8	8	9	82.5

Fig. 5.10 Marks table

**Extracurricular table:** Fig 5.11 shows entries of Extracurricular table.

actname	level	exp	img
badmittion	intermediate	5	[BLOB - 90.5 KiB]
tennis	beginner	3	[BLOB - 452 B]
chess	Beginner	2	[BLOB - 8.1 KiB]
Chess	Beginner	5	[BLOB - 141.2 KiB]
chess	Beginner	3	[BLOB - 141.2 KiB]
Hockey	Beginner	5	[BLOB - 141.2 KiB]

Fig. 5.11 Extracurricular table

**Internship table:** Fig 5.12 shows entries of Internship table.

role	company	loc	start	end	type	certif
1	dd	fff	01-01-2022	31-01-2023	paid	[BLOB - 5.9 KiB]
product intern	kkk	up	04-01-2022	06-01-2022	paid	[BLOB - 73.3 KiB]
AI intern	Elite techno	Bangalore	01-01-2022	17-01-2023	paid	[BLOB - 93.3 KiB]
intern	Vivo	Hyderabad	01-01-2022	07-01-2023	unpaid	[BLOB - 93.3 KiB]
Tech intern	Amazon	Toronto	01-01-2022	01-02-2022	paid	[BLOB - 93.3 KiB]

Fig. 5.12 Internship table



## **CHAPTER 6**

### **CONCLUSION AND FUTURE ENHANCEMENTS**

#### **6.1 Conclusion**

The purpose of Internships 101- An internship agency is to provide a platform for Internships and provide students a chance to collaborate without them having to visit various websites to find a suitable internship. Keeping that in mind, we have collected details of students that is: their basic details, marks, extracurricular activities and previous internship experience. The list of students who have entered their basic details can view it. It provides both sign up and login option.

Our back-end tech is reliable and efficient enough to not create any hurdles from the time the user logs into the website to the time when he/she has found the information needed. We hope that the users have a seamless experience.

#### **6.2 Future Enhancements**

The project database could be improved by adding some more attributes and tables. Table marks can further be improved by adding another trigger to calculate the cgpa. Further changes can be easily done by changing the code. The front end of the website can be made more attractive by using more advanced concept of jframes.

## **BIBLIOGRAPHY**

1. RamezElmarsri and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7<sup>th</sup> Edition.
2. <https://stackoverflow.com/>
3. <https://www.w3schools.com/>
4. [www.php.net](http://www.php.net)
5. <https://trainings.internshala.com/web-development>