**ONLINE COLLEGE YEAR BOOK SYSTEM**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** |  |
|  | **INTRODUCTION** |  |
|  | * 1. General Introduction |  |
|  | * 1. Project Objectives |  |
|  | * 1. Problem Statement |  |
|  | **SYSTEM** **PROPOSAL** |  |
|  | * 1. Existing System |  |
|  | * + 1. Advantages |  |
|  | * 1. Proposed System |  |
|  | 2.2.1 Disadvantages |  |
|  | * 1. Related Works |  |
|  | **SYSTEM DIAGRAMS** |  |
|  | * 1. Architecture Diagram |  |
|  | * 1. Flow Diagram |  |
|  | * 1. UML Diagrams |  |
|  | **IMPLEMENTATION** |  |
|  | * 1. Modules |  |
|  | * 1. Modules Description |  |
|  | **SYSTEM** **REQUIREMENTS** |  |
|  | * 1. Hardware Requirements |  |
|  | * 1. Software Requirements |  |
|  | * 1. Software Description |  |
|  | * 1. Testing of Products |  |
|  | **CONCLUSION** **AND** **FUTURE** **ENHANCEMENT** |  |
|  | * 1. Conclusion |  |
|  | * 1. Future Enhancement |  |
|  | **SAMPLE** **CODING** **AND** **SAMPLE** **SCREENSHOT** |  |
|  | **REFERENCES** |  |

**ABSTRACT**

A yearbook, also known as an annual, is a type of a book published annually. One use is to record, highlight, and commemorate the past year of a school or college. The term also refers to a book of statistics or facts published annually this book was improve college or school worth. We create website for yearbook because all people easily access our yearbook anytime any ware and improve college worth hypertext Mark-up Language, is used to create web pages. Site authors useHTML to format text as titles and headings, to arrange graphics on a webpage, to link to different pages within a website, and to link to different websites.MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications. The most common use for MySQL however, is for the purpose of a web database.

1. **INTRODUCTION**

**1.1** **GENERAL INTRODUCTION**

A yearbook, also known as an annual, is a type of [a book published annually](https://en.wikipedia.org/wiki/Annual_publication). One use is to record, highlight, and commemorate the past year of a [school](https://en.wikipedia.org/wiki/School). The term also refers to a book of statistics or facts published annually. A yearbook often has an overarching theme that is present throughout the entire book.

Many [high schools](https://en.wikipedia.org/wiki/Secondary_education), [colleges](https://en.wikipedia.org/wiki/College), and [elementary](https://en.wikipedia.org/wiki/Primary_school) and [middle schools](https://en.wikipedia.org/wiki/Middle_school) publish yearbooks; however, many schools are dropping yearbooks or decreasing page counts given social media alternatives to a mass-produced physical photographically-oriented record.

India does not have a long history of publishing school yearbooks. However, top Business schools and Engineering colleges publish custom yearbooks. This is typically created by the final year students of the batch. A yearbook or a memory book would consist of testimonials and common pages such as Directors address and events, festivals picture collages.

Most top schools do create schools magazines which are shared with each student. Some of the early adopters among school students are starting to create custom yearbooks in the same line as created by students from US or Europe. This trend is likely to pick up with the advent of technology platforms that make it easy for students to create them.

Yearbooks are generally compiled by a student club or a yearbook class, usually advised by a faculty member. The yearbook staff usually has one or more editors who are responsible for collecting and compiling all of the information to be contained within the book, also deciding the layout and allocation of space for each contributor.

Several pages are often used for pages chronicling activities undertaken by students, such as trips abroad, activity trips, sporting and other special events. This part of the book often covers students' lives both inside and outside of the campus.

Sometimes members of a yearbook write editorial and journalistic content about life as a student, current events (local, national, and international), and other matters of interest to the peer group.

This section describes student organizations (sometimes referred to as clubs) and what they did during the year. These descriptions are often accompanied by a photo or photos of the organizations' members. This section sometimes includes a list of the members of each organization.

Yearbook companies that use off-set printing require that groups of pages be sent periodically, rather than all at once, to the plant. This is done to stagger the work required to complete yearbooks for all the schools they cover. After the editors review each page and make changes, the pages are sent to the yearbook plant, usually via the Internet. Yearbook companies that use digital printing methods may only require one submission since the entire book is printed at once.

If the proofing process is not performed on-line, the adviser and editors receive proofs (typically full size prints) about a week or so after the submission of pages. This gives the school a final opportunity to make adjustments or changes. After all the proofs have been returned to the printing company the requested corrections are made, the books are printed, bound, and then sent to the school for distribution. Two examples of printing companies include Balfour and Jostens.

A number of educational institutions and yearbook printing companies operate camps and summer sessions at which student yearbook personnel can learn or polish the skills they need in their work.

A digital yearbook or eYearbook is a yearbook holding memories of a given time with a given group of people—most commonly, a school year at a particular school—that exists in digital form.

A digital yearbook may contain text, images, audio, and video. While a traditional paper yearbook may contain 300+ pages, a digital yearbook can contain unlimited pages.

Forever Connected created the first widely-adopted interactive, mobile yearbook, based on the print edition. Students can sign, sticker and send video to classmates right from their mobile devices. It is the most widely used digital yearbook and was originally sold by traditional yearbook publishers and as add-on to a print purchase. In 2019 Forever Connected rebranded as FC Yearbook and began offering their platform directly to schools. A number of patents were awarded on the technology to the parent company Yearbooker, Inc. In 2021 FC Yearbook accommodated remote and in-person student signatures and offered a print edition with the digital signatures printed in the book, as a direct-to-home offer. This became the origin of "virtual yearbook day" where in-person and remote students could simultaneously sign each others' yearbook in a yearbook party. On May 14, 2021 FC Yearbook announced the first print yearbook ordered from a digital yearbook in an app, marking "digital-first" as a unique change in the yearbook commerce model.

* 1. **PROJECT OBJECTIVES**
* The main aim & objective of development of Online College Yearbook java project is to provide an online platform to the college management, Staff and student alumni.
* With the help of this project Students & Staff can able see the tasks & achievements done in previous years.
  1. **PROBLEM STATEMENT**
* Every year, for each institution it is a part of the curriculum to showcase the tasks, achievements completed by students and teachers, manually.
* It is not suitable and chores take longer time to prepare.

1. **SYSTEM PROPOSAL**

**2.1 EXISTING SYSTEM**

* In existing system, Yearbook is done manually in existing system.
* It takes a lot of time while updating each events in year book.
* Manual errors may happen.
* Maintanence of records is a major issue.
* It has lot of loss information.

**2.1.1 DISADVANTAGES**

* Less Secure
* Data loss
* Maintenance is low.
* Cost is high

**2.2 PROPOSED SYSTEM**

* The proposed model All the events can be updated easily using digital manner,
* Everyone can view the details.
* Images, exact information about the events can be viewed in a digital manner.
* Year book is not done manually.

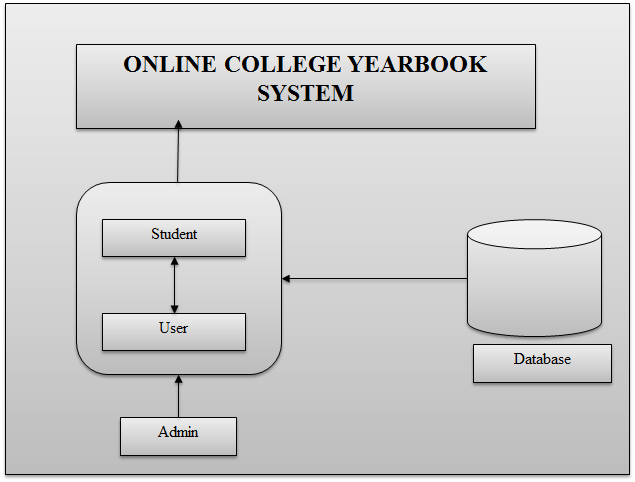
**2.2.1 ADVANTAGES**

* The Year book is prepared in a digital manner.
* Updating the events is done easily.
* Time taken is less compared to manual updation.
* It is very cost efficient.
* Manual errors are avoided.
  1. **RELATED WORKS**

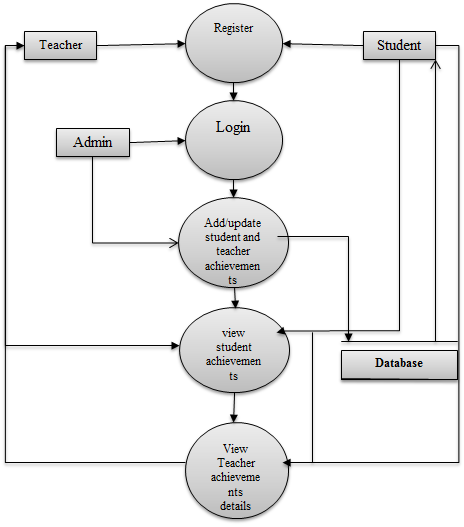
The best research method, according to Ann Hill Duin in her article on usability, involves methodological triangulation using multiple methods to collect data and arrive at a greater understanding of the topic. Accordingly, I investigated my research questions using a combination of methods drawn from qualitative case study methods and rhetorical analysis. The research for this historical case study of production practices for a particular college yearbook incorporated 1) a rich description of yearbook production from 2003-05, and 2) a rhetorical analysis of the spreads and images of the two college yearbooks produced during that period. The description of production relies on materials and documents used in generating the yearbooks as well as a personal interview with a publishing representative and retrospective description of personal experience. The rhetorical analysis of yearbook spreads and images is of two yearbooks from 2004 and 2005, applying the concepts of Kress and van Leuven. The design and implementation of a comprehensive Online College Management System and user interface is to replace the current paper records. College Staff are able to directly access all aspects of a student’s academic progress through a secure, online interface embedded in the college’s website. The system utilizes user authentication, displaying only information necessary for an individual’s duties. Additionally, each sub-system has authentication allowing authorized users to create or update information in that subsystem. All data is thoroughly reviewed and validated on the server before actual record alteration occurs. In addition to a staff user interface, the system plans for student user interface, allowing users to access information and submit requests online thus reducing processing time. All data is stored securely on SQL servers managed by the college administrator and ensures highest possible level of security. The system features a complex logging system to track all users’ access and ensure conformity to data access guidelines and is expected to increase the efficiency of the college’s record management thereby decreasing the work hours needed to access and deliver student records to users. Yearbooks are an important part of an individual’s history. In the United States, high school and college yearbooks are extraordinarily well known as a genre, yet the publication of yearbooks is an area largely unstudied by researchers. Students are introduced to yearbooks at an early age. In elementary and middle school students learn what a yearbook is, and at many schools between 60-80 percent of students purchase a yearbook (Walsh “Preserving Student Memories a $500 Million Industry”). In high school, there is an entire day devoted to yearbook distribution. Students crowd in line to get their copy of the yearbook and find their picture in it. Then, they rush to get all their friends and acquaintances to sign their yearbook—personalizing it and creating a lasting memory of their school year.

1. **SYSTEM DIAGRAMS**

**3.1 ARCHITECTURE DIAGRAM**

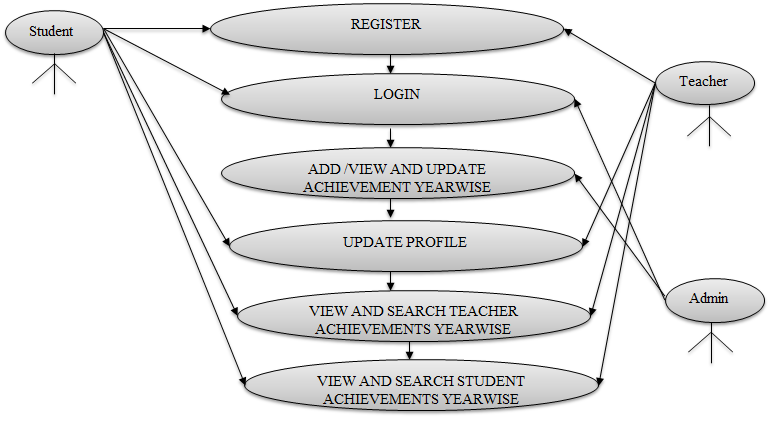


**3.2 FLOW DIAGRAM**

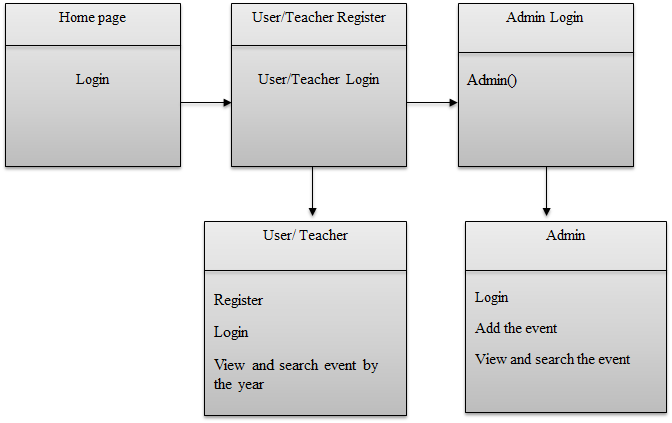


**3.3 UML DIAGRAMS**

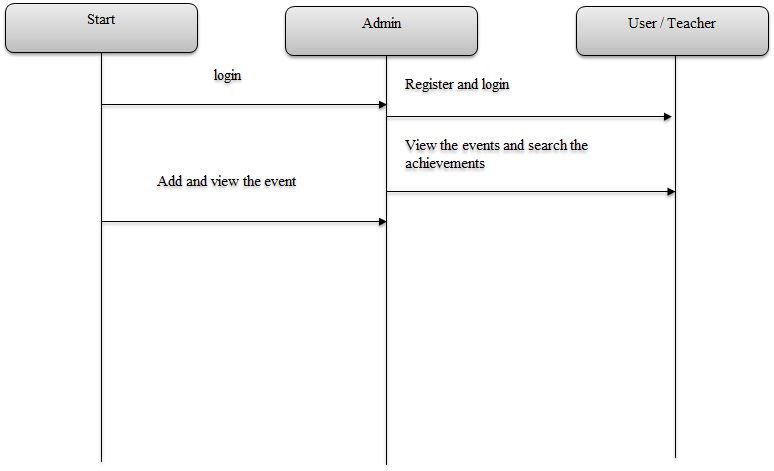
**USE CASE DIAGRAM**



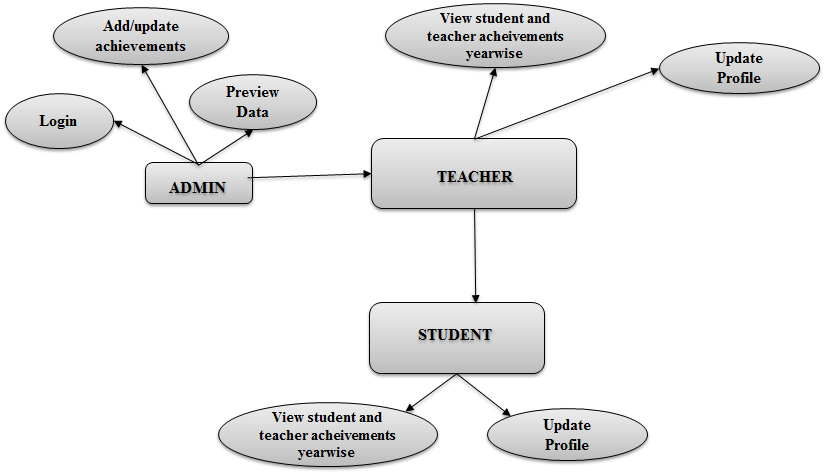
**CLASS DIAGRAM**



**SEQUENCE DIAGRAM**

****

**ER DIAGRAM**

****

1. **IMPLEMENTATION**

**4.1 MODULES**

* Admin
* Teacher
* Students
  1. **MODULE DESCRIPTION**

**ADMIN**

* + Admin can login with their username and password.
  + Then add/update the achievements with the particular details like branch name, experience, year and image.
  + Then admin can view the data and also search with the branch name, year.

**TEACHER:**

* The teacher can register with some details and also login with the username and password that given in register page.
* Teacher can update the profile with their details if they want.
* Teachers can view the yearwise achievements of students and teachers search using year and the branch name.

**STUDENTS:**

* The teacher can register with some details and also login with the username and password that given in register page.
* Teacher can update the profile with their details if they want.
* Teachers can view the yearwise achievements of students and teachers search using year and the branch name.

1. **SYSTEM REQUIREMENTS**

**5.1 Hardware Requirements**

* Processor : Intel Core i5
* Hard Disk : 200 GB
* Monitor : 18’ LED color
* Mouse : DELL.
* Keyboard : 110 keys enhanced
* RAM : 3GB

**5.2 Software Requirements**

* Operating System : Windows 7 / 8 / 10
* Language Used : Java (Servlet, JSP)
* Database : My SQL
* User Interface Design : HTML, CSS , JAVA SCRIPT
* Web Browser : Mozilla, IE8, Chrome
* Server : Glass Fish Server 4.1

**5.3 SOFTWARE DESCRIPTION**

**Java**

Java is a programming language originally developed by James Gosling at Sun Microsystems (now a subsidiary of Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere." Java is currently one of the most popular programming languages in use, particularly for client-server web applications.

**Java Platform**

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware.

End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets. Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executables would. Just-in-Time compilers were introduced from an early stage that compiles byte codes to machine code during runtime.

Just as application servers such as Glass Fish provide lifecycle services to web applications, the Net Beans runtime container provides them to Swing applications. All new shortcuts should be registered in "Key maps/Net Beans" folder. Shortcuts installed INS Shortcuts folder will be added to all key maps, if there is no conflict. It means that if the same shortcut is mapped to different actions in Shortcut folder and current key map folder (like Key map/Net Beans), the Shortcuts folder mapping will be ignored.

* + Database Explorer Layer API in Database Explorer
  + Loaders-text-dB schema-Actions in Database Explorer
  + Loaders-text-sq.-Actions in Database Explorer
  + Plug-in Registration in Java EE Server Registry

The keyword public denotes that a method can be called from code in other classes, or that a class may be used by classes outside the class hierarchy. The class hierarchy is related to the name of the directory in which the .java file is located.

The keyword static in front of a method indicates a static method, which is associated only with the class and not with any specific instance of that class. Only static methods can be invoked without a reference to an object. Static methods cannot access any class members that are not also static. The keyword void indicates that the main method does not return any value to the caller. If a Java program is to exit with an error code, it must call System. Exit () explicitly.

The method name "main" is not a keyword in the Java language. It is simply the name of the method the Java launcher calls to pass control to the program. Java classes that run in managed environments such as applets and Enterprise JavaBeans do not use or need a main () method. A Java program may contain multiple classes that have main methods, which means that the VM needs to be explicitly told which class to launch from.

The Java launcher launches Java by loading a given class (specified on the command line or as an attribute in a JAR) and starting its public static void main(String[]) method. Stand-alone programs must declare this method explicitly. The String [] rags parameter is an array of String objects containing any arguments passed to the class. The parameters to main are often passed by means of a command line.

**Java a High-level Language**

A high-level programming language developed by Sun Microsystems. Java was originally called OAK, and was designed for handheld devices and set-top boxes. Oak was unsuccessful so in 1995 Sun changed the name to Java and modified the language to take advantage of the burgeoning World Wide Web.

Java source code files (files with a .java extension) are compiled into a format called byte code (files with a .class extension), which can then be executed by a Java interpreter. Compiled Java code can run on most computers because Java interpreters and runtime environments, known as Java Virtual Machines (VMs). Byte code can also be converted directly into machine language instructions by a just-in-time compiler (JIT).

Java is a general purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer.

Object-Oriented Software Development using Java: Principles, Patterns, and Frameworks contain a much applied focus that develops skills in designing software-particularly in writing well-designed, medium-sized object-oriented programs. It provides a broad and coherent coverage of object-oriented technology, including object-oriented modeling using the Unified Modeling Language (UML) object-oriented design using Design Patterns, and object-oriented programming using Java.

**Net Beans**

The **Net Beans Platform** is a reusable framework for simplifying the development of Java Swing desktop applications. The Net Beans IDE bundle for Java SE contains what is needed to start developing Net Beans plug-in and Net Beans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally-signed upgrades and new features directly into the running application.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

* User interface management (e.g. menus and toolbars)
* User settings management
* Storage management (saving and loading any kind of data)
* Window management
* Wizard framework (supports step-by-step dialogs)
* Net Beans Visual Library
* Integrated Development Tools

**J2EE**

A **Java EE application** or a **Java Platform, Enterprise Edition application** is any deployable unit of Java EE functionality. This can be a single Java EE module or a group of modules packaged into an EAR file along with a Java EE application deployment descriptor.

Enterprise applications can consist of the following:

* EJB modules (packaged in JAR files)
* Web modules (packaged in WAR files)
* connector modules or resource adapters (packaged in RAR files)
* Session Initiation Protocol (SIP) modules (packaged in SAR files)
* application client modules
* Additional JAR files containing dependent classes or other components required by the application.

**Java Server Pages (JSP)**

* It stands for **Java Server Pages**.
* It is a server side technology.
* It is used for creating web application.
* It is used to create dynamic web content.
* In this JSP tags are used to insert JAVA code into HTML pages.
* It is an advanced version of Servlet Technology.
* It is a Web based technology helps us to create dynamic and platform independent web pages.
* In this, Java code can be inserted in HTML/ XML pages or both.
* JSP is first converted into servlet by JSP container before processing the client’s request.

**JSP pages are more advantageous than Servlet:**

* They are easy to maintain.
* No recompilation or redeployment is required.
* JSP has access to entire API of JAVA.
* JSP are extended version of Servlet.

**Features of JSP**

* **Coding in JSP is easy**: - As it is just adding JAVA code to HTML/XML.
* **Reduction in the length of Code**: - In JSP we use action tags, custom tags etc.
* **Connection to Database is easier**:-It is easier to connect website to database and allows reading or writing data easily to the database.
* **Make Interactive websites**: - In this we can create dynamic web pages which helps user to interact in real time environment.
* **Portable, Powerful, flexible and easy to maintain**: - as these are browser and server independent.
* **No Redeployment and No Re-Compilation**:- It is dynamic, secure and platform independent so no need to re-compilation.
* **Extension to Servlet**: - as it has all features of servlets, implicit objects and custom tags.

**Wamp Server**

**WAMP**s are packages of independently-created programs installed on computers that use a Microsoft Windows operating system.

Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL database manager, or the alternative scripting languages Python or Perl.

**MySQL**

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

Free-software-open source projects that require a full-featured database management system often use MySQL. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, hob, Drupal and other software built on the LAMP software stack.

**Platforms and interfaces**

Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interface called Modoc allows additional programming languages that support the ODBC interface to communicate with a MySQL database, such as ASP or ColdFusion. The MySQL server and official libraries are mostly implemented in ANSI C/ANSI C++.

**FEASIBILITY STUDY**

The feasibility study is carried out to test whether the proposed system is worth being implemented. The proposed system will be selected if it is best enough in meeting the performance requirements.

The feasibility carried out mainly in three sections namely.

**•** Economic Feasibility

• Technical Feasibility

• Behavioural Feasibility

**Economic Feasibility**

Economic analysis is the most frequently used method for evaluating effectiveness of the proposed system. More commonly known as cost benefit analysis. This procedure determines the benefits and saving that are expected from the system of the proposed system. The hardware in system department if sufficient for system development.

**Technical Feasibility**

This study centre around the system’s department hardware, software and to what extend it can support the proposed system department is having the required hardware and software there is no question of increasing the cost of implementing the proposed system. The criteria, the proposed system is technically feasible and the proposed system can be developed with the existing facility.

**Behavioural Feasibility**

People are inherently resistant to change and need sufficient amount of training, which would result in lot of expenditure for the organization. The proposed system can generate reports with day-to-day information immediately at the user’s request, instead of getting a report, which doesn’t contain much detail.

**5.4 TESTING OF PRODUCT**

**Testing of Product**

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system.  System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.  The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test.  A series of tests are performed before the system is ready for the user acceptance testing.  Any engineered product can be tested in one of the following ways.  Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational.  Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

**UNIT TESTING**

Unit testing is the testing of each module and the integration of the overall system is done.  Unit testing becomes verification efforts on the smallest unit of software design in the module.  This is also known as ‘module testing’.  The modules of the system are tested separately.  This testing is carried out during the programming itself.  In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module.  There are some validation checks for the fields.  For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included.  It is very easy to find error and debug the system.

**INTEGRATION TESTING**

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function.  Integrated testing is systematic testing that can be done with sample data.  The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

1. Top-down integration testing.
2. Bottom-up integration testing.

**WHITE BOX TESTING**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases.  Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

**BLACK BOX TESTING**

* Black box testing is done to find incorrect or missing function
* Interface error
* Errors in external database access
* Performance errors
* Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’.  It tests the external behaviour of the system.  Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

**VALIDATION TESTING**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

# USER ACCEPTANCE TESTING

User acceptance of the system is the key factor for the success of the system.  The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

# OUTPUT TESTING

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format.  The output displayed or generated by the system under consideration.  Here the output format is considered in two ways.  One is screen and the other is printed format.  The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs.  For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

**Agile Testing**

Agile Testing is a type of software testing that accommodates agile software development approach and practices. In an Agile development environment, testing is an integral part of software development and is done along with coding. Agile testing allows incremental and iterative coding and testing.

**API Testing**

API testing is a type of testing that is similar to unit testing. Each of the Software APIs are tested as per API specification. API testing is mostly done by testing team unless APIs to be tested or complex and needs extensive coding. API testing requires understanding both API functionality and possessing good coding skills.

**Automated testing**

This is a testing approach that makes use of testing tools and/or programming to run the test cases using software or custom developed test utilities. Most of the automated tools provided capture and playback facility, however there are tools that require writing extensive scripting or programming to automate test cases.

**End-to-end Testing**

End to end testing is performed by testing team, focus of end to end testing is to test end to end flows e.g. right from order creation till reporting or order creation till item return etc. and checking. End to end testing is usually focused mimicking real life scenarios and usage. End to end testing involves testing information flow across applications.

**Exploratory Testing**

Exploratory testing is an informal type of testing conducted to learn the software at the same time looking for errors or application behaviour that seems non-obvious. Exploratory testing is usually done by testers but can be done by other stake holders as well like Business Analysts, developers, end users etc. who are interested in learning functions of the software and at the same time looking for errors or behaviour is seems non-obvious.

**Performance Testing**

**It** is a type of software testing and part of performance engineering that is performed to check some of the quality attributes of software like Stability, reliability, availability. Performance testing is carried out by performance engineering team. Unlike Functional testing, Performance testing is done to check non-functional requirements. Performance testing checks how well software works in anticipated and peak workloads. There are different variations or sub types of performance like load testing, stress testing, volume testing, soak testing and configuration testing.

**Penetration Testing**

**It** is a type of security testing, also known as pen test in short. Penetration testing is done to tests how secure software and its environments (Hardware, Operating system and network) are when subject to attack by an external or internal intruder. Intruder can be a human/hacker or malicious programs. Pen test uses methods to forcibly intrude (by brute force attack) or by using a weakness (vulnerability) to gain access to a software or data or hardware with an intent to expose ways to steal, manipulate or corrupt data, software files or configuration. Penetration Testing is a way of ethical hacking, an experienced Penetration tester will use the same methods and tools that a hacker would use but the intention of Penetration tester is to identify vulnerability and get them fixed before a real hacker or malicious program exploits it.

**Security Testing**

**It** is a type of software testing carried out by specialized team of software testers. Objective of security testing is to secure the software is to external or internal threats from humans and malicious programs. Security testing basically checks, how good is software’s authorization mechanism, how strong is authentication, how software maintains confidentiality of the data, how does the software maintain integrity of the data, what is the availability of the software in an event of an attack on the software by hackers and malicious programs is for Security testing requires good knowledge of application, technology, networking, security testing tools. With increasing number of web applications necessarily of security testing has increased to a greater extent.

**Sanity Testing**

**It** is a type of testing that is carried out mostly by testers and in some projects by developers as well. Sanity testing is a quick evaluation of the software, environment, network, external systems are up & running, software environment as a whole is stable enough to proceed with extensive testing. Sanity tests are narrow and most of the time sanity tests are not documented.

**Scalability Testing**

**It** is a non-functional test intended to test one of the software quality attributes i.e. “Scalability”. Scalability test is not focused on just one or few functionality of the software instead performance of software as a whole. Scalability testing is usually done by performance engineering team. Objective of scalability testing is to test the ability of the software to scale up with increased users, increased transactions, increase in database size etc., It is not necessary that software’s performance increases with increase in hardware configuration, scalability tests helps to find out how much more workload the software can support with expanding user base, transactions, data storage etc.,

**Stability Testing**

**It** is a non-functional test intended to test one of the software quality attributes i.e. “Stability”. Stability testing focuses on testing how stable software is when it is subject to loads at acceptable levels, peak loads, loads generated in spikes, with more volumes of data to be processed. Scalability testing will involve performing different types of performance tests like load testing, stress testing, spike testing, soak testing, spike testing etc.…

**Static Testing** is a form of testing where in approaches like reviews, walkthroughs are employed to evaluate the correctness of the deliverable. In static testing software code is not executed instead it is reviewed for syntax, commenting, naming convention, size of the functions and methods etc. Static testing usually has check lists against which deliverables are evaluated. Static testing can be applied for requirements, designs, and test cases by using approaches like reviews or walkthroughs.

**Stress Testing** is a type of performance testing, in which software is subjected to peak loads and even to a break point to observe how the software would behave at breakpoint. Stress testing also tests the behaviour of the software with insufficient resources like CPU, Memory, Network bandwidth, Disk space etc. Stress testing enables to check some of the quality attributes like robustness and reliability.

1. **CONCLUSION** **AND** **FUTURE** **ENHANCEMENT**
   1. **CONCLUSION**

We conclude that the online yearbook system can be update all the achievements of students and teachers based on the previous year easily using digital manner. Everyone can easily access the webpage with the proper authentication using username and password to enter into the process to update profile and view the achievements of the students and teachers. This process overcomes the manual process and it is converted into digital manner effectively.

* 1. **FUTURE ENHANCEMENT**

In Future, for the students who make the yearbook—their view of information value, salience, and framing could be different than the intended audience’s view. For each of these areas of further study a larger sample of yearbooks would be needed.

1. **SAMPLE** **CODING** **AND** **SAMPLE** **SCREENSHOT**

**CODING**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Online College Year Book</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">

</head>

<body>

<%

if (request.getParameter("m1") != null) {%>

<script>alert('Login Failed..!');</script>

<%}%>

<%

if (request.getParameter("msg") != null) {%>

<script>alert('Registered Successfully..!');</script>

<%}%>

<!-- Main Header -->

<header class="main-header">

<center><h1>Online College Year Book</h1></center>

</header>

<!-- Dropdown Navbar -->

<nav>

<ul>

<li><a href="index.html" >Home</a></li>

<li><a href="login.jsp" class="active">Login</a></li>

<li><a href="register.jsp">Register</a></li>

</ul>

</nav>

<!-- Login Form -->

<div class="form-div animated flip">

<img src="img/user.png">

<h2> Login Page</h2>

<form action="loginact.jsp" method="post">

<input type="text" placeholder="Username" name="username" required="" autocomplete="off">

<br>

<input type="password" placeholder="Password" name="password" required="">

<br>

<select name="utype" id="" required="">

<option value="">--Select--</option>

<option value="Admin">Admin</option>

<option value="Teacher">Teacher</option>

<option value="Student">Student</option>

</select>

<br>

<br>

<input type="submit" value="Login">

<br>

<br>

</form>

</div>

<br><br>

<br><br><br>

<br>

<!-- Main Footer -->

<footer class="main-footer">

<p>Developed by <a href="" target="\_blank"> </a></p>

</footer>

</body>

</html>

<%@page import="java.sql.\*"%>

<%@ include file="connect.jsp" %>

<%@ page session="true" %>

<%

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

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

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

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

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

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

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

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

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

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

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

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

try{

PreparedStatement ps = connection.prepareStatement("insert into teacherevents(username,designation,yr,branch,dob,qualificaion,email,mobile,exp1,specialization,subjects,image) values(?,?,?,?,?,?,?,?,?,?,?,?)");

ps.setString(1,username);

ps.setString(2,designation);

ps.setString(3,year);

ps.setString(4,branch);

ps.setString(5,dob);

ps.setString(6,qualificaion);

ps.setString(7,email);

ps.setString(8,mobile);

ps.setString(9,exp);

ps.setString(10,specialization);

ps.setString(11,subjects);

ps.setString(12,image);

ps.executeUpdate();

response.sendRedirect("addtachievements.jsp?msg=success");

}

catch(Exception e1)

{

response.sendRedirect("addtachievements.jsp?m1=Failed");

out.println(e1.getMessage());

}

%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Online College Year Book</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">

</head>

<body>

<%

if (request.getParameter("msg") != null) {%>

<script>alert('Success..!');</script>

<%}%>

<%

if (request.getParameter("m1") != null) {%>

<script>alert('Failed..!');</script>

<%}%>

<!-- Main Header -->

<header class="main-header">

<center><h1>Online College Year Book</h1></center>

</header>

<!-- Dropdown Navbar -->

<nav>

<ul>

<li><a href="adminhome.jsp">Home</a></li>

<li><a href="teacherachievements.jsp">Teacher Achievements</a></li>

<li><a href="studentachievements.jsp" class="active">Student Achievements</a></li>

<li><a href="logout.jsp">Logout</a></li>

</ul>

</nav>

<!-- Login Form -->

<div class="form-div animated flip">

<img src="img/user.png">

<h2>Add Achievements</h2>

<form action="addsachievementsact.jsp" method="post">

<br>

<select name="year" id="" required="">

<option value="">--Select Year--</option>

<option value="2015">2015</option>

<option value="2016">2016</option>

<option value="2017">2017</option>

<option value="2018">2018</option>

<option value="2019">2019</option>

<option value="2020">2020</option>

<option value="2021">2021</option>

<option value="2022">2022</option>

<option value="2023">2023</option>

<option value="2024">2024</option>

<option value="2025">2025</option>

<option value="2026">2026</option>

<option value="2027">2027</option>

<option value="2028">2028</option>

<option value="2029">2029</option>

<option value="2030">2030</option>

</select>

<br>

<input type="text" placeholder="Roll No" name="rno" required="" autocomplete="off">

<br>

<input type="text" placeholder="Name" name="username" required="" autocomplete="off">

<br>

<select name="branch" id="" required="">

<option value="">--Select Branch--</option>

<option value="CSE">CSE</option>

<option value="IT">IT</option>

<option value="Automobile">Automobile</option>

<option value="Civil">Civil</option>

<option value="Mech">Mech</option>

<option value="ECE">ECE</option>

<option value="EEE">EEE</option>

</select>

<br>

<input type="text" name="dob" required="" autocomplete="off" placeholder="DOB- DD/MM/YYYY">

<br>

<input type="text" placeholder="Email ID" name="email" required="" autocomplete="off">

<br>

<input type="text" placeholder="Mobile No" name="mobile" required="" autocomplete="off">

<br>

<input type="text" placeholder="Face Book ID" name="fbid" required="" autocomplete="off">

<br>

<input type="text" placeholder="Address" name="address" required="" autocomplete="off">

<br>

<input type="text" placeholder="About him/her" name="about" required="" autocomplete="off">

<br><br/>

<input type="file" name="image" required="" autocomplete="off">

<br>

<br>

<input type="submit" value="ADD">

<br>

<br>

</form>

</div>

<br>

<!-- Main Footer -->

<footer class="main-footer">

<p>Developed by <a href="" target="\_blank"> </a></p>

</footer>

</body>

</html>

<%@page import="java.sql.\*"%>

<%@ include file="connect.jsp" %>

<%@ page session="true" %>

<%

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

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

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

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

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

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

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

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

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

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

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

try{

PreparedStatement ps = connection.prepareStatement("insert into studentevents(username,rno,yr,branch,dob,email,mobile,fbid,address,about,image) values(?,?,?,?,?,?,?,?,?,?,?)");

ps.setString(1,username);

ps.setString(2,rno);

ps.setString(3,year);

ps.setString(4,branch);

ps.setString(5,dob);

ps.setString(6,email);

ps.setString(7,mobile);

ps.setString(8,fbid);

ps.setString(9,address);

ps.setString(10,about);

ps.setString(11,image);

ps.executeUpdate();

response.sendRedirect("addsachievements.jsp?msg=success");

}

catch(Exception e1)

{

response.sendRedirect("addsachievements.jsp?m1=Failed");

out.println(e1.getMessage());

}

%>

<%@page import="java.sql.\*"%>

<%@ include file="connect.jsp" %>

<%@ page session="true" %>

<%

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

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

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

if(utype.endsWith("Admin")){

String query = "select \*from login where username='"+username+"' and password='"+password+"' and utype = '"+utype+"'";

Statement st = connection.createStatement();

ResultSet rs = st.executeQuery(query);

if(rs.next()){

response.sendRedirect("adminhome.jsp?msg=success");

}

else{

response.sendRedirect("login.jsp?m1=Failed");

}

}

if(utype.endsWith("Student")){

String query = "select \*from login where username='"+username+"' and password='"+password+"' and utype = '"+utype+"'";

Statement st = connection.createStatement();

ResultSet rs = st.executeQuery(query);

if(rs.next()){

session.setAttribute("user", username);

response.sendRedirect("studenthome.jsp?msg=success");

}

else{

response.sendRedirect("login.jsp?m1=Failed");

}

}

if(utype.endsWith("Teacher")){

String query = "select \*from login where username='"+username+"' and password='"+password+"' and utype = '"+utype+"'";

Statement st = connection.createStatement();

ResultSet rs = st.executeQuery(query);

if(rs.next()){

session.setAttribute("user", username);

response.sendRedirect("teacherhome.jsp?msg=success");

}

else{

response.sendRedirect("login.jsp?m1=Failed");

}

}

%>

%>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Online College Year Book</title>

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">

</head>

<body>

<!-- Main Header -->

<header class="main-header">

<center><h1>Online College Year Book</h1></center>

</header>

<!-- Dropdown Navbar -->

<nav>

<ul>

<li><a href="index.html" >Home</a></li>

<li><a href="login.jsp">Login</a></li>

<li><a href="register.jsp" class="active">Register</a></li>

</ul>

</nav>

<!-- Login Form -->

<div class="form-div animated flip">

<img src="img/user.png">

<h2>Registration Form</h2>

<form action="registeract.jsp" method="post">

<input type="text" placeholder="ID" name="id" required="" autocomplete="off">

<br>

<input type="text" placeholder="User Name" name="username" required="" autocomplete="off">

<br>

<input type="password" placeholder="Password" name="password" required="" autocomplete="off">

<br>

<input type="email" placeholder="E-mail Id" name="email" required="" autocomplete="off">

<br>

<input type="text" placeholder="Address" name="address" required="" autocomplete="off">

<br>

<input type="text" placeholder="Contact No" name="mobile" required="" autocomplete="off">

<br>

<select name="utype" id="" required="">

<option value="">--Select--</option>

<option value="Teacher">Teacher</option>

<option value="Student">Student</option>

</select>

<br>

<br>

<input type="submit" value="REGISTER">

<br>

<br>

</form>

</div>

<br>

<!-- Main Footer -->

<footer class="main-footer">

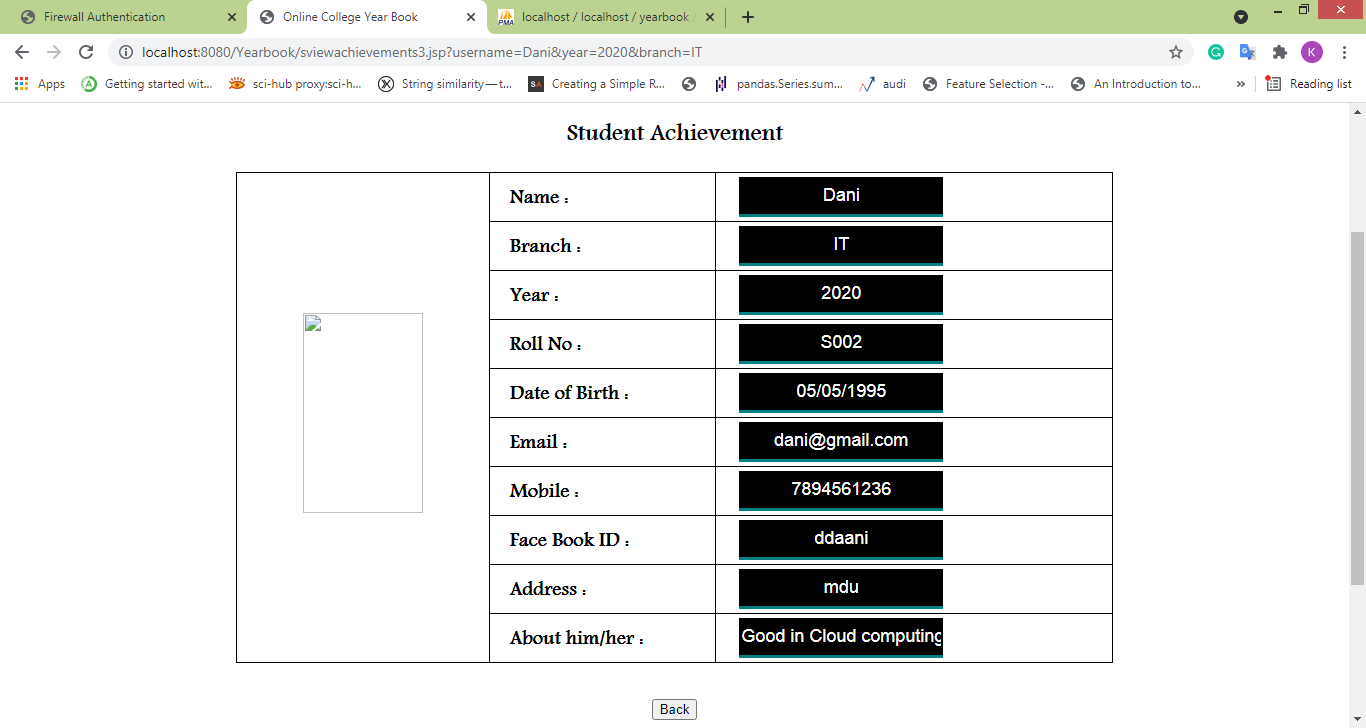
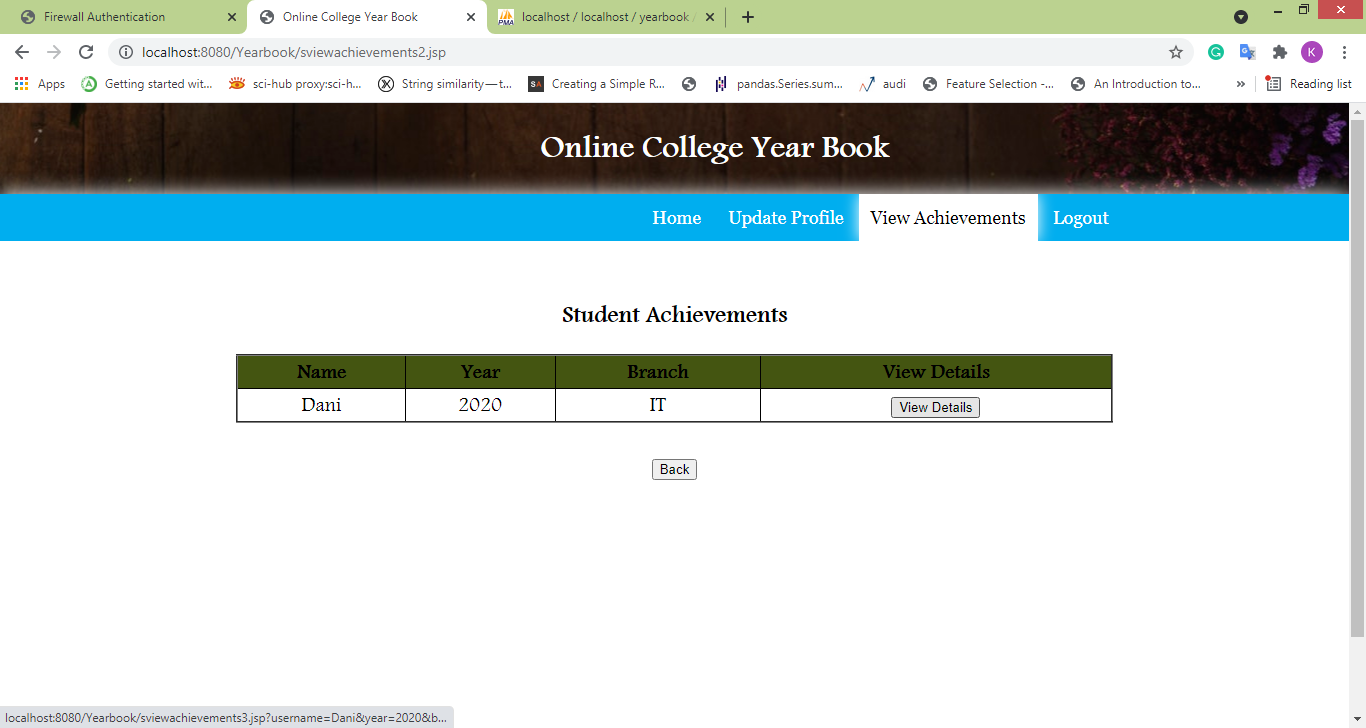
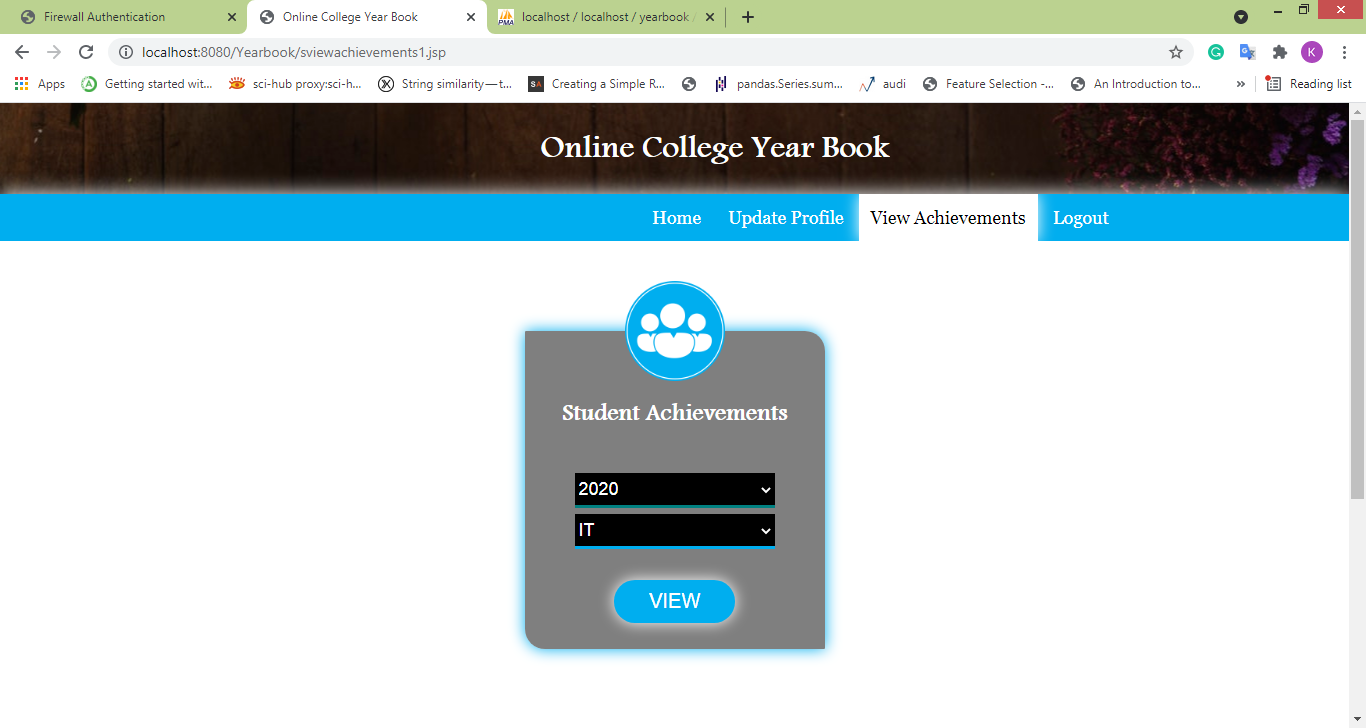
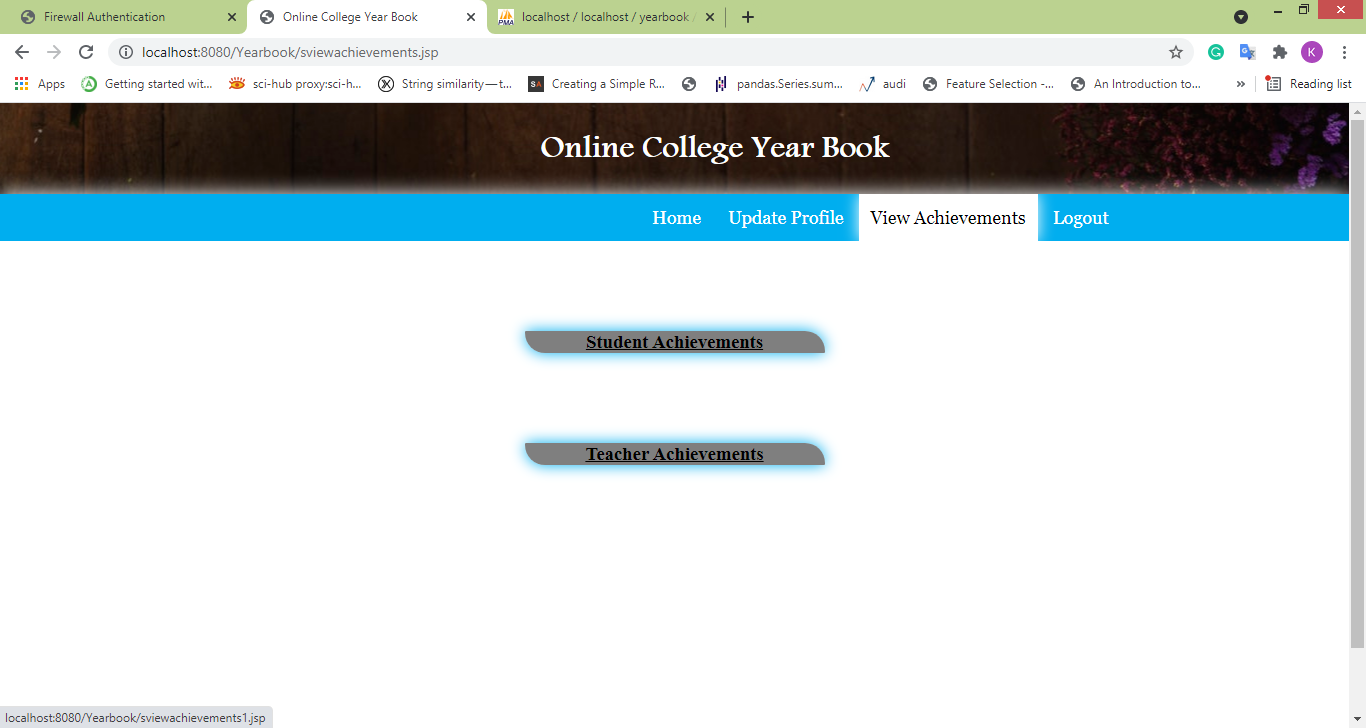
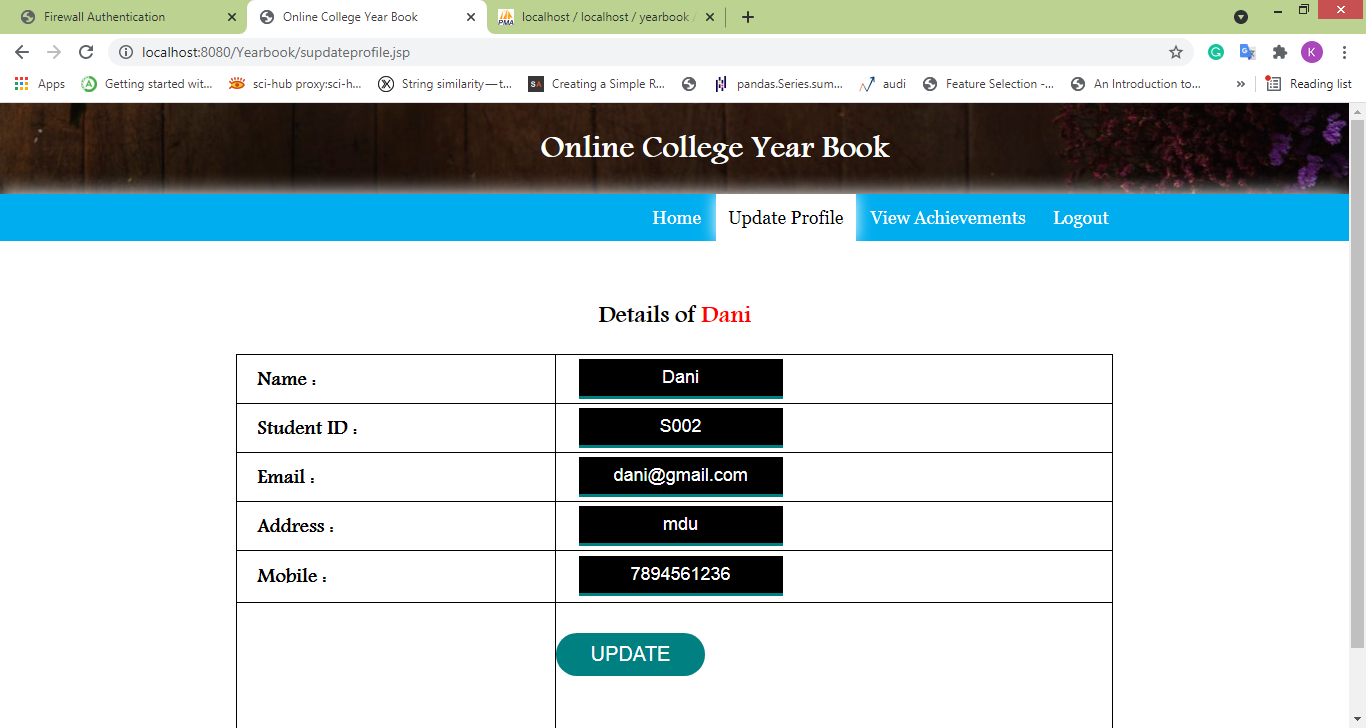
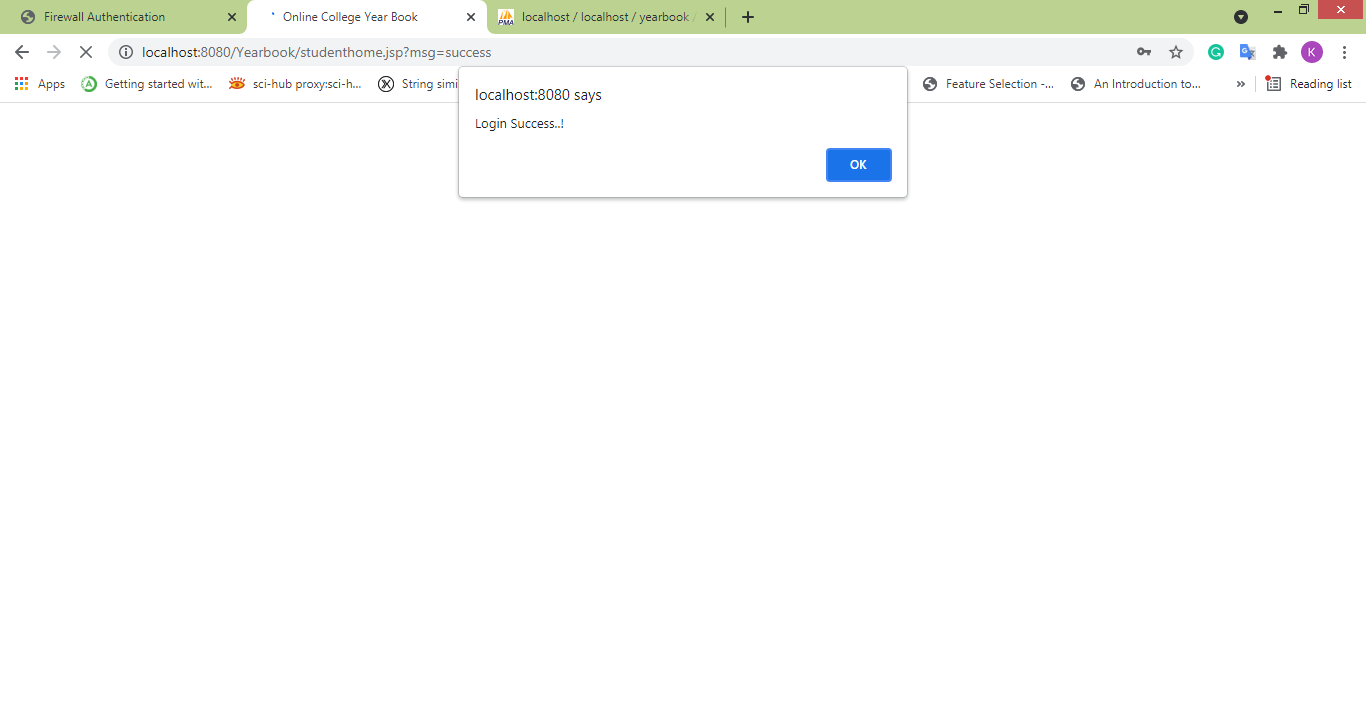
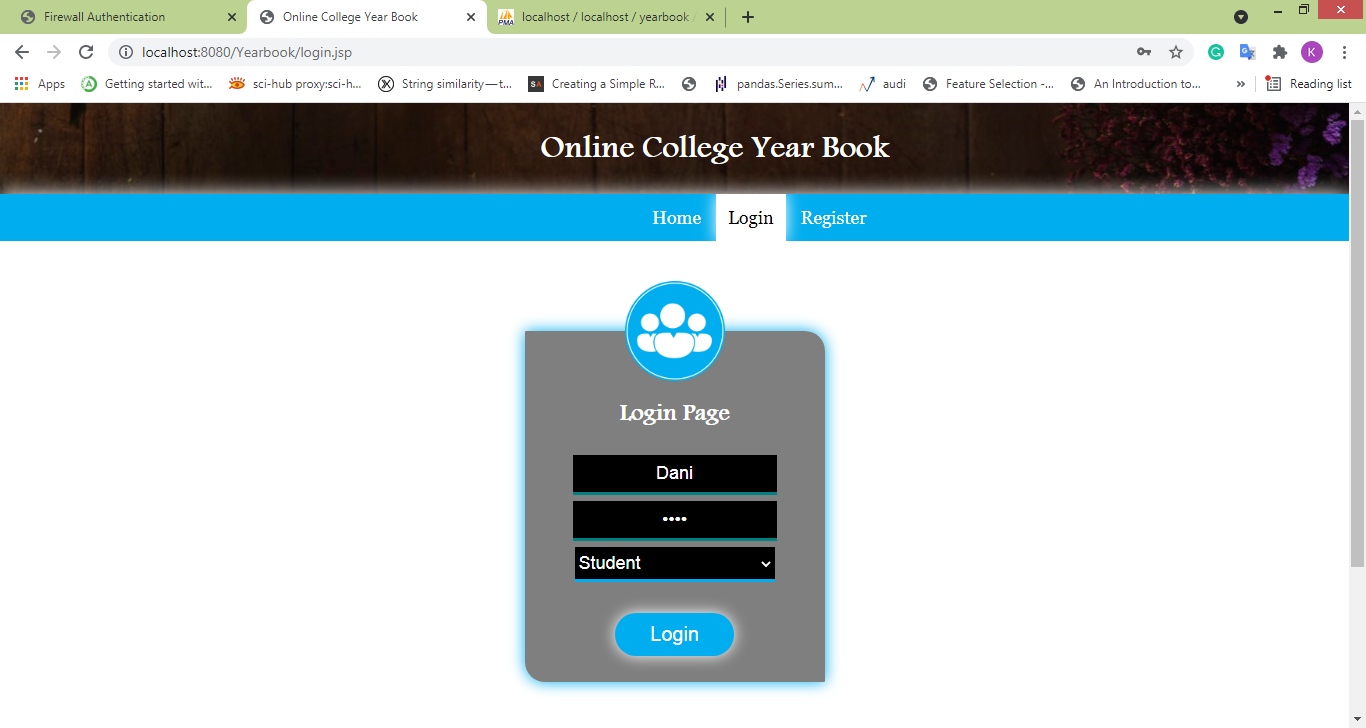
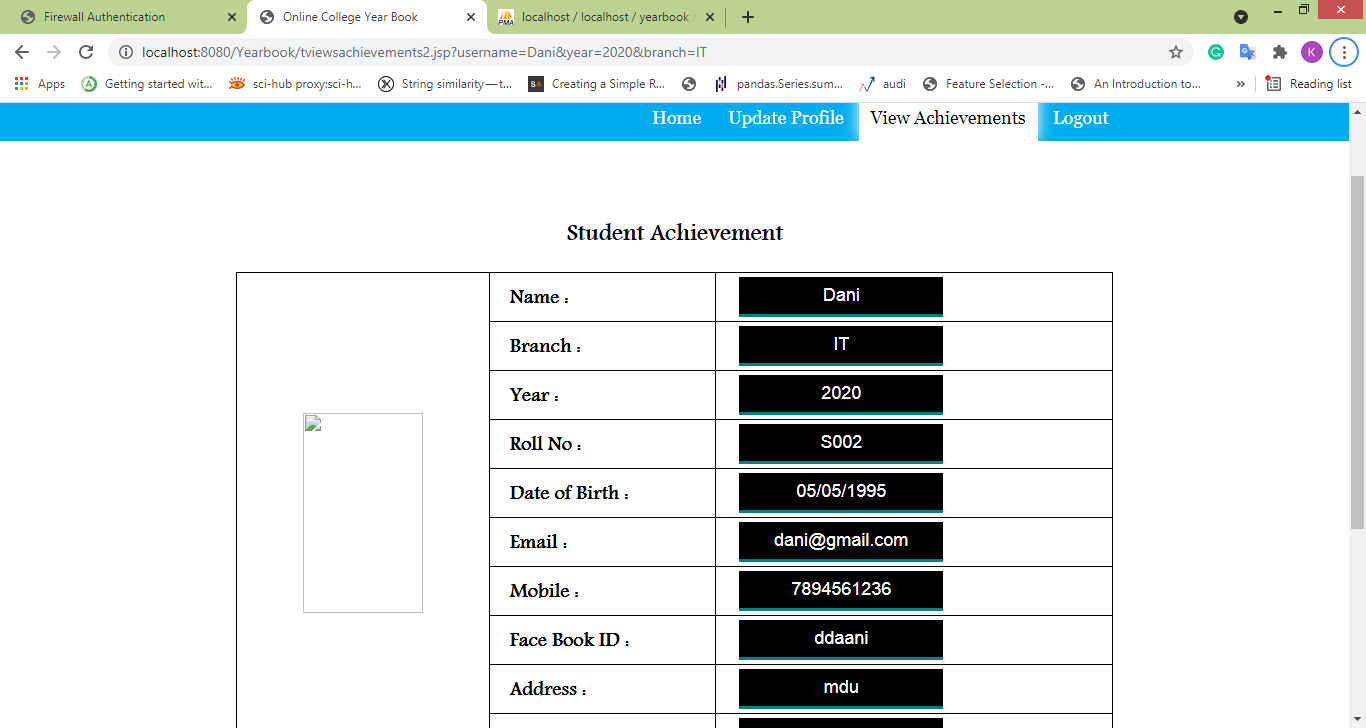
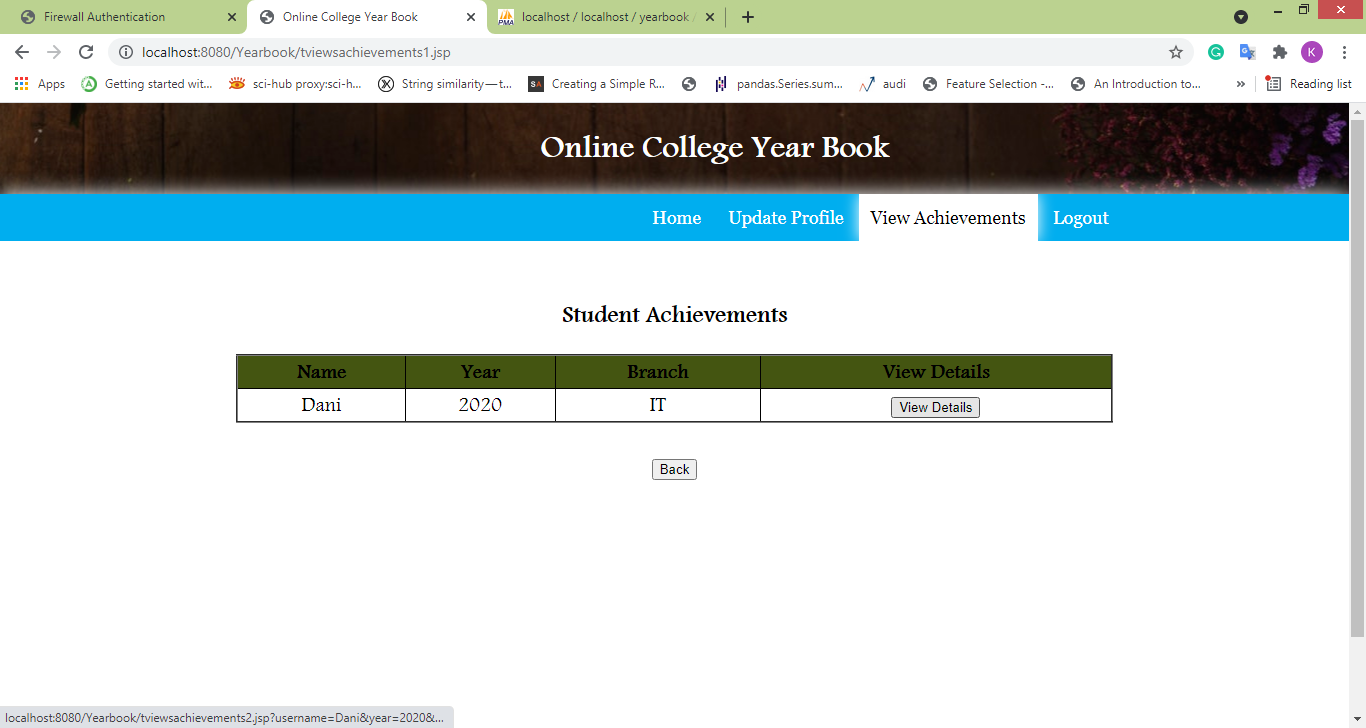
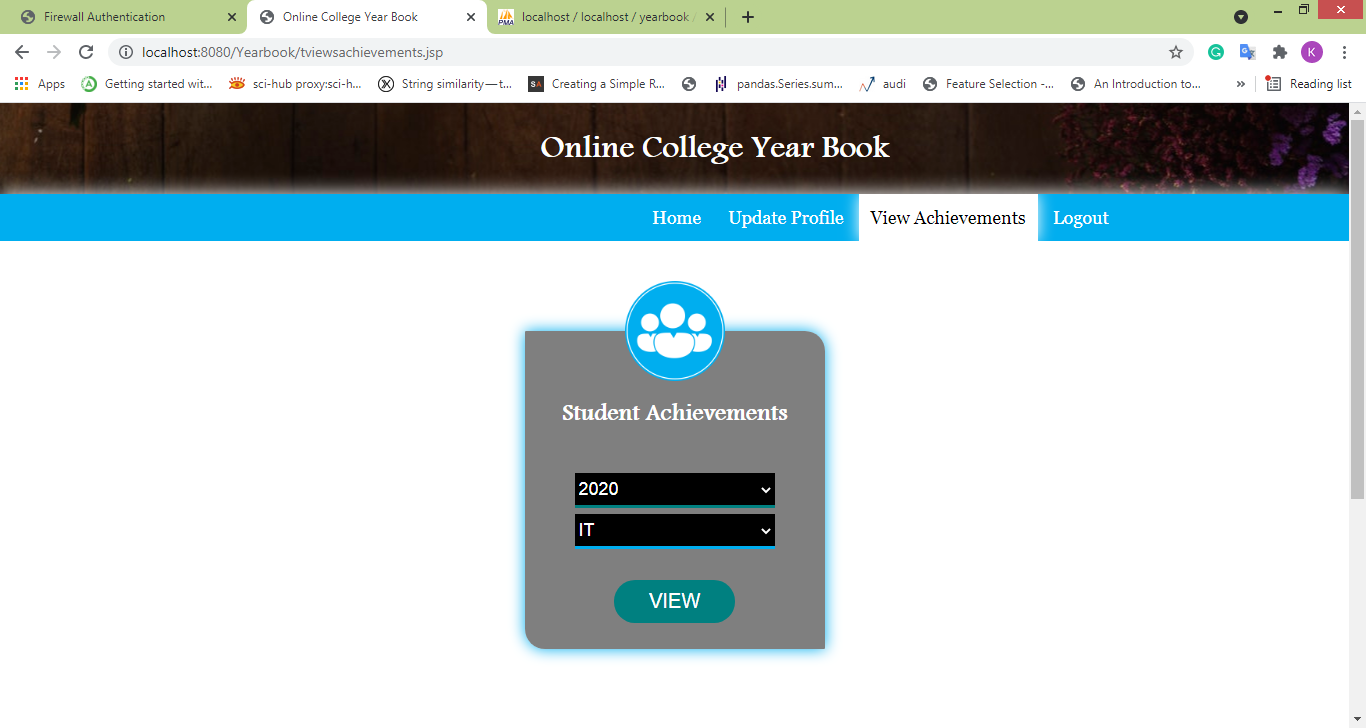
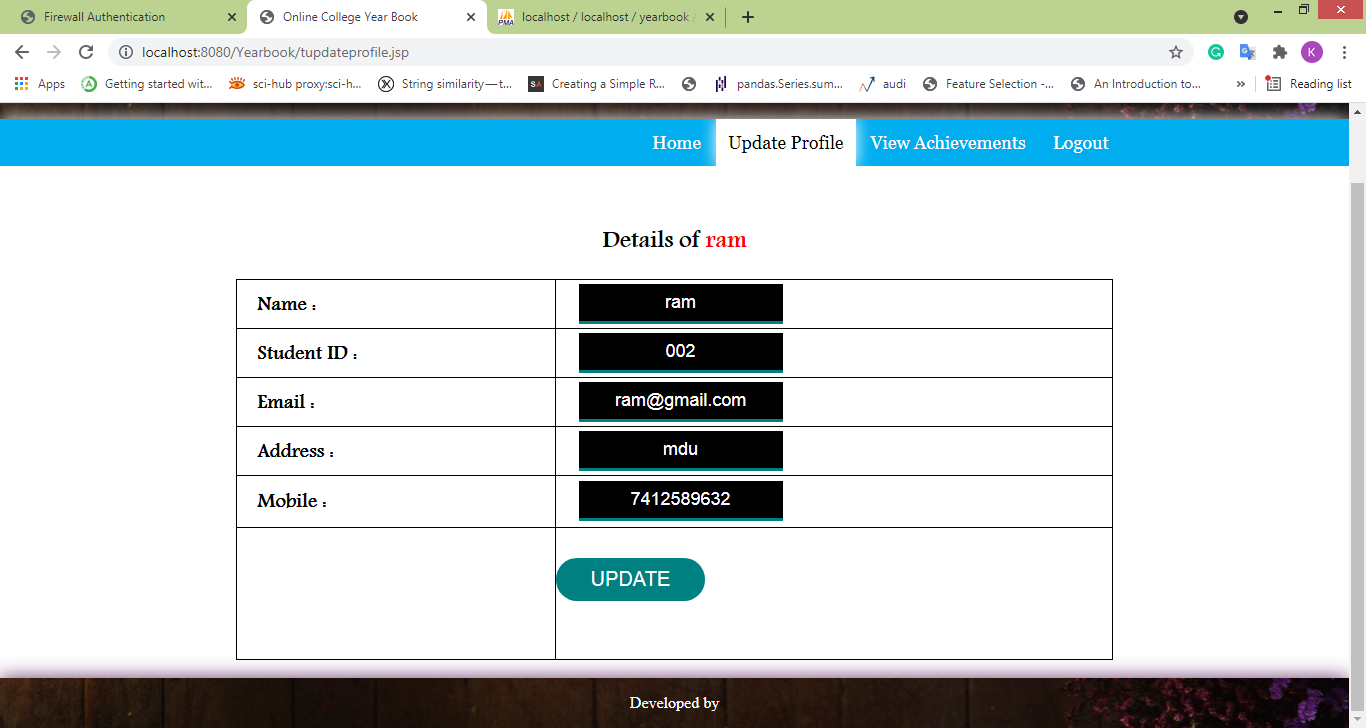
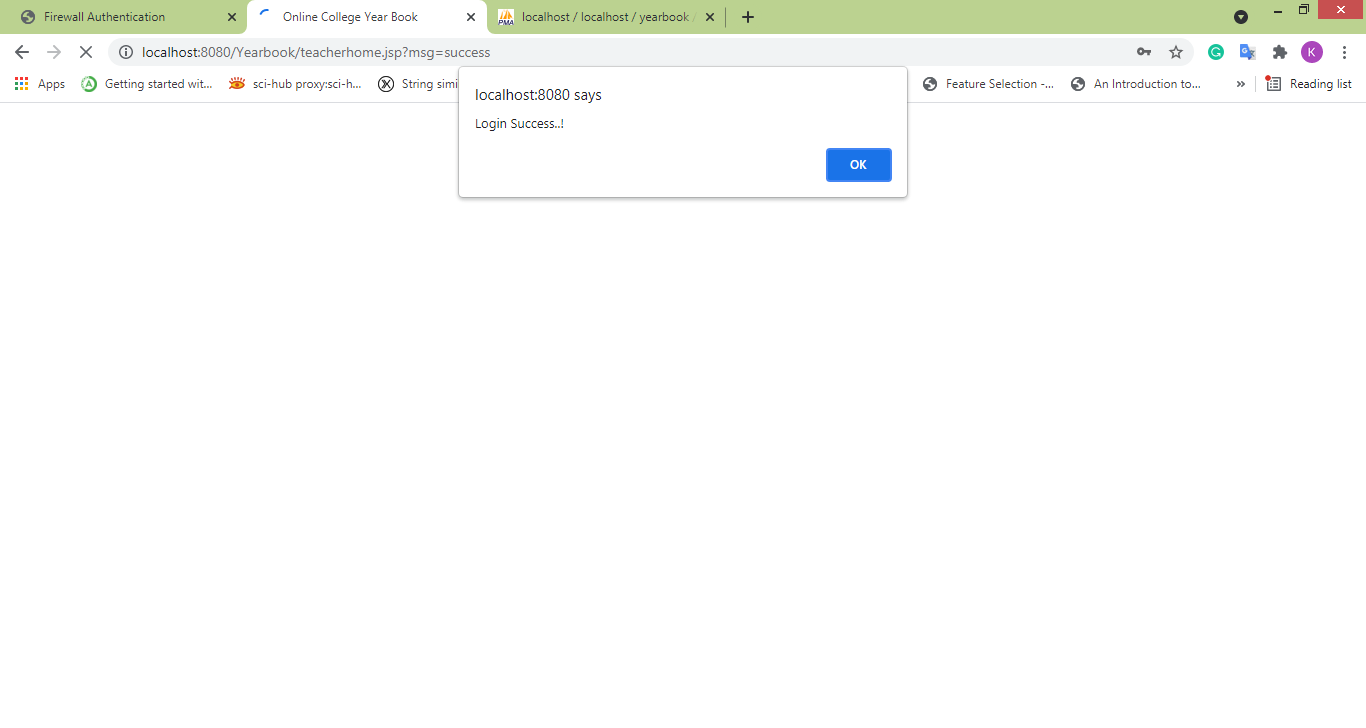
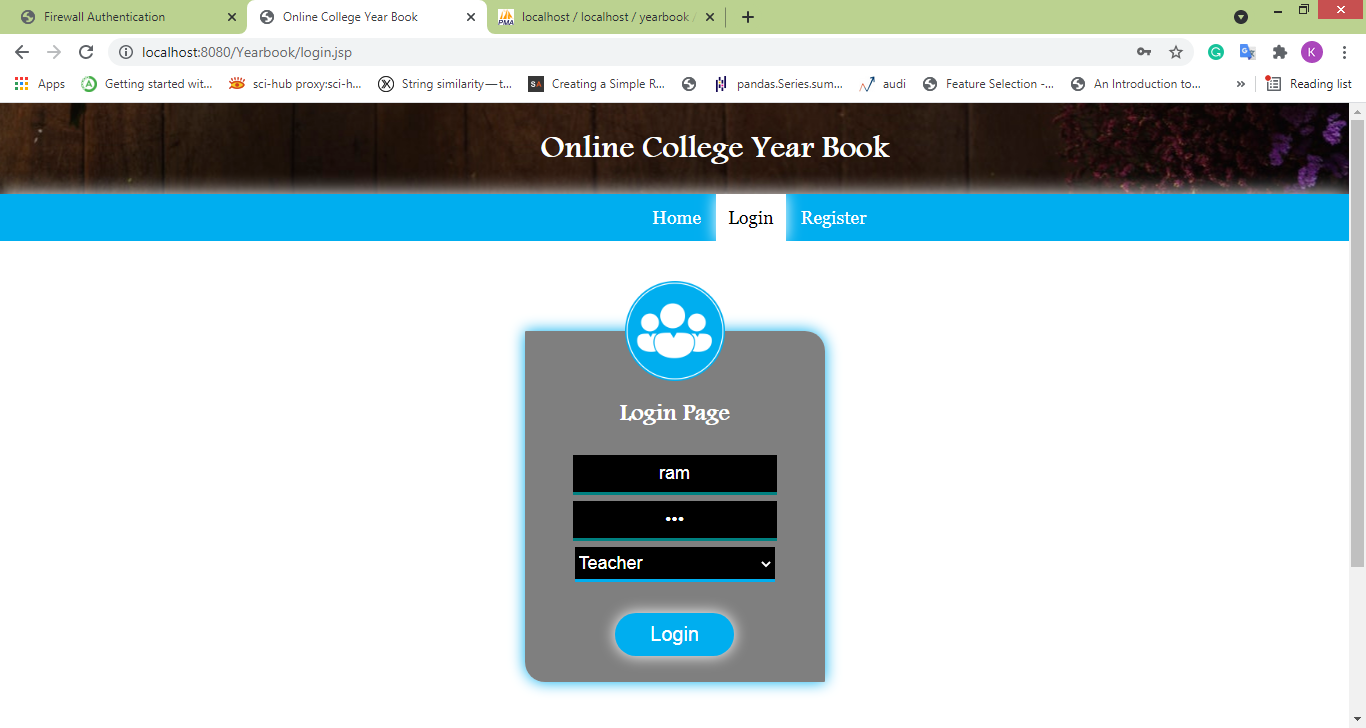
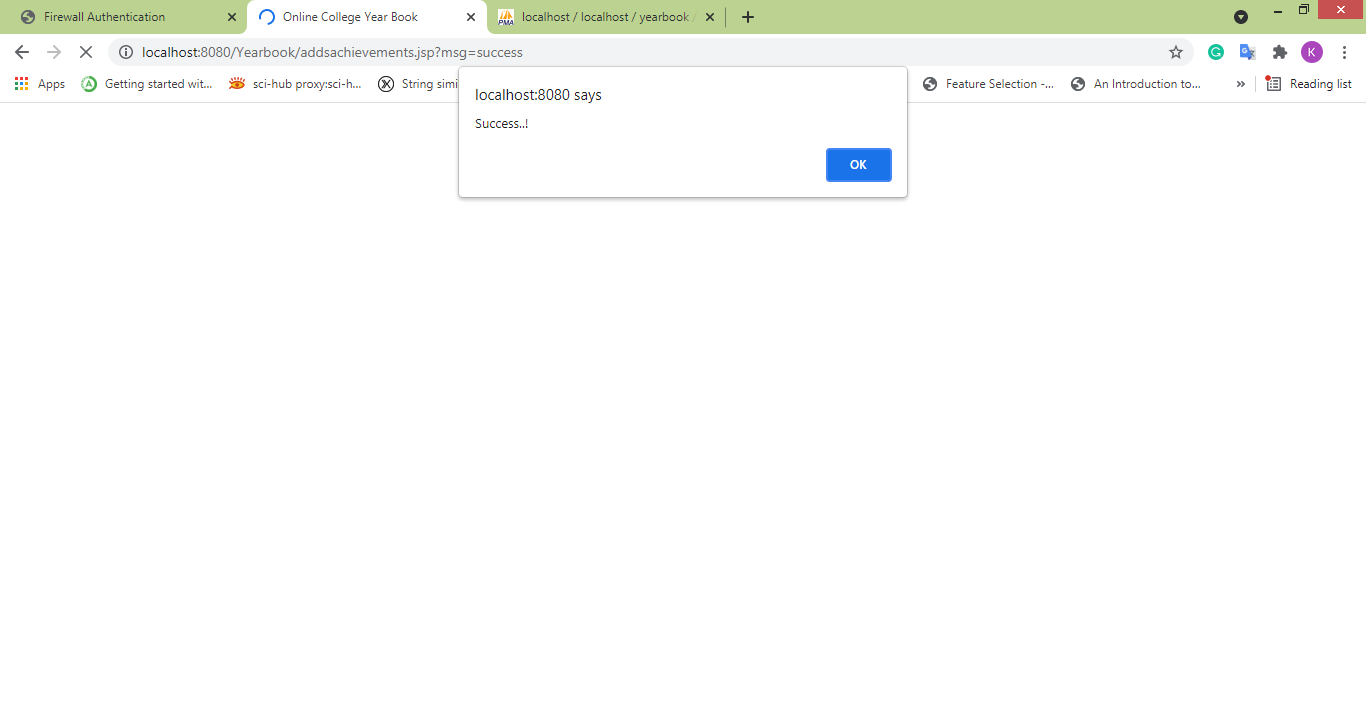
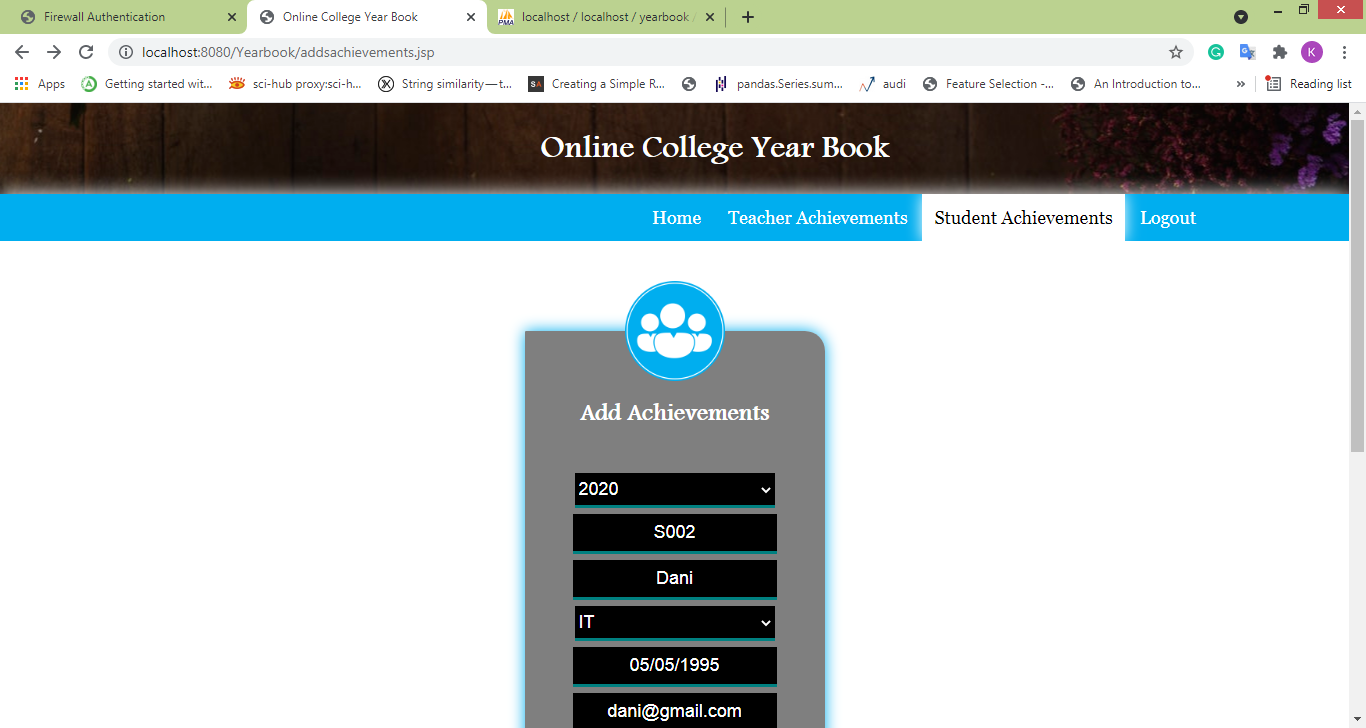
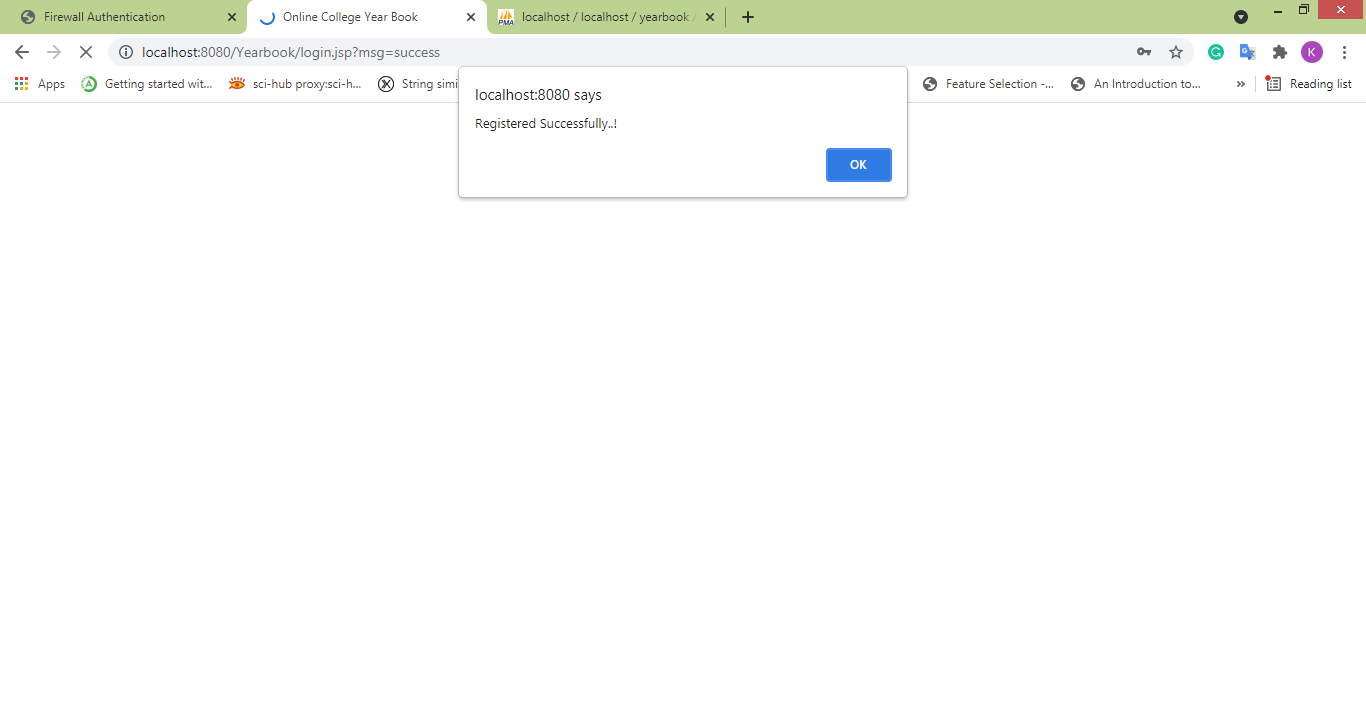
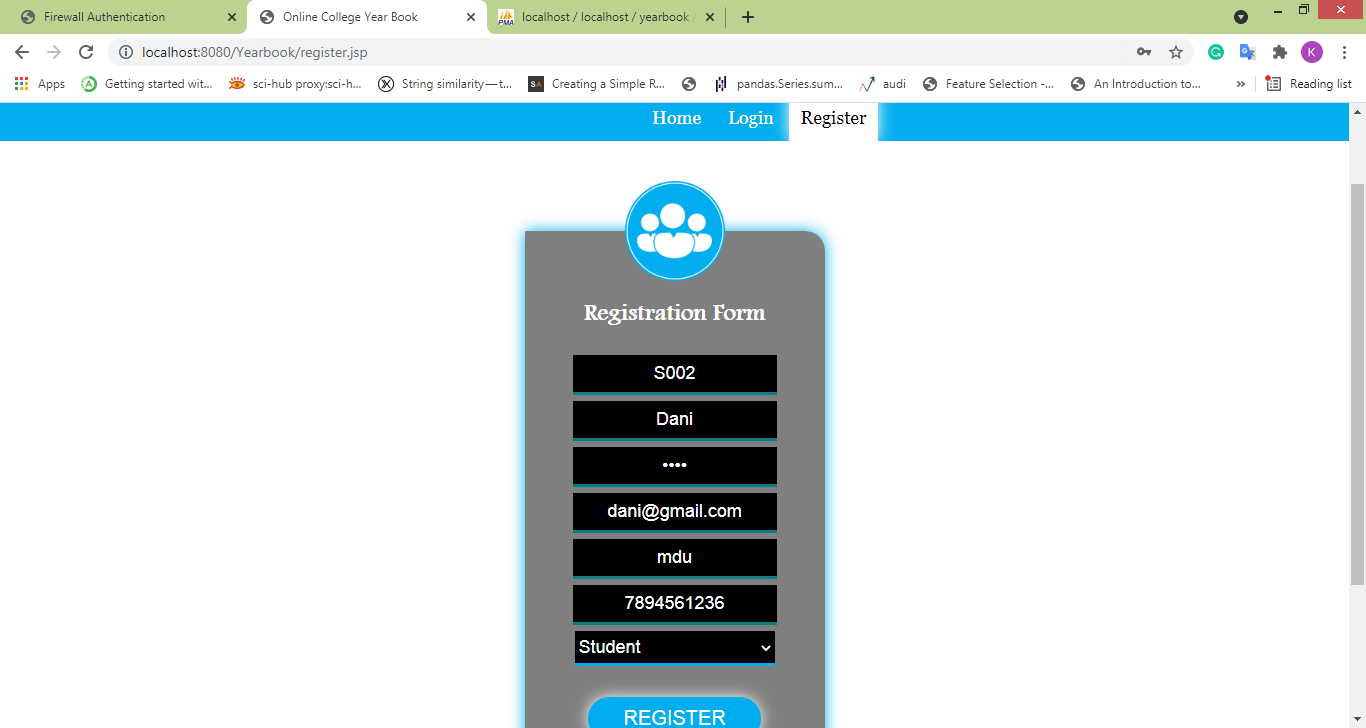
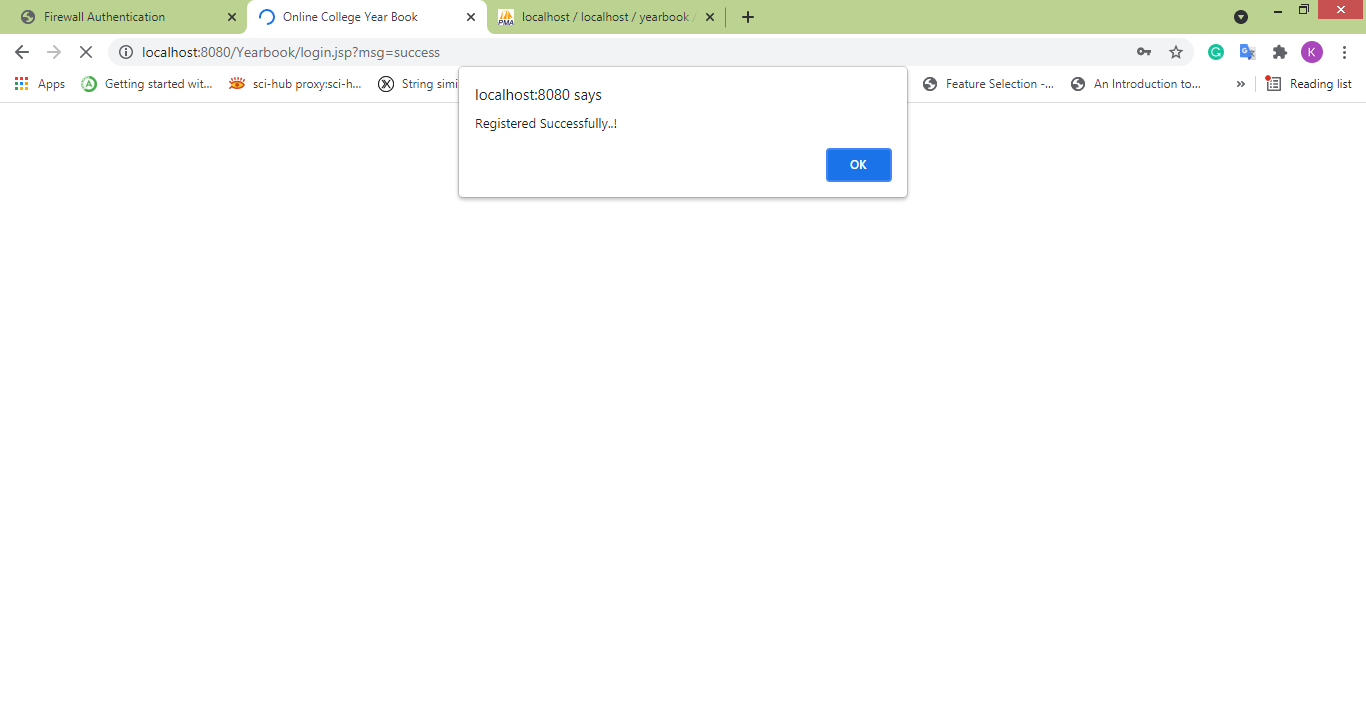
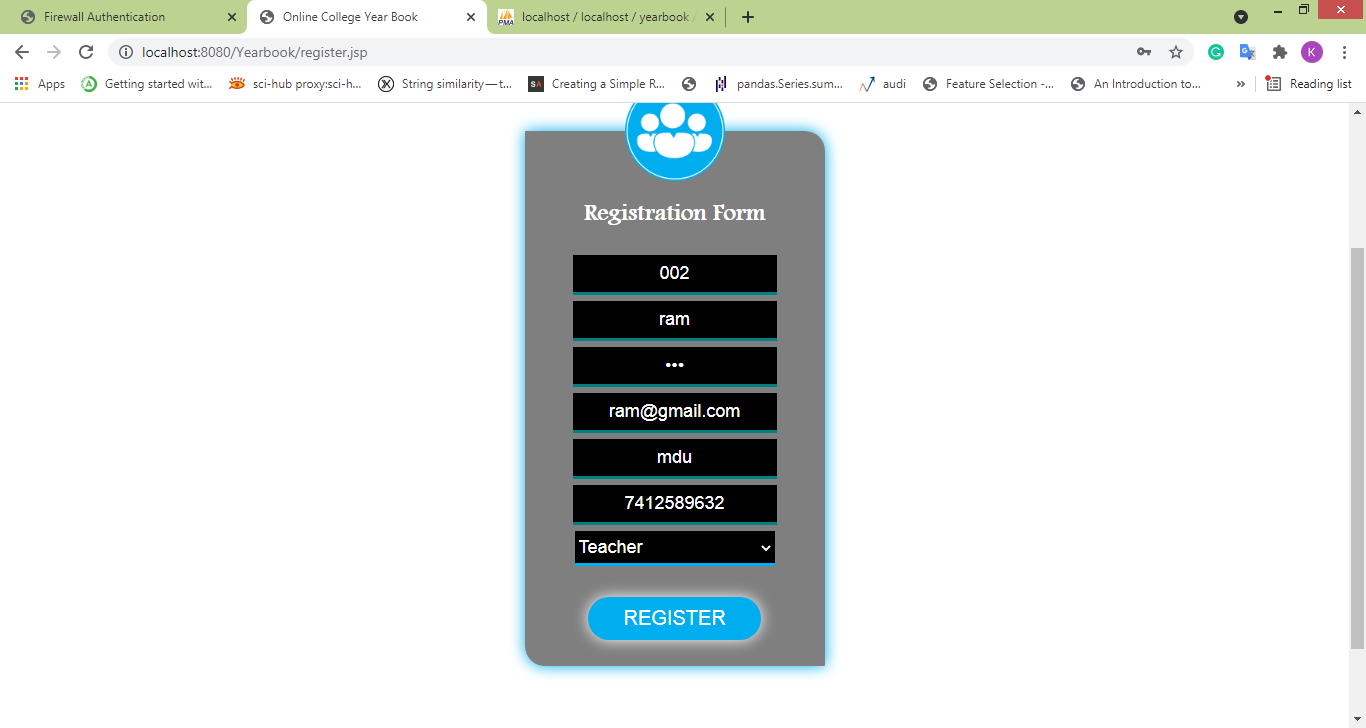
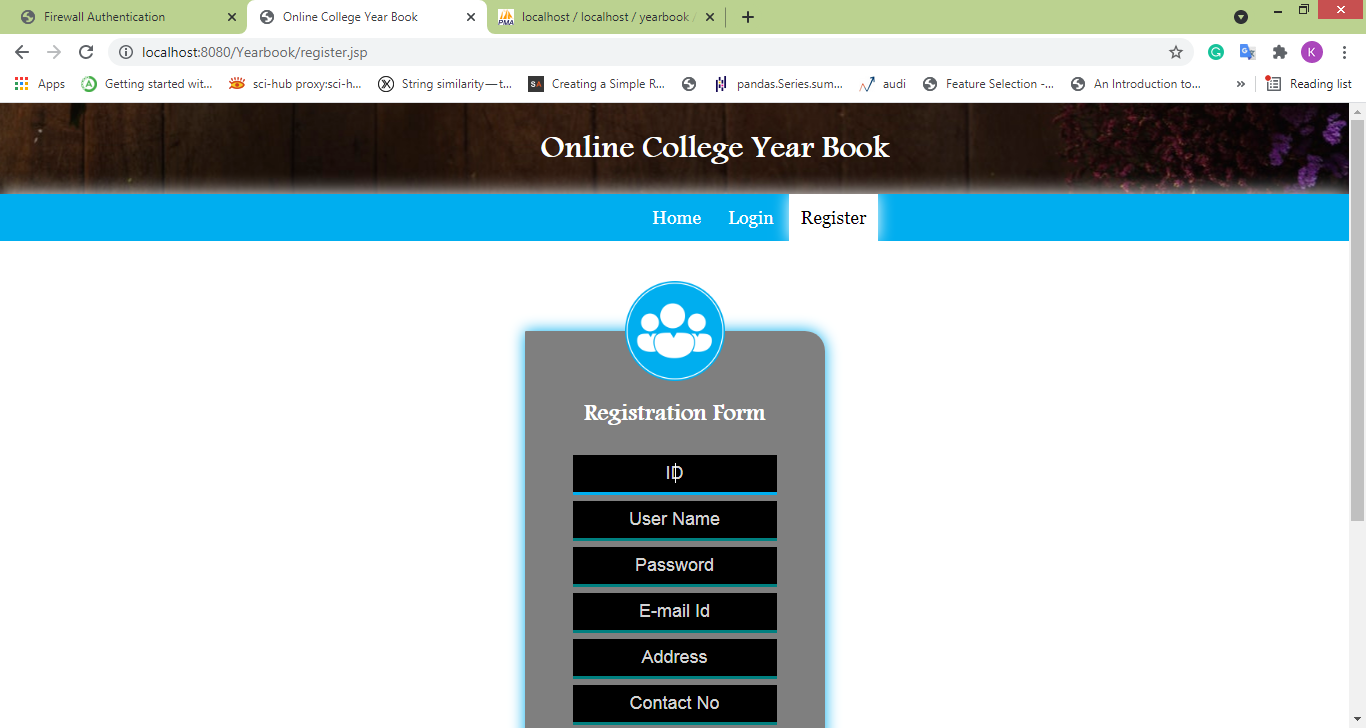
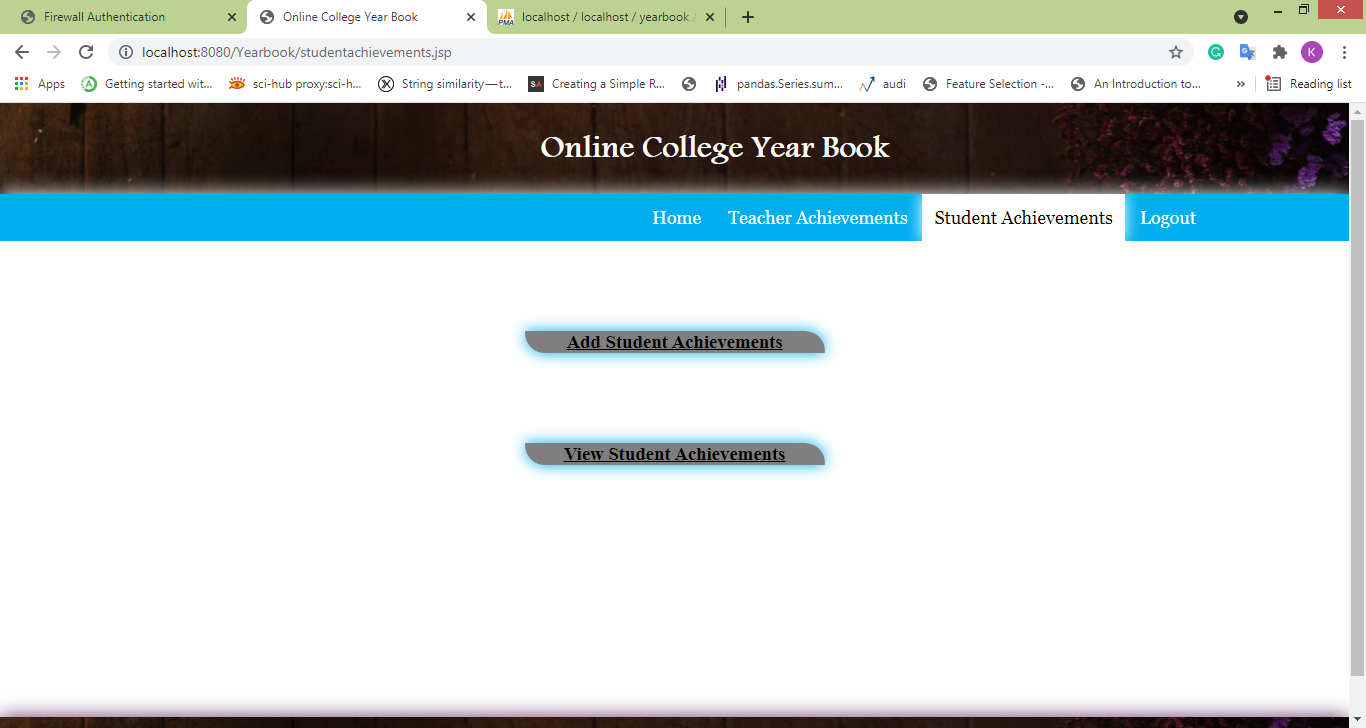
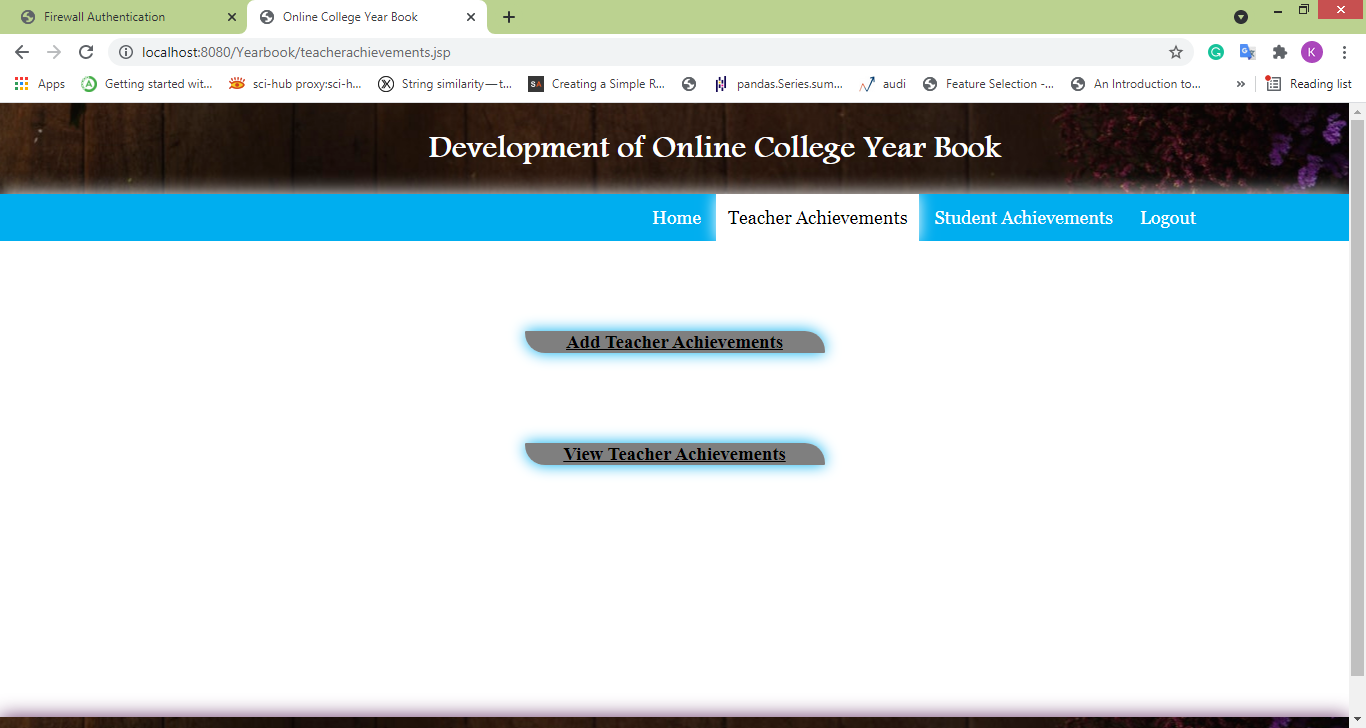
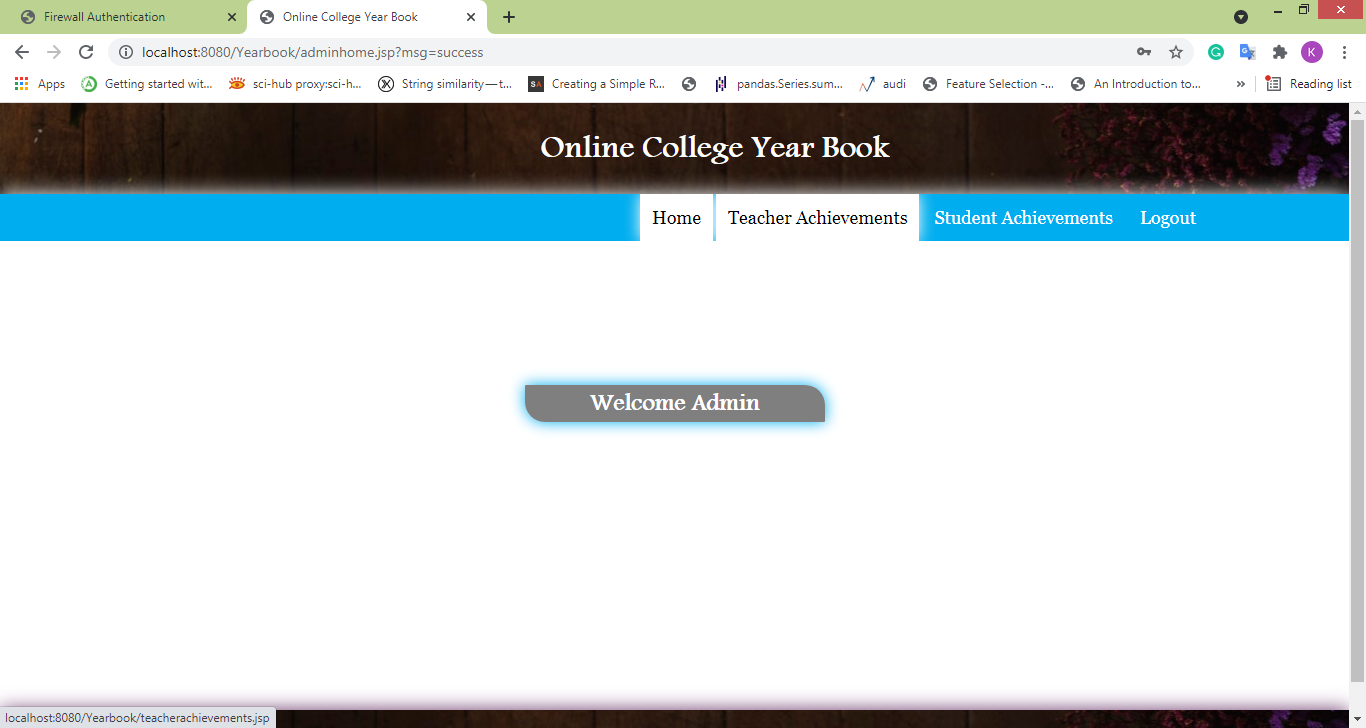
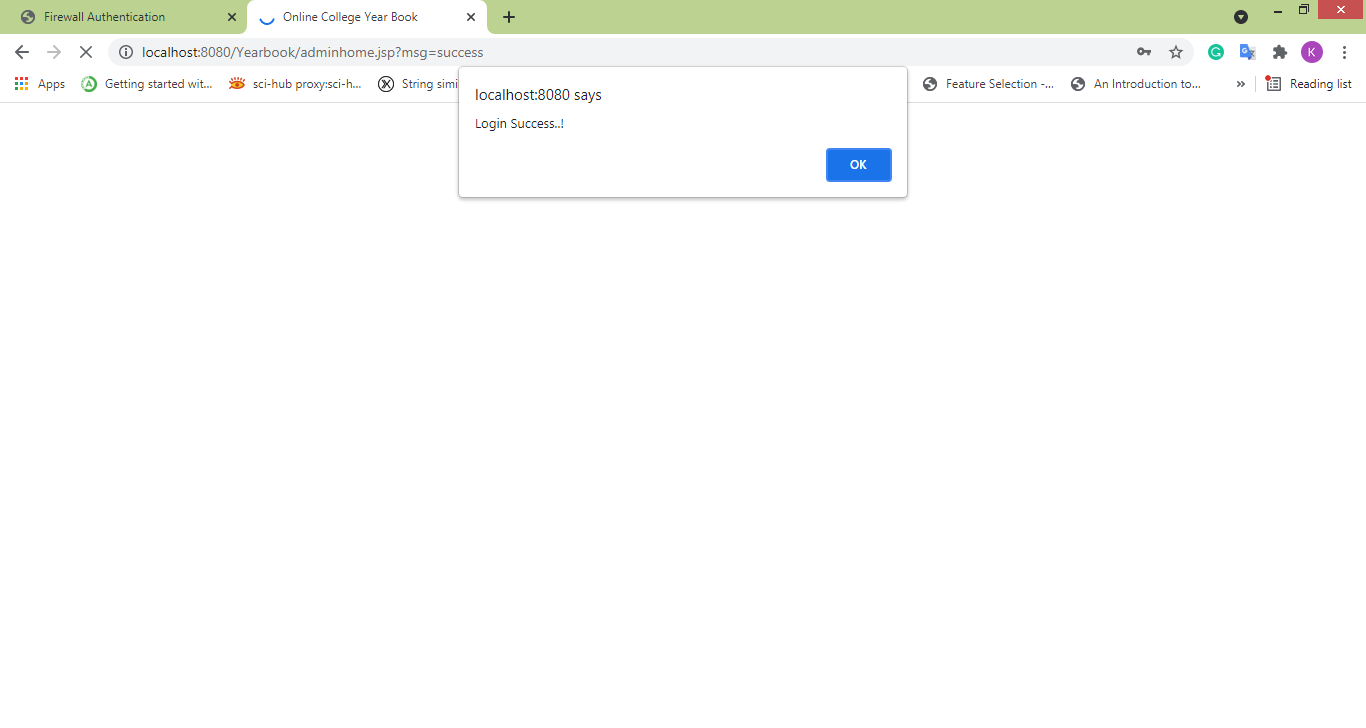
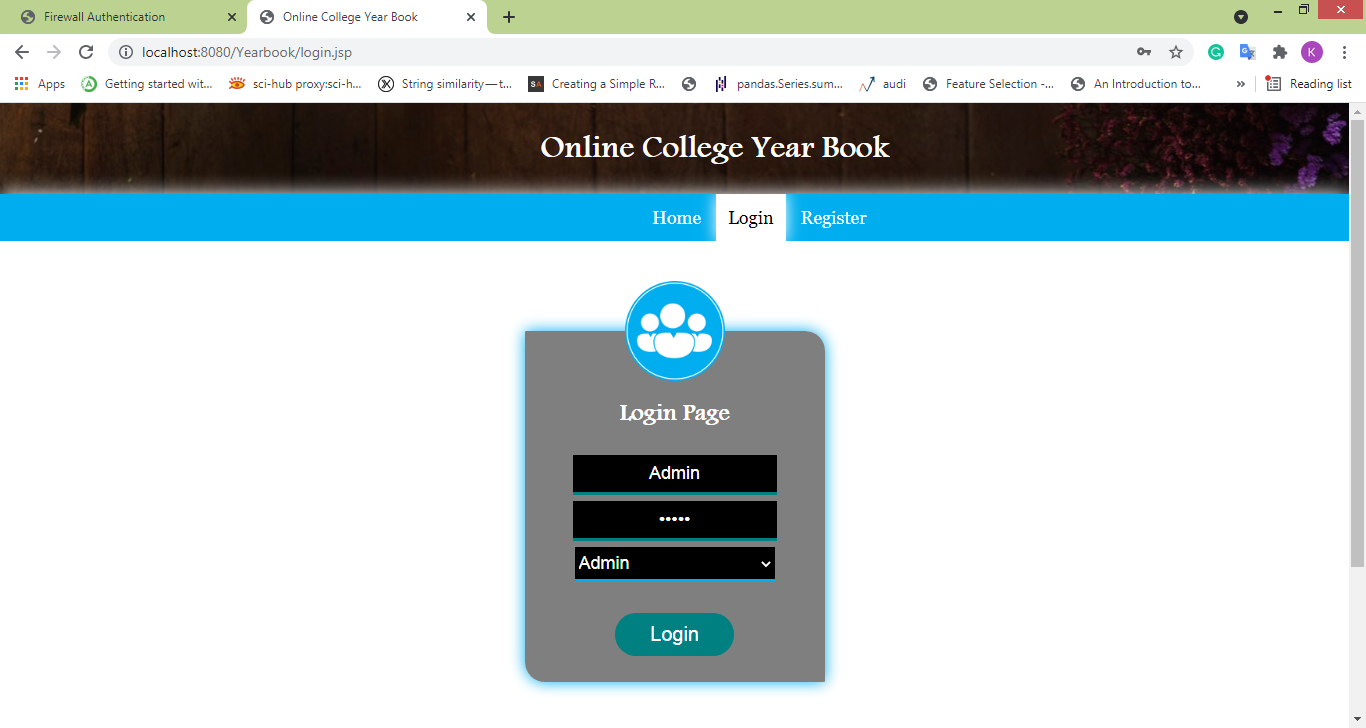
<p>Developed by <a href="" target="\_blank"> </a></p>

</footer>

</body>

</html>

**SCREENSHOTS:**



1. **REFERENCES**

[1] https://www.geeksforgeeks.org/introduction-to-jsp/

[2] https://www.javatpoint.com/servlet-tutorial

[3] https://www.mysql.com/