VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JnanaSangama, Belgaum-590014

A Web Mini Project Report

On

"INTERNSHIP DATABASE MANAGEMENT SYSTEM"

Submitted in Partial fulfillment of the Requirements for the VII Semester of the Degree of

Bachelor of Engineering
In
Computer Science & Engineering
By

AKSHAY AMRUT MORAB(1CE17CS008)
BHUVANESHWARI M(1CE17CS024)

Under the Guidance of

Mrs. LAXMI M C
Asst. Professor, Dept. of CSE

CITY ENGINEERING COLLEGE

Doddakallasandra, Kanakapura Road,
Bengaluru-560061

CITY ENGINEERING COLLEGE

Doddakallasandra, Kanakapura Road, Bengaluru-560061

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Web Mini Project work entitled **“INTERNSHIP DATABASE MANAGEMENT SYSTEM”** has been carried out by **AKSHAY AMRUT MORAB (1CE17CS008) and BHUVANESHWARI M (1CE17CS024),** bonafide students of City Engineering College in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2020-2021**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Web Mini Project Report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Mrs.Laxmi M C	**Mr. B Vivekavardhana Reddy**	**Dr. V. S Rama Murthy**
Asst.Prof, Dept.of CSE	Head, Dept. of CSE	Principal

External Viva

Name of the examiners   Signature with date

1.

2.

ACKNOWLEDGEMENT

While presenting this Web Mini Project on "**Internship Database Management System",** we feel that it is our duty to acknowledge the help rendered to us by various persons.

Firstly we thank God for showering his blessings on us. We are grateful to our institution City Engineering College for providing us a congenial atmosphere to carry out the project successfully.

We would like to express our heartfelt gratitude to **Dr. V S Ramamurthy,**
Principal, CEC, Bangalore, for extending his support**.**

We would also like to express our heartfelt gratitude to **Prof. Vivekavardhana Reddy**, HOD, Computer Science and Engineering whose guidance and support was truly invaluable**.**

We are very grateful to our guide, **Mrs. Laxmi M C**, Asst. Prof., Department of Computer Science, for her able guidance and valuable advice at every stage of our project which helped me in the successful completion of our project.

We would also have indebted to our Parent and Friends for their continued moral and material support throughout the course of project and helping me in finalize the presentation.

Our hearty thanks to all those have contributed bits, bytes and words to accomplish this Project.

AKSHAY AMRUT MORAB(1CE17CS008)

BHUVANESHWARI M(1CE17CS024)

ABSTRACT

Internship Database Management System automates the process of managing and tracking multiple information of each individual students and projects where information can be accessed. The Internship Database Management System maintains a database of each info including every courses/projects provided to the student which is being provided to him, and the domain which has been allotted for him and Instructor can view details of the student and also he can allocate the domain, course and project topic for the student, and the student can input his details about his name, usn, year of joining, year of passing and other personal details about him while joining internship.

The purpose of this project is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, full-filling their requirements such that their valuable information/data can be stored for a longer period of time with easy accessing and manipulation of the same. The required software and hardware are easy to work with.

Internship Database Management System, as described above, can lead to error-free, secure, reliable and fast management system. It can assist the other-personnel to concentrate on their other activities rather than keeping their focus on this topic. The organization can maintain computerized records without redundant entries, that means that one need not be distracted by information that is not relevant, while being able to reach the information

The main aim is to gather valuable data/information that can be stored for a longer period of time with easy accessing and manipulation of the same. Basically the project describes how to manage good performance and better services for the internship management.

CONTENTS

LIST OF FIGURES

PROJECT TITLE

INTERNSHIP DATABASE MANAGEMENT SYSTEM

CHAPTER-1

INTRODUCTION

1.1 Introduction to the title of project

The management of the internship databases can be managed throughout the organisation through divided units of any management and supervisors who have been allotted for the required each system\functions. They have features like information storage, supervision, equipment maintenance and details provided by the student databases.

The project has been planned to be having the view of disturbed architecture, with centralized storage of databases. The application for the storage of the data is planned. Using the constructs of MY-SQL Server and all the user interfaces have been designed by using the JAVA-SERVELT (J2EE) technology. The database connectivity is planned using the “SQL-connection” methodology. The standards of security and data protective mechanism has been given a big choice of proper usage. The application takes care of different modules and their associated reports, which are produced as per the applicable strategies and standards that are put forward by the administrative staff

This entire project has been developed keeping in view of the disturbed client server computing technology in mind. The specification has been normalized to 3NF to eliminate all the anomalies that may arise due to database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as MY-SQL Server. The basic constructs of table specs, clusters and indexes has been exploited to provide higher consistency and reliability for the data storage manages the data consistency with proper business rules or validations. The database connectivity was planned using the latest “SQL-Connection” technology provided by the Microsoft Corporation. The authentication and authorization were crosschecked at all relevant stages. The user level accessibility has been restricted to two zones.

1.2 Architecture

The process of constructing each kind of system is not so simple. It involves a mutual development of application program and database. The application program is actually the bridge between the users and the database, where the data is stored. Thus, the well-developed application program and the database are very important for the reliability, flexibility and functionality of the system. The so-defined system differentiates to each other and their development comprises a great variety of tasks to be resolved and implemented. "Organizations depend on information-systems in order to stay competitive. Productivity, which is crucial to staying competitive, can be increased through better information-systems."

CHAPTER-2

PROBLEM STATEMENT

2.1 Problem Discussion

This report's documentation goes through the whole process of both application program and database development. It also comprises the development tools which are being utilized for these purposes.

The system should consist of an application program, on one hand, and a database (repository of data) on the other. The program should perform the basic operations upon the database retrieving, inserting, updating and deleting data. Any additional functionality is a goal of further module development. It is a kind of strategy to start the development from designing and constructing the database, as this structure will determine the further structure of the application program. The logical database model (tables, their content and the relationship between them) should respond to the given task and cover the basic requirements. The interface of the program should be user friendly, and the program should be easy to use as if possible. Both controls forms should logically and functionally be related within the program and fully respond to the structure of the database, every time when a query is needed to be performed upon it. Exception-handling should also be taken into an account during the system's development due to an eventual exception that may occur.

2.2 Instructor Domain Diagram and Analysis

The instructor domain diagram explains how the registration of instructor and student runs in the real world. After entering into the home page we have two choices, instructor and student. Student can register his details through the registration page. In instructor registration page, we have two choices after registering the instructor details namely, view and register. If we select the register option, it leads us to register other details like usn, project name, project topic. If we select the view option we can view the details of the student with particular usn.

2.3 ER-Diagram:

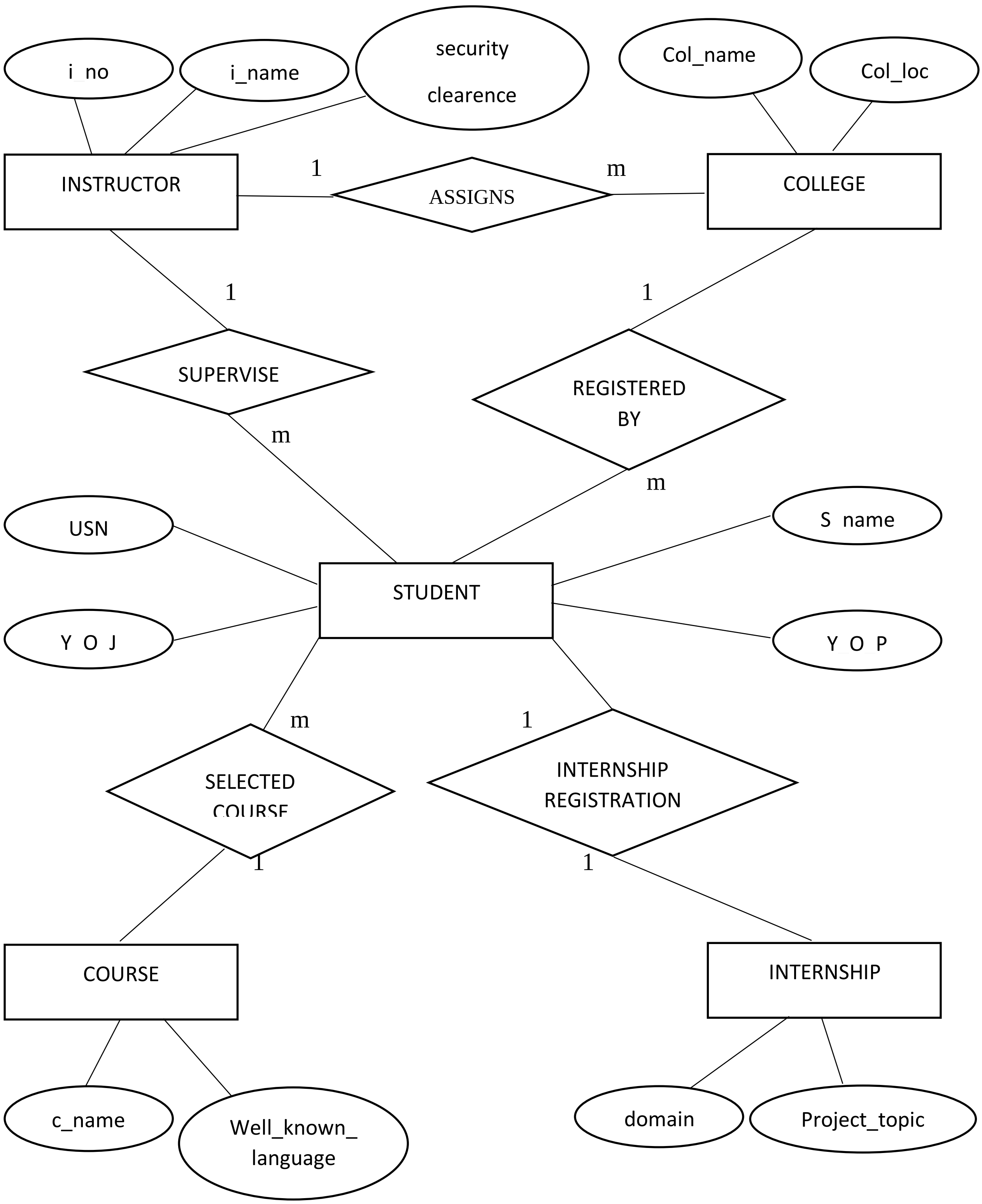
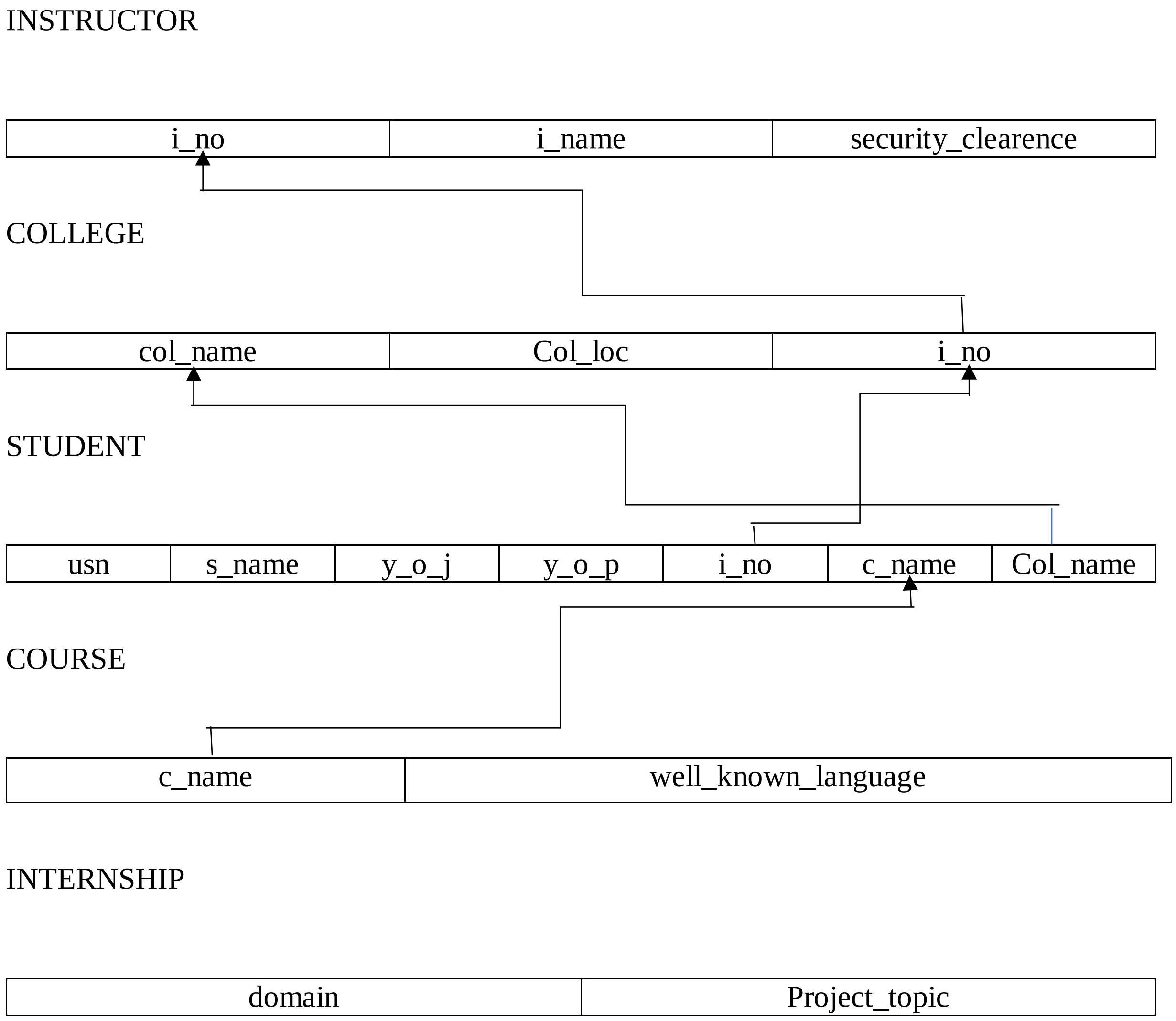


Fig.no-2.3

2.4 Schema-Diagram:



INSTRUCTOR

i_no	i_name	security_clearence

COLLEGE

col_name	Col_loc	i_no

STUDENT

usn	s_name	y_o_j	y_o_p	i_no	c_name	Col_name

COURSE

c_name	well_known_language

INTERNSHIP

domain	Project_topic

Fig no-2.4

2.5 Study of the System

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI’s at the top level have been categorized as

1) Administrative User Interface
2) The Operational / Generic User Interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administration with all the transactional states like data insertion, data deletion and data updating along with the extensive data search capabilities.

The Operational / Generic User Interface helps the user upon the system in transactions through the existing data and required services. The Operational User Interface also helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities

2.5.1 Number of Modules

The system after careful analysis has been identidied to be presented with the following modules:

The modules involved are:

1. Instructor
2. Student

Instructor:

In this module, we have two choices after registering the commander details namely, view and register. If we select the register option, it leads us to register other details like college, course and internship. If we select the view option we can view the details of the student with particular usn.

Student:

Student can register his details through the registration page. The student can select college, course and internship details in it. Later the student is forwarded to the success page.

CHAPTER – 3

SOFTWARE REQUIREMENTS AND ANALYSIS

3.1 Software Requirement Specification

Operating system: Windows-10

Database Connectivity: MYSQL

Programming Tools: NetBeans 8.2, Eclipse

Server-side Scripting: Java

Front-End Design: HTML, CSS

3.2 Functional Requirements

User ID is provided with the register. The system must only allow user with valid-ID and password to enter the system. The System performs authorization process which decides what user level can access to. The user must be able to logout after they finish using the system. The system must be able to delete information if it is specified wrong.

3.3 Software and Hardware Requirements

3.3.1 Software Requirements

Operating System- Windows 10 is used as the operating system as it is stable and supports more features and is friendlier user interface.

Database MYSQL- My SQL is used as the database as it is easy to maintain and retrieve records by simple queries which are in English language which are used to understand and it is also easy to write.

Development Tools and Programming Database- HTML is used to write the whole code and develop web pages with CSS, NetBeans and server side scripting by JAVA.

HTML-5

Hypertext-Mark-up language is the standard mark-up language for creating web pages and web applications. With cascading style sheets and java, it forms a triad of cornerstone technologies for the World Wide Web.

APACHE-TOMCAT SERVER

Apache-tomcat server is an open source implementation of Java-Servlet, JavaServer Pages, Java Expressions Language and WebSocket technologies. Tomcat provides a pure-Java HTTP server web environment in which java code can run.

SQL

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for the stream processing in a relational data stream management system.

ECLIPSE (IDE)

It is an Integrated Development Environment (IDE) used in computer programming. It contains a base workspace and extensible plug-in system for customizing the environment. Eclipse is mostly written in Java and its main purpose is developing java applications, but it may be also used in development of web pages using HTTP.

CASCADING STYLE SHEETS(CSS)

Cascading style sheets is a style sheet language used to describe the presentation of a document written in HTML or XML. It describes how elements should be rendered on screen, on paper, in-speech or other media.

CHAPTER-4

DESIGN

4.1 Introduction To Database Design Model

The database of library system is intended to automate the library activities such as creating new receiver, givers and maintaining the details of all the objects which has been given or taken. This also helps in maintenance of providing category.

A database is an integrated collection of logically related records or files consolidated into a common pool that provides data for one or more multiple uses. One way of classifying databases involves the type of content, for example: Bibliographic, Full-Text, Numeric and others.

Other classification methods start from examining database models or database architectures. The data in database is arranged according to a database model. Relational model is the most common. A database management system (DBMS) consists of software that organizes the storage of the data. A DBMS controls the creation and maintenance and use of the database storage structures of organisations and of their end users. It allows organizations to place control of organization-wide database development in the hands of the Database Administrators (DBA) and other specialists. In large systems, DBMS allows users and other software to store and retrieve data in the structured way. Database management systems are usually categorized according to the database model that they support, such as the network, relational or object model. The model tends to determine the query languages that are available to access the database. One commonly used query for the relational database is SQL, although SQL syntax and function can vary from one DBMS to another. A great deal of the internal engineering of a DBMS is independent of the data model, and is concerned with managing factors such as performance, concurrency, integrity and recovery from hardware failures.

4.2 Functional Dependencies

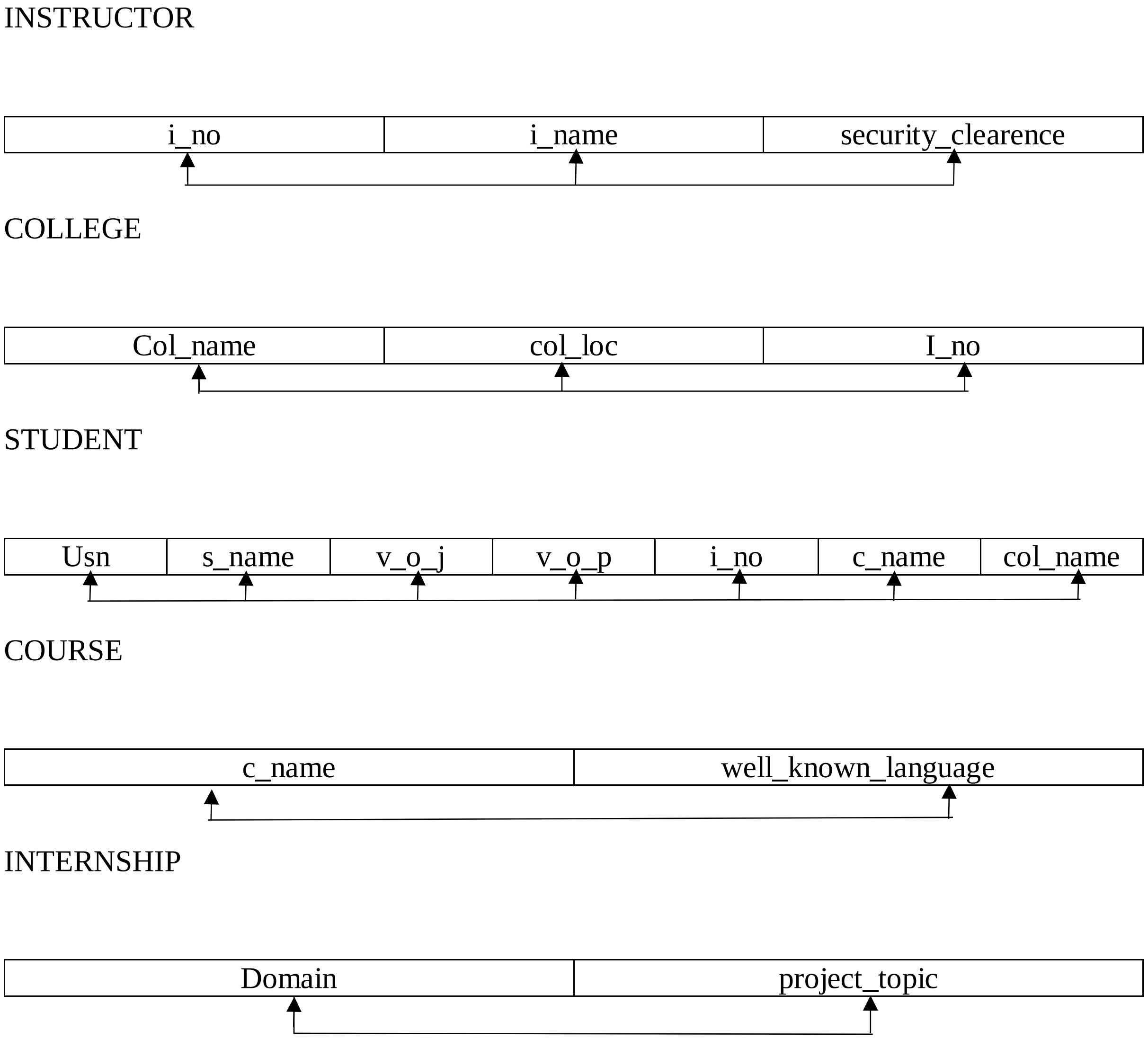


Fig no-4.2

4.3 NORMALIZATION IN SCHEMAS

- There are no multivalued attributes in any of the tables so it is in 1NF
- Every non-prime attribute is fully functionally dependent on the primary key so it is in 2NF
- No transitive keys so 3NF

CHAPTER-5

DATABASE AND FRONTEND DESCRIPTION

5.1 Database Table Creation

Here MYSQL is used to create database where the database where the data which is being executed will be stored in that. To create database in MYSQL the query "create database "will be used and that creates a new table. If the database already exists, the query "use database" is used to choose the given the database which already exists

5.2 Creation of Tables

Here first the database is selected and next after the selection, tables are ready to be created. Here we use the query "create table" with the table name specified after that and the other commands are given in brackets and ended with a semicolon.

After pressing enter the table will be created in SQL and to view the tables give the command "desc tables" to view the tables.

To insert the values into the tables use query "insert into table value" to inert the data into the table and to view the data use query "select * from table" and you can view the command.

5.3 Servlet And MYSQL Connection

It stands for 'Java Database Connectivity' adn it is an API (Application Programming Interface) which consists of a set of java classes, interfaces and exceptions and a specification to which both JDBC driver vendors and JDBC developers adhere when developing applications. RDBMS (Relational Database Management Systems) or third-party vendors develop drivers which adhere to the JDBC specification. Other developers use these drivers to develop applications which access those databases. JDBC driver can access MYSQL. It helps in accessing different RDBMS using different JDBC drivers.

5.4 Student And Instructor Features

5.4.1 Instructor Features

Instructor has features like he can view all the details about any student or internship he is being positioned for the given data and even about the course and domain which is being provided for him and all the other personal information. The Instructor can access all the information by giving the codename “internship” and if he gives any wrong codename he will not be directed access to the information.

5.4.2 Student Features

A student has features such as he can register or enrol to any prescribed domain which he is being specified for it. He can also choose to what course he needs to join and domain he wants to join.

5.5 Front End Design

Front end is designed using of HTML, Java-Servlet, and CSS

HTML – Hyper Text Mark-Up Languages the main mark-up-language for creating web pages and other information that can be displayed in a web browser. HTML is written in form of HTML elements consisting of tags enclosed in angle brackets (<html>) within the web page content. HTML tags commonly come in pairs like <h1> or </h1>, although some tags represent empty elements and so unpaired, for example <img>. The first tag in a pair is the start tag and the second tag is the end tag and in between these tags web designers can add text, further tags, comments and other type of texts based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML elements from building blocks to all websites. HTML allows images and objects embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java-Servlet which effects the behavioural of HTML web pages.

CSS (Cascading Style Sheets)- is a style sheet used for describing the look and formatting of a document written in mark-up language. While most often style used to style web pages and interfaces is written in HTML and XHTML, the language can be applied to any kind of XML document, including the plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages uses CSS style sheets to describe the presentation.CSS is designed primarily to enable the document content from the document presentation, including elements such as the layout, colours, and fonts. This separation can improve more content accessibility, more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content. CSS can also allow the same mark-up page to be presented in different styles for different rendering methods, such as on-screen, in-print, by-voice and on Braille-based tactile devices.

Java-Servlet- is a java software component that extends the capabilities of server. Although servlet can respond to many types of requests, they most commonly implement web containers for hosting web applications on web servers and thus qualify as a server-side servlet web-API. Such web servlets are the java counterpart to other dynamic web content technologies such as PHP and ASP.NET.

5.6 Back-End Design

MYSQL- Is an open-source relational database management system. Its name is a combination of "My", the name of co-founder is Michael-Widenius's daughter and MYSQL full form is Structured Query Language. It is a multithreaded, multiuser, SQL data management system. The project source code is available under the terms of GNU (General Public License) as well as under a variety of property arguments. A table collection of related data entries and it consists of columns and rows. The databases are useful when storing information categorically.

JDBC- It stands for 'Java Database Connectivity' adn it is an API (Application Programming Interface) which consists of a set of java classes, interfaces and exceptions and a specification to which both JDBC driver vendors and JDBC developers adhere when developing applications. RDBMS (Relational Database Management Systems) or third-party vendors develop drivers which adhere to the JDBC specification. Other developers use these drivers to develop applications which access those databases. JDBC driver can access MYSQL. It helps in accessing different RDBMS using different JDBC drivers.

5.7 Programming language selection

Java Servlet is a java software component that extends the capabilities of server. Although servlet can respond to many types of requests, they most commonly implement web containers for hosting web applications on web servers and thus qualify as a server-side servlet web-API. Such web servlets are the java counterpart to other dynamic web content technologies such as PHP and ASP.NET.

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for the stream processing in a relational data stream management system.

5.7.1 Sample Coding:

1.Home.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<img src="https://sgmaersk.files.wordpress.com/2015/07/von-1.jpg"style="z-index:-
1;position:fixed;width:100%;height:100%"/>
<h1><b><center><font size=7 color=white>    INTERNSHIP DATABASE
MANAGEMENT</font></center></b><br><br></h1>
<a href="InstructorForm.html"> <font size=6 color=white>INSTRUCTOR</font></a><br><br><br>
<a href="StudentForm.html"> <font size=6
color=white>STUDENT</font></a><br><br><br><br><br><br><br><br><br><br><br><br><br><
br><br><br><br><br><br><br><br><br>
</body>
</html>
```

2.InstructorForm.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<img src="https://www.growthengineering.co.uk/wp-content/uploads/2016/09/super-teamwork-
01.png" style="z-index:-1;position:fixed;width:100%;height:100%"/>
<h1><b><center><font size=7 color=black>INSTRUCTOR
REGISTRATION</font></center></b><br></h1>
<form action="Instructor" method="post"><br><br><br>
<font size=6 color=white>INSTRUCTOR NUMBER:<input type='text'
name='instructor_numb'/></font><br><br><br>
<font size=6 color=white>INSTRUCTOR NAME:<input type='text'
name='instructor_name'/></font><br><br><br>
<font size=6 color=white>SECURITY CLEARENCE:<input type='text'
name='security_clearence'/></font><br><br><br>
<input type='submit' value='ENTER'/><br><br>
<font size=6 color=black><a href="VisionForm.html">VIEW </a><br></font>
<br><br><br><br><br><br><br><br>
<blockquote><footer><i><center><font size=7 color=black>"Practice is the best of all instructors."
</font></center></i></footer></blockquote>
</form>
```

```
</body>
</html>
```

3.StudentForm.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<font size=7 color=white><img src="https://s31450.pcdn.co/wp-content/uploads/2018/04/tips-online-faculty-id621613860-180420.jpg" style="z-index:-1;position:fixed;width:100%;height:100%"/>
</font>
<h1><b><center><font size=7 color=black>REGISTRATION PAGE</font></center></b><br><br></h1>
<a href="Student"><font size=7 color=black>REGISTER</font></a>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<blockquote><footer><i><center><font size=7 color=black>“There are no shortcuts to any place worth going.”
</font></center></i></footer></blockquote>
</body>
</html>
```

4.InternshipForm.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<img src="https://m.economictimes.com/thumb/msid-69278574,width-1200,height-900,resizemode-4,imgsize-169697/internship-getty.jpg" style="z-index:-1;position:fixed;width:100%;height:100%"/>
<h1><b><center><font size=7 color=black>INTERNSHIP REGISTRATION</font></center></b></h1><br>
<form action="Internship" method="post"><br><br>
<font size=6 color=black>DOMAIN:<input type='text' name='domain'/></font><br><br><br><br>
<font size=6 color=black>PROJECT TOPIC:<input type='text' name='project_topic'/></font><br><br><br><br>
<input type='submit' value='ENTER'/>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br>
```

```
<blockquote><footer><i><center><font size=7 color=black>"The ideal intern is committed, creative, organized, ambitious, independent, and able to crack a smile, whether meeting a celebrity or folding socks."
</font></center></i></footer></blockquote>
</form>
</body>
</html>
```

5.CollegeForm.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<img
src="https://upload.wikimedia.org/wikipedia/commons/9/9d/Morgan_Hall_of_Williams_College_in_the_fall_%2827_October_2010%29.jpg" style="z-index:-1;position:fixed;width:100%;height:100%"/>
<h1><b><center><font size=7 color=black>COLLEGE DETAILS</font></center></b><br></h1>
<form action="College" method="post"><br><br><br>
<font size=6 color=white>COLLEGE NAME:<input type='text'
name='college_name'/></font><br><br><br><br>
<font size=6 color=white>COLLEGE LOCATION:<input type='text'
name='college_loc'/></font><br><br><br><br>
<input type='submit' value='ENTER'/><br><br><br>
<br><br><br><br><br><br><br><br><br>
<blockquote><footer><i><center><font size=7 color=white>"A mind is a fire to be kindled, not a vessel to be filled."
</font></center></i></footer></blockquote>
</form>
</body>
</html>
```

6.CourseForm.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<img src="https://img.jakpost.net/c/2019/03/02/2019_03_02_66706_1551461528._large.jpg"
style="z-index:-1;position:fixed;width:100%;height:100%"/>
<h1><b><center><font size=7 color=black>COURSE DETAILS</font></center></b><br></h1>
```

```
<form action="Course" method="post"><br><br><br>
<font size=6 color=black>Course Name:<input type='text'
name='course_name'/></font><br><br><br>
<font size=6 color=black>Well Known Language:<input type='text'
name='well_known_language'/></font><br><br><br>
<font size=6 color=black><input type='submit' value='ENTER'/>
<br><br><br><br><br>
<blockquote><footer><i><center><font size=7 color=black>"An investment in knowledge always
pays the best interest."
</font></center></i></footer></blockquote>
</form>
</body>
</html>
```

7.VisionForm.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<img src="https://msp7l1170302145284.blob.core.windows.net/ms-p7-l1-170302-1453-24-
assets/1920_Panel1_FullBleed_Students.jpg" style="z-index:-
1;position:fixed;width:100%;height:100%"/>
<h1><b><center><font size=7 color=white>VISION</font></center></h1><br><br>
<form action="Vision" method="post"><br><br>
<font size=6 color=BLACK>SELECT STUDENT USN :<input type='text'
name='usn'/></font><br><br>
<input type='submit' value='SELECT'/>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br
>
<blockquote><footer><i><center><font size=7 color=BLACK>"Dont let what you cannot do
interfere with what you can do."</font></center></i></footer></blockquote>
</form>
</body>
</html>
```

1.College.java

```
package com.test1;


import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;


import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


/**
 * Servlet implementation class Division
 */
@WebServlet("/College")
public class College extends HttpServlet {
	private static final long serialVersionUID = 1L;


    /**
     * @see HttpServlet#HttpServlet()
     */
    public College() {
        super();
        // TODO Auto-generated constructor stub
    }
	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		response.getWriter().append("Served at: ").append(request.getContextPath());
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		try {
			response.setContentType("text/html");
			PrintWriter out = response.getWriter();

			String a=request.getParameter("college_name");
			String b=request.getParameter("college_loc");

			Class.forName("com.mysql.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/project","root","root");

			PreparedStatement ps=con.prepareStatement(
			"insert into college values(?,?)");

			ps.setString(1,a);
			ps.setString(2,b);
```

```
                              int i=ps.executeUpdate();

                  RequestDispatcher rd = request.getRequestDispatcher("CourseForm.html");
                  rd.forward(request, response);

                              out.close();

                              }catch (Exception e2) {System.out.println(e2);}



          }

}
```

2.Course.java

```
package com.test1;



import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
/**
 * Servlet implementation class Course
 */
@WebServlet("/Course")
public class Course extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Course() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		response.getWriter().append("Served at: ").append(request.getContextPath());
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		try {
			response.setContentType("text/html");
```

```
                                PrintWriter out = response.getWriter();


                                String a=request.getParameter("course_name");
                                String b=request.getParameter("well_known_language");


                                Class.forName("com.mysql.jdbc.Driver");
                                Connection con=DriverManager.getConnection(
                                "jdbc:mysql://localhost:3306/project","root","root");


                                PreparedStatement ps=con.prepareStatement(
                                "insert into course values(?,?)");




                                ps.setString(1,a);
                                ps.setString(2,b);


                                int i=ps.executeUpdate();


                        RequestDispatcher rd = request.getRequestDispatcher("InternshipForm.html");
                        rd.forward(request, response);


                                out.close();


                                }catch (Exception e2) {System.out.println(e2);}




                }


}
```

3.Instructor.java

```
package com.test1;


import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;


import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class Commander
 */
@WebServlet("/Instructor")
public class Instructor extends HttpServlet {
	private static final long serialVersionUID = 1L;


    /**
     * @see HttpServlet#HttpServlet()
     */
    public Instructor() {
        super();
        // TODO Auto-generated constructor stub
    }


	/**
```

```
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		response.getWriter().append("Served at: ").append(request.getContextPath());
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		try {
			String code="internship";
			response.setContentType("text/html");
			PrintWriter out = response.getWriter();

			int a=Integer.parseInt(request.getParameter("instructor_numb"));
			String b=request.getParameter("instructor_name");
			String c=request.getParameter("security_clearence");

			Class.forName("com.mysql.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/project","root","root");

			PreparedStatement ps=con.prepareStatement("insert into instructor
values(?,?,?)");

			ps.setInt(1,a);
			ps.setString(2,b);
			ps.setString(3,c);
```

```
                              int i=ps.executeUpdate();
                              if(c.equals(code))
                {
                  RequestDispatcher rd = request.getRequestDispatcher("CollegeForm.html");
                  rd.forward(request, response);
                }
                              else
                              {
                                        RequestDispatcher rd = request.getRequestDispatcher("Home.html");
                            rd.forward(request, response);
                              }
                              out.close();

                              }catch (Exception e2) {System.out.println(e2);}

          }

}
```

4.Internship.java

```
package com.test1;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;


/**

 * Servlet implementation class Internship

 */

@WebServlet("/Internship")

public class Internship extends HttpServlet {

	private static final long serialVersionUID = 1L;


    /**

     * @see HttpServlet#HttpServlet()

     */

    public Internship() {

        super();

        // TODO Auto-generated constructor stub

    }


	/**

	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

	 */

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

		// TODO Auto-generated method stub

		response.getWriter().append("Served at: ").append(request.getContextPath());

	}


	/**
```

```
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		 try {
			response.setContentType("text/html");
			PrintWriter out = response.getWriter();

			String a=request.getParameter("domain");
			String b=request.getParameter("project_topic");

			Class.forName("com.mysql.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/project","root","root");

			PreparedStatement ps=con.prepareStatement(
			"insert into internship values(?,?)");

			ps.setString(1,b);
			ps.setString(2,b);

			int i=ps.executeUpdate();

			RequestDispatcher rd = request.getRequestDispatcher("Home.html");
		rd.forward(request, response);

			out.close();

			}catch (Exception e2) {System.out.println(e2);}
```

```
            }

}
```

5.StoreData.java

```
package com.test1;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class storeData
 */
@WebServlet("/storeData")
public class storeData extends HttpServlet {
            private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public storeData() {
        super();
        // TODO Auto-generated constructor stub
    }


        /**
         * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                // TODO Auto-generated method stub
                response.getWriter().append("Served at: ").append(request.getContextPath());
        }


        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                try {
                response.setContentType("text/html");
                PrintWriter out = response.getWriter();


                int m=Integer.parseInt(request.getParameter("usn"));
                String n=request.getParameter("name");
                int o=Integer.parseInt(request.getParameter("join_yr"));
                int p=Integer.parseInt(request.getParameter("pass_yr"));
                String q=request.getParameter("choice0");
                String r=request.getParameter("choice1");
                String s=request.getParameter("choice2");
                String t=request.getParameter("choice3");
```

```
                    String u=request.getParameter("choice4");
                    String v=request.getParameter("choice5");

                    Class.forName("com.mysql.jdbc.Driver");
                    Connection con=DriverManager.getConnection(
                    "jdbc:mysql://localhost:3306/project","root","root");

                    PreparedStatement ps=con.prepareStatement(
                    "insert into student values(?,?,?,?,?,?,?,?,?,?)");

                      ps.setInt(1,m);
                          ps.setString(2,n);
                          ps.setInt(3,o);
                          ps.setInt(4,p);
                          ps.setString(5,q);
                          ps.setString(6,r);
                          ps.setString(7,s);
                          ps.setString(8,t);
                          ps.setString(9,u);
                          ps.setString(10,v);

                         int i=ps.executeUpdate();
                         if(i>0)
                         {
                              out.println("<html>"
                                        + "<img
src=\"https://i0.wp.com/www.awakenthegreatnesswithin.com/wp-content/uploads/2018/10/God-
Quotes-1.jpg\" style=\"z-index:-1;position:fixed;width:100%;height:100%\"/>"
                                        +" <h1><b><center><font size=7
color=black>SUCCESS PAGE</font></center></h1><br><br>");
                              out.println("<font size=6 color=white>Successfully
Registered</font><br><br><br><br>");
```

```
                                        out.println("<html><form method=post action=Home.html>"
                                                        + "<input type=submit name=submit
value=Back></form>"

        +"<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br
><br><br><br><br>"
                                                        +"<blockquote><footer><i><center><font size=7
color=white>'The harder you work for something, the greater you'll feel when you achieve
it.'\r\n</font></center></i></footer></blockquote>"
                                                        + "</html>");
                                }

                        } catch (SQLException | ClassNotFoundException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
        }
}
```

6.Student.java

```
package com.test1;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Student
 */
@WebServlet("/Student")
public class Student extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Student() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
	{
		response.setContentType("text/html");
		PrintWriter ps = response.getWriter();

		try
		{
			Class.forName("com.mysql.jdbc.Driver");
```

```
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "root");

            String sql ="select * from college";

            Statement st = con.createStatement();

            ps.println("<html>"

                    + "<img
src=\"https://www.quickanddirtytips.com/sites/default/files/images/15096/how-to-get-a-dream-job-
compressor.png\" style=\"z-index:-1;position:fixed;width:100%;height:100%\"/>"

                    +" <h1><b><center><font size=7 color=black>STUDENT
REGISTRATION</font></center></b><br></h1>");

            ps.println("<form method=post action=storeData>"

                    +

                    "<font size=6 color=black>USN:<input type='text'
name='usn'/>\r\n" +

                    "<font size=6 color=black>STUDENT NAME:<input
type='text' name='name'/>\r\n" +

                    "<font size=6 color=black>JOINED YEAR:<input
type='text' name='join_yr'/>\r\n" +

                    "<font size=6 color=black>PASSING YEAR:<input
type='text' name='pass_yr'/></font><br>"

                    + ""

                    + "<h1><font size=6 color=black>College
Name</h1><select name=choice0></font>");

            ResultSet rs = st.executeQuery(sql);

            while(rs.next())

            {


                ps.print("<option>"+rs.getString(1)+"</option>");

            }

            ps.println("</select>");


            String sql1 ="select * from college";

            Statement st1 = con.createStatement();
```

```
                              ps.println("<h1><font size=6 color=black>College Location</h1><select
name=choice1></font>");
                              ResultSet rs1 = st1.executeQuery(sql1);
                              while(rs1.next())
                              {

                                        ps.print("<option>"+rs1.getString(2)+"</option>");
                              }
                              ps.println("</select>");

                              String sql2 ="select * from course";
                              Statement st2 = con.createStatement();
                              ps.println("<h1><font size=6 color=black>Course Name</h1><select
name=choice2></font>");
                              ResultSet rs2 = st2.executeQuery(sql2);
                              while(rs2.next())
                              {

                                        ps.print("<option>"+rs2.getString(1)+"</option>");
                              }
                              ps.println("</select>");

                              String sql3 ="select * from course";
                              Statement st3 = con.createStatement();
                              ps.println("<h1><font size=6 color=black>Well Known
Language</h1><select name=choice3></font>");
                              ResultSet rs3 = st3.executeQuery(sql3);
                              while(rs3.next())
                              {

                                        ps.print("<option>"+rs3.getString(2)+"</option>");
                              }
```

```
                        ps.println("</select>");


                        String sql4 ="select * from internship";
                        Statement st4 = con.createStatement();
                        ps.println("<h1><font size=6 color=black>Domain</h1><select
name=choice4></font>");
                        ResultSet rs4 = st4.executeQuery(sql4);
                        while(rs4.next())
                        {

                                ps.print("<option>"+rs4.getString(1)+"</option>");
                        }
                        ps.println("</select>");


                        String sql5 ="select * from internship";
                        Statement st5 = con.createStatement();
                        ps.println("<h1><font size=6 color=black>Project Topic</h1><select
name=choice5></font>");
                        ResultSet rs5 = st5.executeQuery(sql5);
                        while(rs5.next())
                        {

                                ps.print("<option>"+rs5.getString(2)+"</option>");
                        }



                        ps.println("</select><br><br><input type=submit name=submit
value=Register></form></html>");


                }
                catch(Exception e)
                {
```

```
                              e.printStackTrace();
                    }
          }
}
```

7.Vision.java

```
package com.test1;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Vision
 */
@WebServlet("/Vision")
public class Vision extends HttpServlet {
          private static final long serialVersionUID = 1L;
```

```
    /**
     * @see HttpServlet#HttpServlet()
     */
    public Vision() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		response.getWriter().append("Served at: ").append(request.getContextPath());
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		try
		{
			response.setContentType("text/html");
			PrintWriter out = response.getWriter();

			Class.forName("com.mysql.jdbc.Driver");
			Connection con=DriverManager.getConnection(
			"jdbc:mysql://localhost:3306/project","root","root");
```

```
                        int a=Integer.parseInt(request.getParameter("usn"));


                        PreparedStatement ps=con.prepareStatement("select * from student where
usn=?");
                        ps.setInt(1,a);
                        ResultSet rs=ps.executeQuery();
                        while(rs.next())
                        {
                                out.println("<html>"
                                                +"<img
src='https://fjwp.s3.amazonaws.com/blog/wp-content/uploads/2019/10/09103114/Career-Change-30-
1.png' style='z-index:-1;position:fixed;width:100%;height:100%'/>"
                                                +" <h1><b><center><font size=7
color=black>STUDENT DETAILS</font></center></h1><br><br><font size=6
color=black>STUDENT USN= ");
                                out.println(rs.getInt(1)+"<br>STUDENT NAME=
"+rs.getString(2)+"<br>JOINING YEAR=  "+rs.getInt(3)+"<br>PASS YEAR=
"+rs.getInt(4)+"<br>COLLEGE NAME= "+rs.getString(5)+"<br>COLLEGE LOCATION=
"+rs.getString(6)+"<br>COURSE NAME= "+rs.getString(7)+"<br>WELL KNOWN LANGUAGE=
"+rs.getString(8)+"<br>DOMAIN= "+rs.getString(9)+"<br>PROJECT TOPIC= "+rs.getString(10));
                                out.println("<br><form method=post action=Home.html></font>"
                                                + "<input type=submit name=submit
value=Back></form>"
                                                +"<br><br><blockquote><footer><i><center><font
size=7 color=black>\"Work to become, not to acquire.\r\n" +
                                                "\"\r\n" +
                                                "</font></center></i></footer></blockquote>"
                                                + "</html>");
                        }


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

```
        }
}
```

5.7.2 User Interface and Functionality Implementation

If the user enters some data, the system must fetch the input and it must take some response through displaying a message on the screen relating to the input or opening a new page according to the input entered by the user. This is known as User Interface.

For example: The user enters the id and password in the registration form, if the data entered is valid, then the user dashboard will be opened. If the data is proper or invalid the system must desclaim some messages saying that “entered data is invalid”.

To check the validity of the data that are entered by the user, there are some functions used in this project.

It fetches the data and checks whether the data is valid or not.

SCREENSHOTS:

1. HOME PAGE

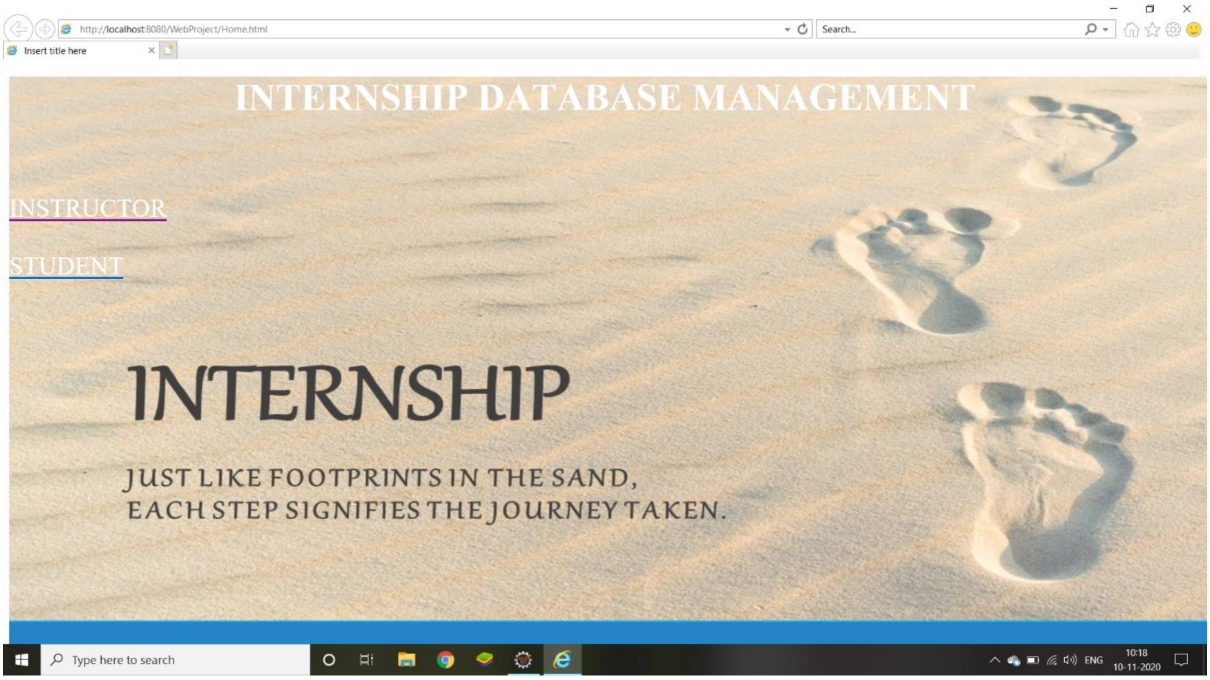
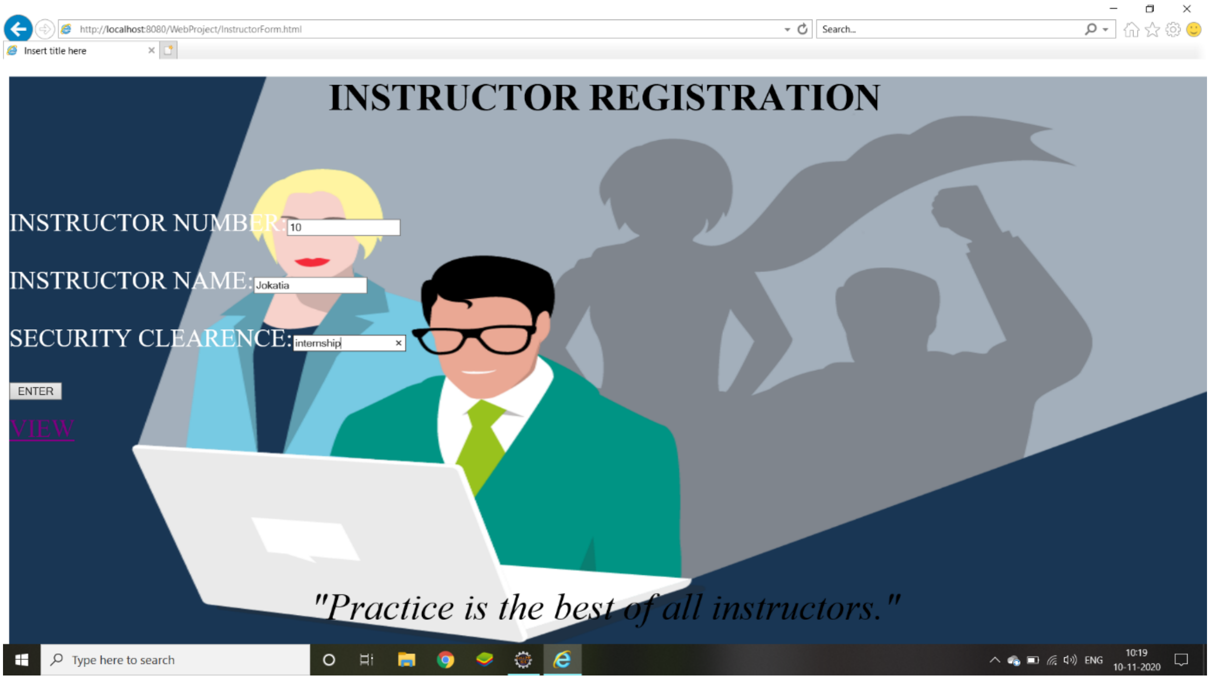

http://localhost:8080/WebProject/Home.html
Search...
Insert title here
INTERNSHIP DATABASE MANAGEMENT
INSTRUCTOR
STUDENT
INTERNSHIP
JUST LIKE FOOTPRINTS IN THE SAND,
EACH STEP SIGNIFIES THE JOURNEY TAKEN.
Type here to search
ENG
10:18
10-11-2020

Fig -5.7: Home page

2. INSTRUCTOR REGISTRATION



INSTRUCTOR REGISTRATION

INSTRUCTOR NUMBER: 10

INSTRUCTOR NAME: Jokatia

SECURITY CLEARENCE: internship

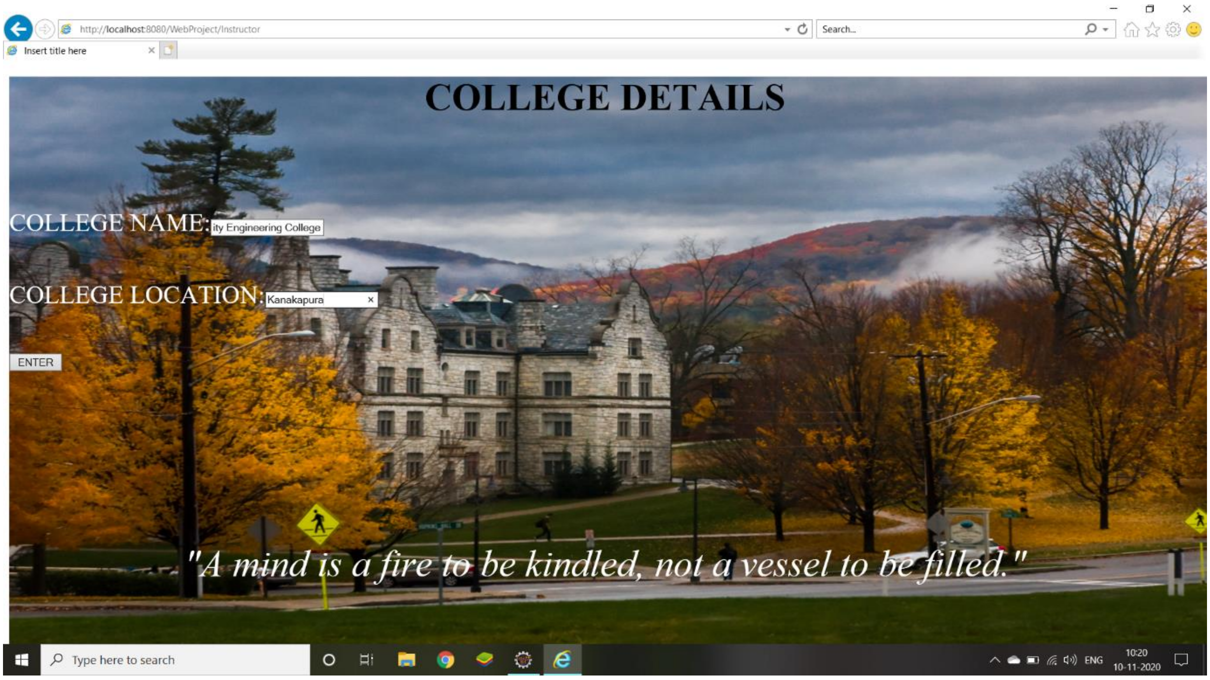
ENTER

VIEW

"Practice is the best of all instructors."

Fig-5.8: Commander Registration

3. COLLEGE DETAILS



COLLEGE DETAILS

COLLEGE NAME: ity Engineering College

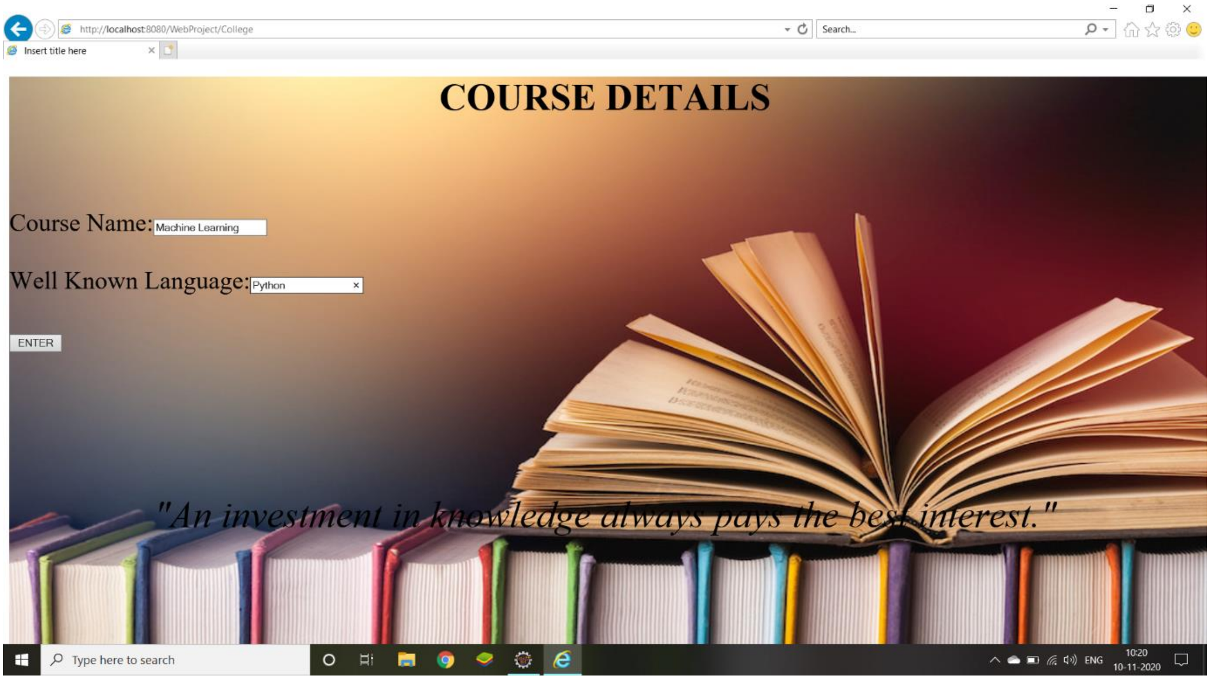
COLLEGE LOCATION: Kanakapura

ENTER

"A mind is a fire to be kindled, not a vessel to be filled."

Fig-5.9: College Details

4. COURSE DETAILS



http://localhost:8080/WebProject/College

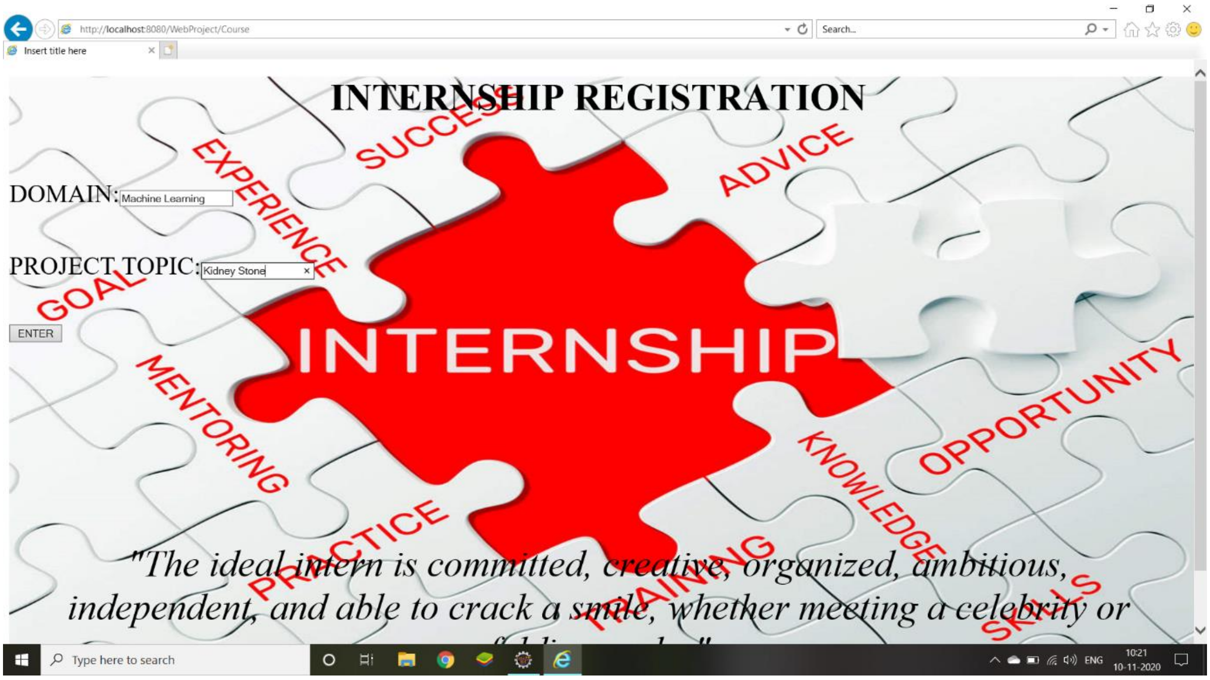
COURSE DETAILS

Course Name: Machine Learning

Well Known Language: Python

Fig-5.10: Course Details

5. INTERNSHIP REGISTRATION



http://localhost:8080/WebProject/Course

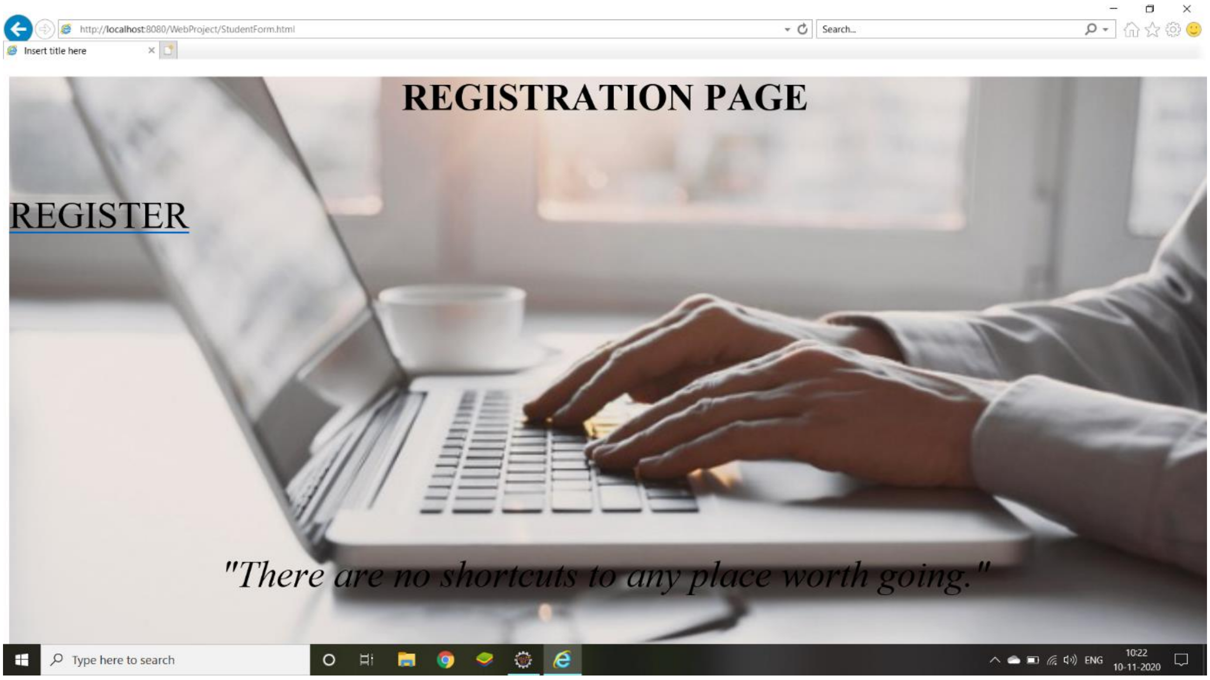
INTERNSHIP REGISTRATION

DOMAIN: Machine Learning

PROJECT TOPIC: Kidney Stone

Fig-5.11: Internship Registration

6. REGISTRATION PAGE



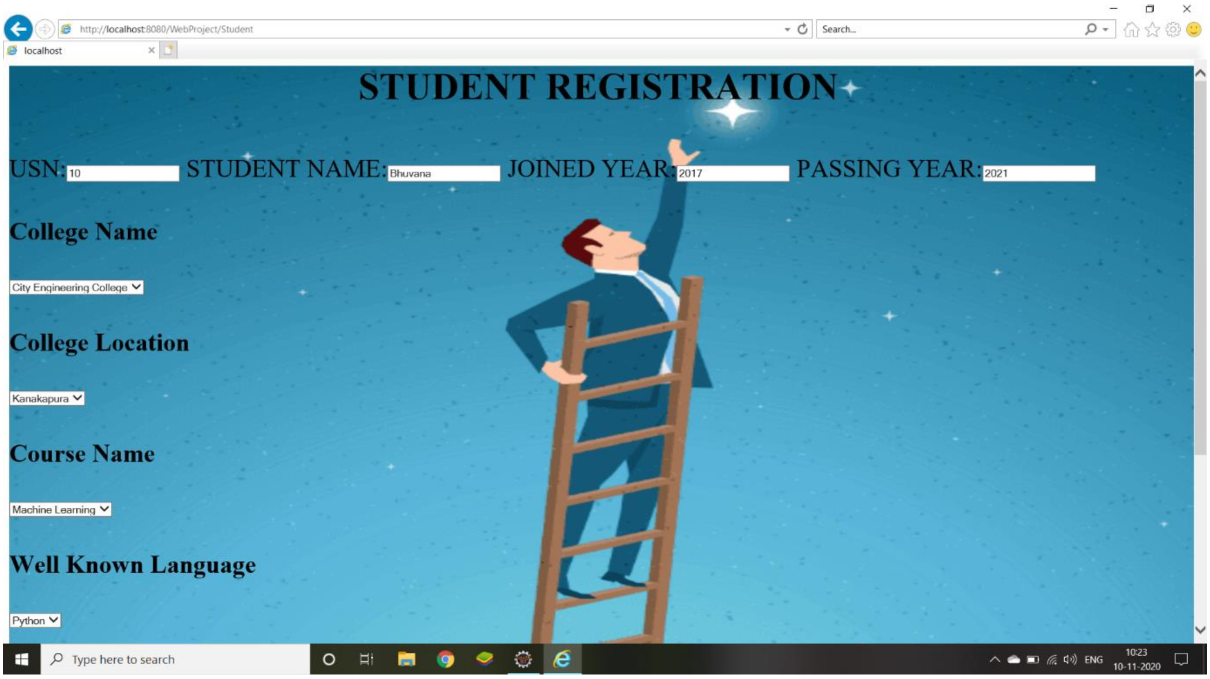
REGISTRATION PAGE

REGISTER

"There are no shortcuts to any place worth going."

Fig-5.12: Registration Page

7. STUDENT REGISTRATION



STUDENT REGISTRATION

USN: 10 STUDENT NAME: Bhuvana JOINED YEAR: 2017 PASSING YEAR: 2021

College Name

City Engineering College

College Location

Kanakapura

Course Name

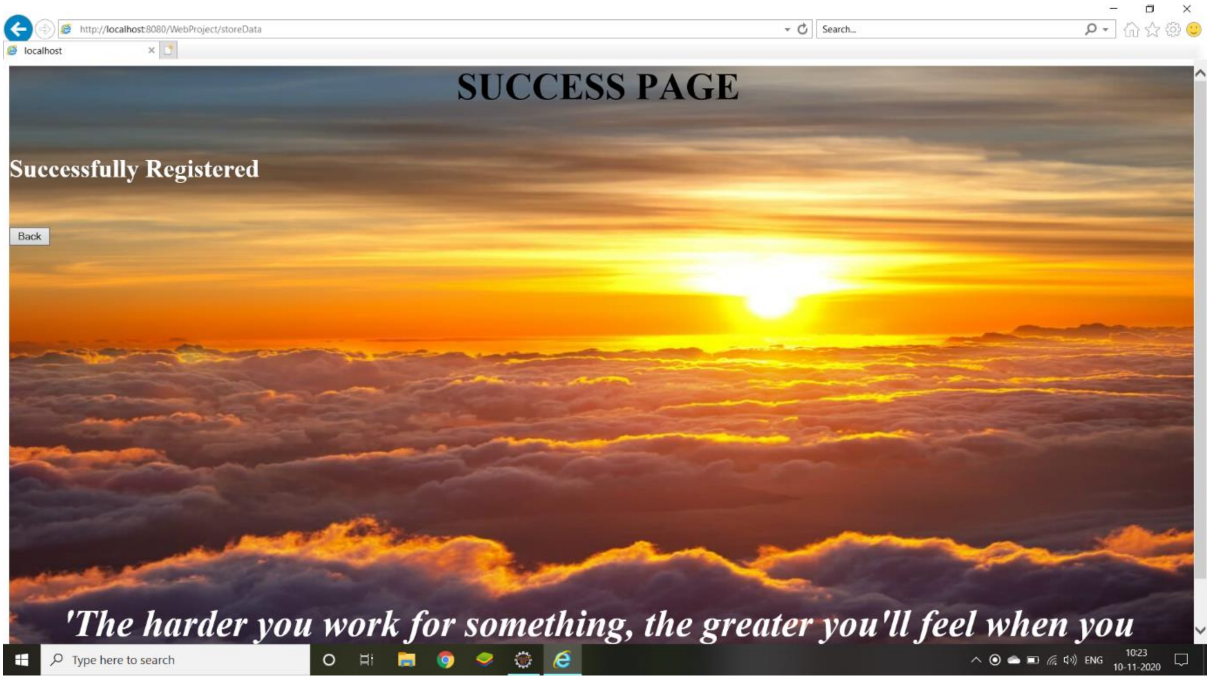
Machine Learning

Well Known Language

Python

Fig-5.14: Student Registration Page

8. SUCCESS PAGE



http://localhost:8080/WebProject/storeData

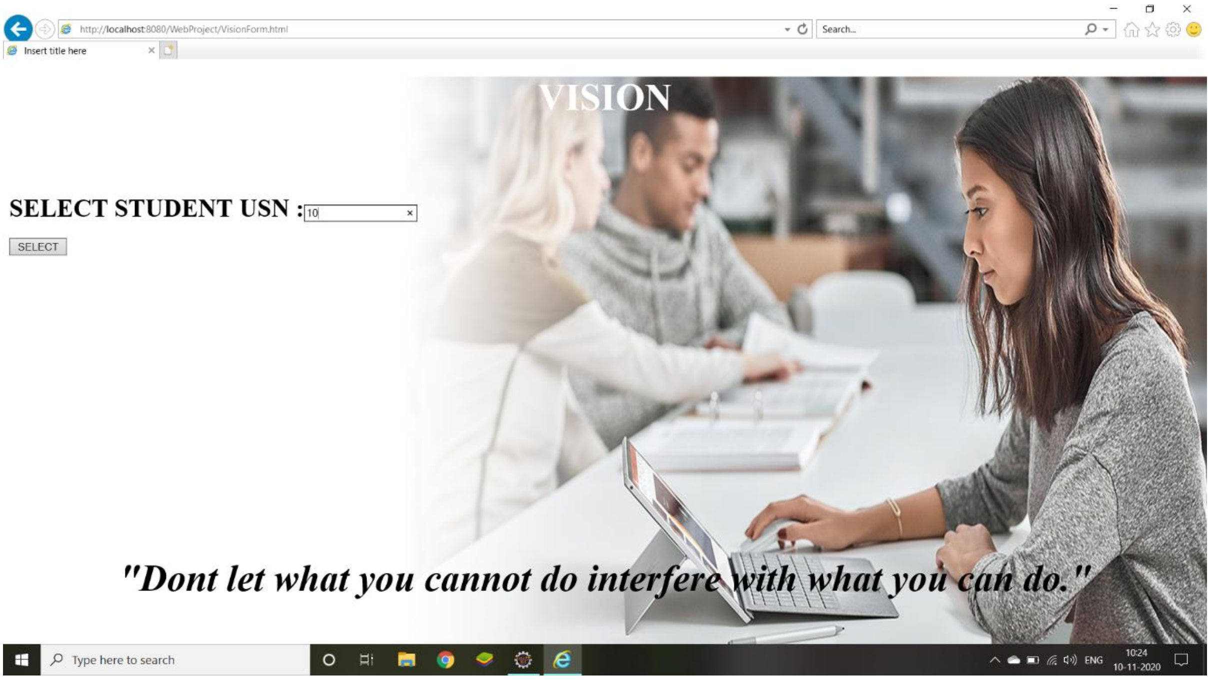
SUCCESS PAGE

Successfully Registered

'The harder you work for something, the greater you'll feel when you

Fig-5.14: Success Page

9. STUDENT DETAILS PAGE



http://localhost:8080/WebProject/VisionForm.html

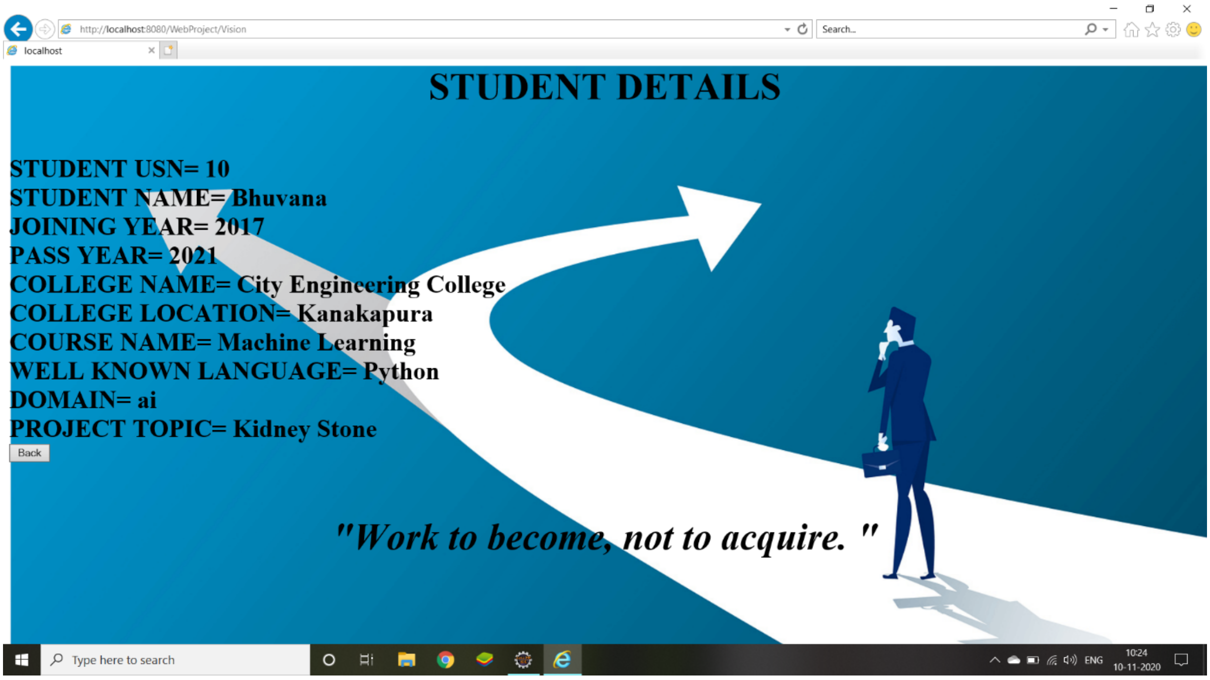
VISION

SELECT STUDENT USN : 10

"Dont let what you cannot do interfere with what you can do."

Fig-5.15: Student Details Page

10. STUDENT DETAILS



STUDENT DETAILS

STUDENT USN= 10
STUDENT NAME= Bhuvana
JOINING YEAR= 2017
PASS YEAR= 2021
COLLEGE NAME= City Engineering College
COLLEGE LOCATION= Kanakapura
COURSE NAME= Machine Learning
WELL KNOWN LANGUAGE= Python
DOMAIN= ai
PROJECT TOPIC= Kidney Stone

"Work to become, not to acquire. "

Fig-5.16: Student Details

CONCLUSION

To conclude, the system is designed to be a useful replacement to the traditional manual record entry that has prevailed. It provides a user with a Intuitive user interface which makes it easy for us to go about the daily job without engaging trivialities like maintaining a record book or updating the columns.

This project achieves the important goal for centralizing all transactions, thus making it easy for the armed forces to keep records.

From a proper analysis of positive points and constraints on the project, it can be concluded that the application makes entire process online. It is highly efficient UI based component which meets all user requirements.

BIBLIOGRAPHY

1) J2EE: The Complete Reference By James Keogh

2) www.w3scholls.com : link-references about HTML

3) “Learning-SQL” a reference book by Alan Beaulieu

DECLARATION

We student of 7th semester BE, Computer Science and Engineering College hereby declare that project work entitled “Internship Database Management System” has been carried out by us at City Engineering College, Bengaluru and submitted in partial fulfillment of the course requirement for the award of the degree of **Bachelor of Engineering in Computer Science and Engineering of Vivesvaraya Technological University, Belgaum**, during the academic year 2019-2020.

We also declare that, to the best of our knowledge and belief, the work reported here does not from the part of dissertation on the basis of which a degree or award was conferred on a earlier occasion on this by any other student.

Date**:**

Place: Bangalore

AKSHAY AMRUT MORAB
(1CE17CS008)

BHUVANESHWARI M
(1CE17CS024)