**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belagavi-560018, Karnataka**

****

**EMERGING TECHNOLOGIES**

REPORT

**On**

**“STUDENT MANAGEMENT WEBAPP”**

**BACHELOR OF ENGINEERING**

**In**

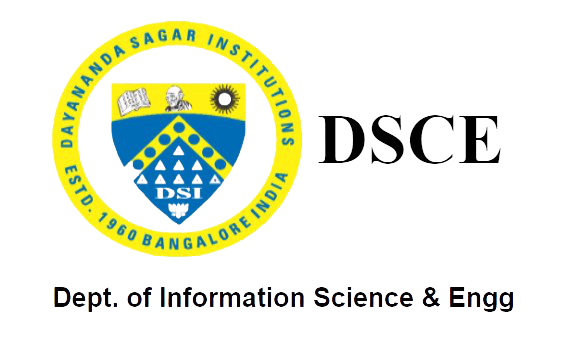
**INFORMATION SCIENCE AND ENGINEERING**

**Submitted by:**

**Anagha H**

**FACULTY IN-CHARGE:**

Mr. Venkatesh Patil



**2020-2021**

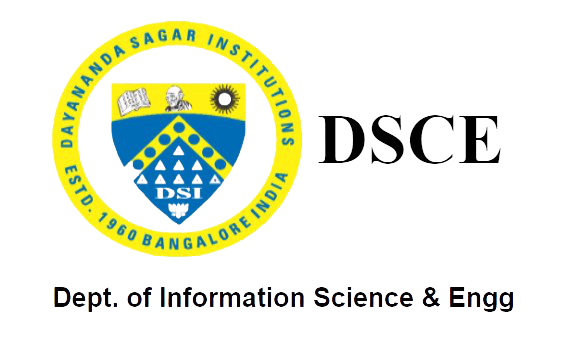
**Department Of Information Science and Engineering**

**DAYANANDA SAGAR COLLEGE OF ENGINEERING**

**SHAVIGE MALLESHWARA HILLS, KUMARASWAMY LAYOUT**

**Bengaluru-560078**

**Department of Information Science & Engineering**

****

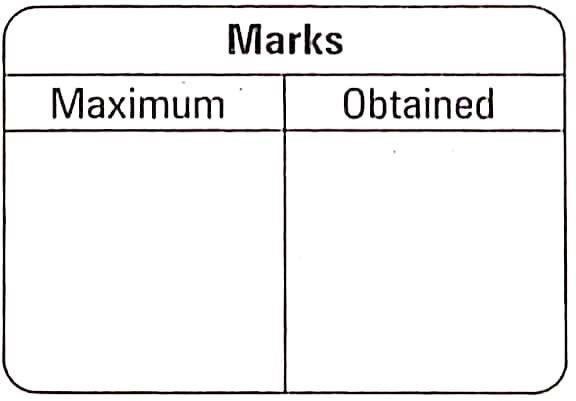
**2020-21**

**CERTIFICATE**

This is to certify that Emerging Technologies, a report entitled

**“STUDENT MANAGEMENT WEBAPP”** is a bonafide work carried out by

**Anagha and Team** in the partial fulfillment for the **5th semester** of Bachelor of Engineering in **Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2020-21.



Signature of Faculty in-charge:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[Mr. Venkatesh Patil]

**Abstract**

Student Management System (SMS), provides a simple interface for maintenance of student information. It can be used by educational institutions to maintain the records of students easily. The creation and management of accurate and up-to-date information regarding a students’ academic career is critically important in the university as well as colleges. Student information system deals with all kind of student details, academic related reports, college details, course details, curriculum and other resource related details too. It tracks all the details of a student from day one to the end of the course which can be used for all reporting purposes, tracking of attendance, progress in the course, completed semesters, years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result and all these will be available through a secure, online interface embedded in the college’s website. It will also have faculty details, batch execution details, students’ details in all aspects, the various academic notifications to the staff and students updated by the college administration. It also facilitates us to explore the activities happening in the college, different reports and queries can be generated based on vast options related to students, batches, courses, faculty, exams, semesters, certification and even for the entire college.

**Contents**

1. Introduction

2.0 DevOps

3.0 DevOps Tools Used

4.0 Implementation of DevOps Tools for Version Management

5.0 Implementation of DevOps Tools for Build Management

6.0 Implementation of DevOps Tools for Test Management

7.0 Conclusion and future scope

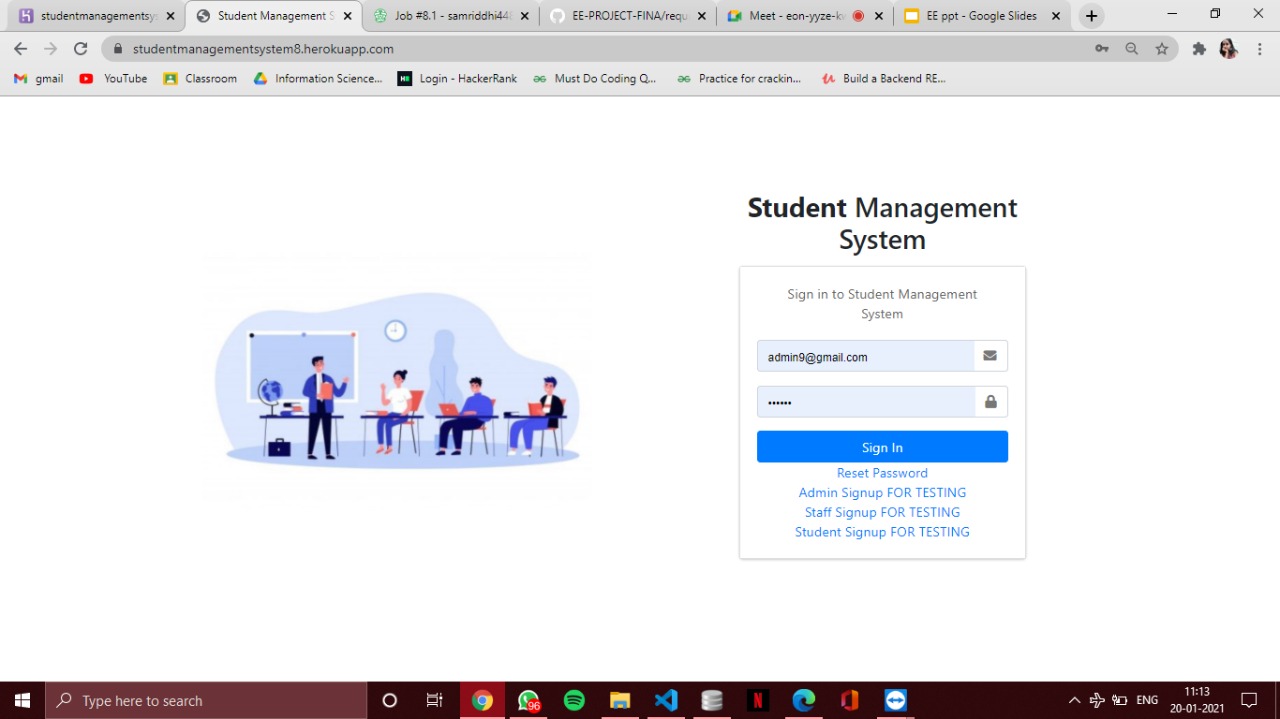
**1.0 Introduction**

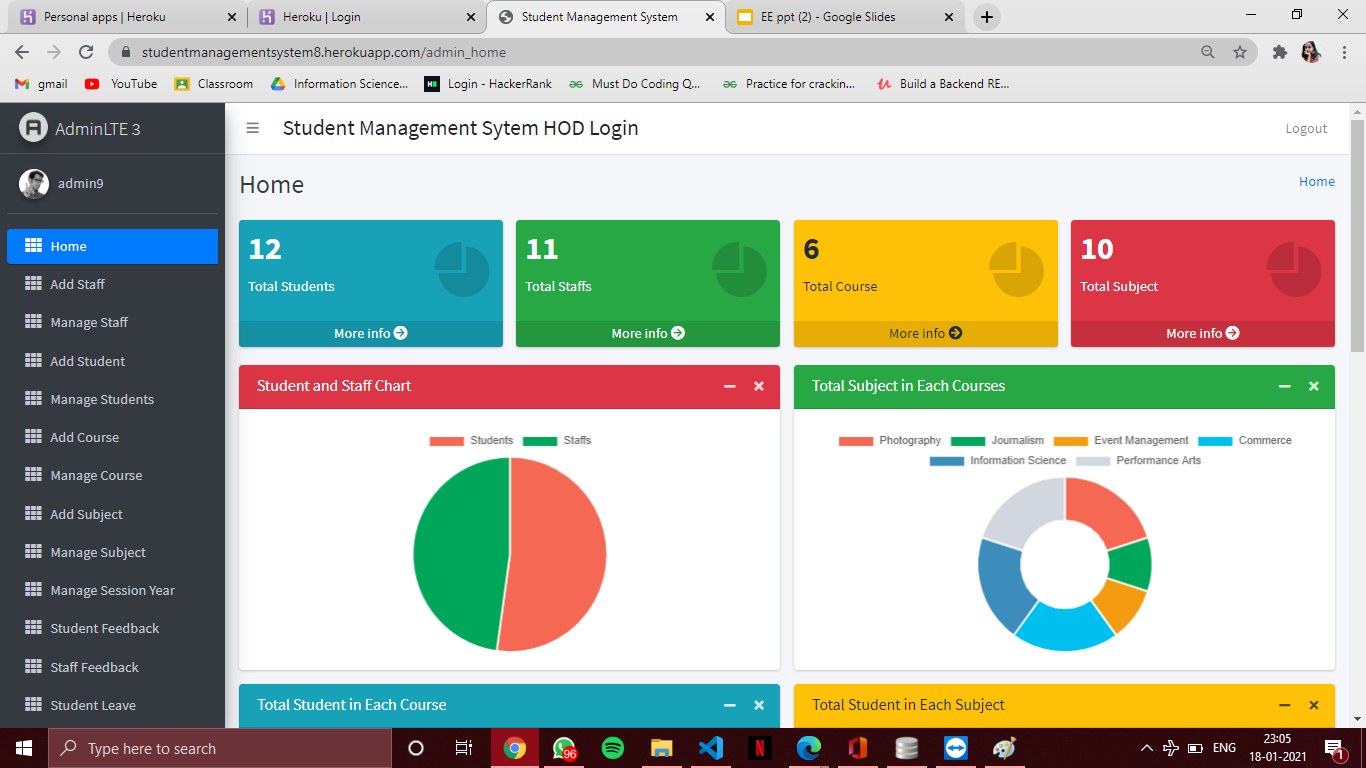
A Student Management web app, that provides different features for different groups. The student can check their attendance status, marks, apply for leave and give their feedback. The staff can mark student’s attendance, update their profile etc.. The head of the department can manage students and staff altogether. They’re considered as the admin. It will also help in saving time and effort. The user interface must be user friendly and easy to understand. The information of the particular student will be obtained in just one mouse click. The advantages it possesses is as follows:

1. **Student Database Management**: The details of the students at the organization can be stored in the database with the use this application.
2. **Results**: The results of the students can also be accessed and stored through this application.
3. **Security**: The data that will be disclosed will be more secure since no access shall be allowed to an unknown user.
4. **Performance**: The performance of the students that might be in curriculum as well as co-curriculum can also be stored through the use of this application.
5. **One click access**: You will obtain the details of the students by entering their name or the roll number just in one-click
6. **User interface**: The user interface should be simple and easy to understand.

FEATURES AND MODULES:

|  |  |  |
| --- | --- | --- |
| STAFF LOGIN | HOD LOGIN | STUDENT LOGIN |
| Apply for leave | Manage students | Apply for leave |
| Student attendance | Manage staff | Student profile |
| Send notice to students | Send notification | Feedback |
| Staff profile | View attendance report | See notification |
|  | Manage subject & Courses |  |

****

****

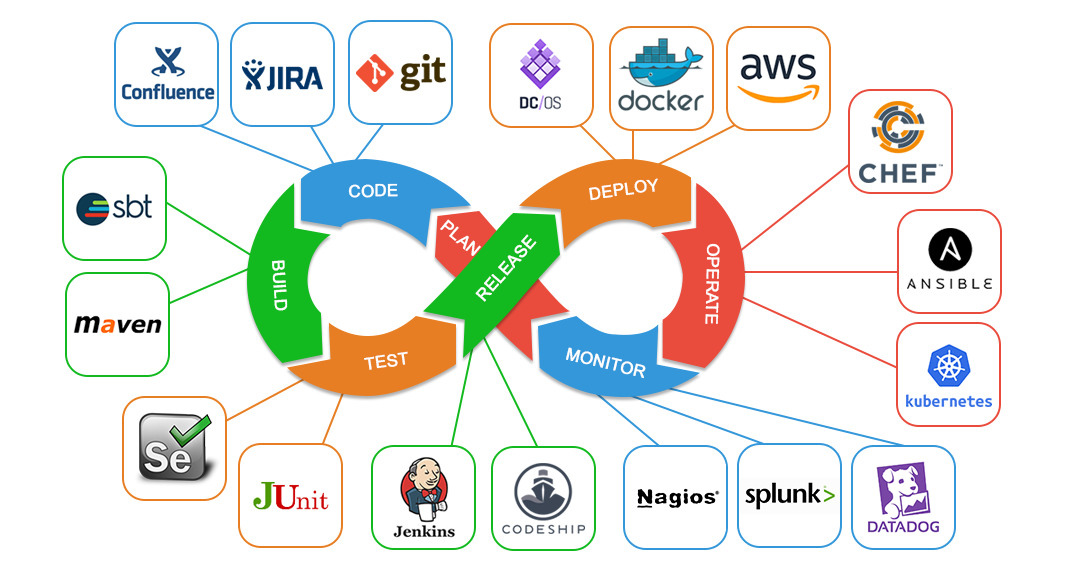
**Team Lead:** ANAGHA H

**Team members along with DevOps tools responsibility:**

1. Anagha H (1DS18IS011): Database, Heroku
2. Samriddhi Jain (1DS18IS056): Backend, Heroku
3. Sanjana S. (1DS18IS059): Frontend, Git
4. Rakshit P. (1DS19IS413): Travis CI
5. Shivani (1DS17IS094): Travis CI
6. Shashank (1DS15IS088): Git
7. Astha (1DS18IS018): Data Entry, GitHub
8. Apoorva (1DS18IS014): Data Entry, GitHub
9. Chaitanya (1DS18IS024): Selenium

**2.0 DevOps**

Introduction:



**DevOps** is a software engineering culture that unifies the development and operations team, under an umbrella of tools to automate every stage so that they can build, test, and release software faster and more reliably. The term DevOps was formed by combining the words “development” and “operations” and signifies a cultural shift that bridges the gap between development and operation teams, which historically functioned in siloes.

Because of the continuous nature of DevOps, practitioners use the infinity loop to show how the phases of the DevOps lifecycle relate to each other. Despite appearing to flow sequentially, the loop symbolizes the need for constant collaboration and iterative improvement throughout the entire lifecycle. The DevOps lifecycle includes phases to plan, build, continuously integrate and deploy (CI/CD), monitor, operate, and respond to continuous feedback.

The **DevOps** approach automates the service management for the support of operational objectives and improves understanding of the layers in the production environment stack.

Details:  
  
DevOps vocabulary:

* **Continuous Delivery**:

It is a software strategy that enables organizations to deliver new features to users  
as fast and efficiently as possible. The core idea is to create a repeatable, reliable

and incrementally improving process for taking software from concept to customer. The goal is to enable a constant flow of changes into production via an automated software.

* **Continuous Delivery pipeline**:

Continuous integration is the practice in software engineering, of merging all developer working copies to a shared mainline several times a day. The main aim of CI is to prevent integration problems.

* **Test-driven development**:

It is a software development process that relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards

* **Test automation**:

It is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes.

DevOps adoption :  
  
The adoption of DevOps is being driven by factors such as:  
  
 **1.** Use of agile and other development processes and methodologies

**2.** Demand for an increased rate of production releases from application and business stakeholders

**3.** Wide availability of virtualized and cloud infrastructure from internal and external providers

**4.** Increased usage of data centred automation and configuration management tools

**5.** It has also been suggested that side-effect of the dominant, traditional US management style (“the Sloan model” vs. “the Toyoda model”) guarantees the development of silos of automation, thus creating “the DevOps gap” that then requires DevOps capabilities to address

Benefits of DevOps:

♣Better align IT responsiveness and capabilities to business needs

♣Produce smaller, more frequent software releases

♣Reduce effort with software development, transition and operation

♣ Improve time to market

♣ Improve quality of code

♣ Improve quality of software deployments

♣Reduce cost of product iteration and delays

♣ Instils a culture of communication and collaboration

♣ Improve productivity

♣ Improve visibility into IT requirements and processes

**3.0 DevOps Used**

1. Version Management – **Git and GitHub**
2. Build Management **- Travis CI and Heroku**

C) Test Management - **Selenium**

­­­­­­­­­­­

**4.0 VERSION MANAGEMENT**

1. Tool Used

 **Git and GitHub -** A free and open-source distributed, SCM (source code management) version control system designed to handle everything from small to very large projects with speed and efficiency. It’s the most commonly used version-control system for tracking changes in any set of files, so you have a record of what has been done, and you can revert to specific versions should you ever need to. It also makes collaboration easier, allowing changes by multiple people to all be merged into one source. It relies on the basis of distributed development of software where more than one developer may have access to the source code of a specific supplication and can modify changes to it, which may be seen by other developers. It allowed us to have “versions” of the project.

This is especially useful when we consider disposable experimentation; notably, when you push to a remote repository, you do not have to push all of your branches. You can choose to share just one of your branches, a few of them, or all of them. This tends to free people to try new ideas without worrying about having to plan how and when they are going to merge it in or share it with others.

GitHub is the code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

1. Version Management/Branching Strategy

It allowed us to work on a line that ran parallel to the main project files. It allowed us to make changes in the project without affecting the original version. The master branch of the version contained the production quality code. New features were tested and worked upon on the branches and further, and merged with the master branch. Branching and merging was easily done with the help of a few Git commands.

Our branching strategy is the GitFlow branching strategy relies on two long-lived branches and some short-lived ones. The permanent ones are good old master and the new kid on the block, “development.” The state of “master” should always be pristine; it reflects the last “good,” stable version that’s in production. “Development,” on the other hand, is always potentially unstable. This is the branch where development happens by the use of the supporting branches like feature branches, release branches and hotfix branches.

1. Features of Tool Used

**Distributed System -** It held a Central repository that could be accessed by all remote collaborators. Having a Central Server results in a problem of Data Loss or Data disconnection in case of a system failure of the central server. To tackle such kind of a situation, we used Git to mirror the whole repository. In case the central server crashed, the copy of repositories could be gained back from the users who had downloaded the latest snapshot of the project.

**Compatibility –** It is compatible with all the operating systems so it was easier for us as users to work on different platforms.

**Speed** – Since Git stores all the data related to a project in the local repository by the process of cloning, it was very much efficient to fetch data from the local repository instead of doing the same from remote repository.

1. Benefits Derived

**Performance –** It performed very strongly and reliably when compared to other version control systems. New code changes could be easily committed, version branches could be effortlessly compared and merged, and code could also be optimized to perform better.

**Flexibility** – A key design objective of Git is the kind of flexibility it offered to

handling small scale project.

­­­­­­­­­­­­­­­­

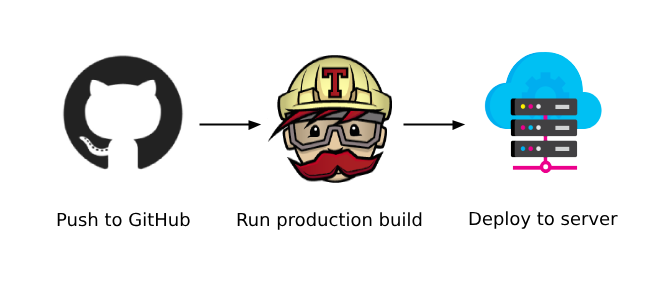
**5.0 BUILD MANAGEMENT**

1. Tool Used

**Travis CI** – It is a hosted continuous integration service used to build and test software projects hosted at GitHub and Bitbucket. It provides services to open – source projects. It provides custom deployments of a proprietary version on the customer’s own hardware. It is available for Linux, macOS and Windows. There is no installation required and you can get started by simply signing up and adding a project. Travis CI is configured by adding a file named .travis.yml, which is a YAML format text file, to the root directory of the repository. This file specifies the programming language used, the desired building and testing environment and various other parameters.

When Travis CI has been activated for a given repository, GitHub will notify it whenever new commits are pushed to that repository or a pull request is submitted. It can also be configured to only run for specific branches. It will then check out the relevant branch and run the commands specified in .travis.yml, which usually build the software and run any automated tests. When that process has completed, Travis notifies the developer(s) in the way it has been configured to do so. In the case of pull requests, the pull request will be annotated with the outcome and a link to the build log, using a GitHub integration. It can be configured to run the tests on a range of different machines, with different software installed and supports building software in numerous languages

1. Features of Tool Used



**Quick Setup** – We could easily login to our cloud platform and tell Travis CI to test the project, and then push.

**Platform Support** – Many databases and services are pre-installed and can be enabled in the build configuration.

**Pull Request Testing** – Made sure every pull request to our project is tested before it’s merged.

**Deployment** – Updating, staging or production as soon as our tests pass is seamless.

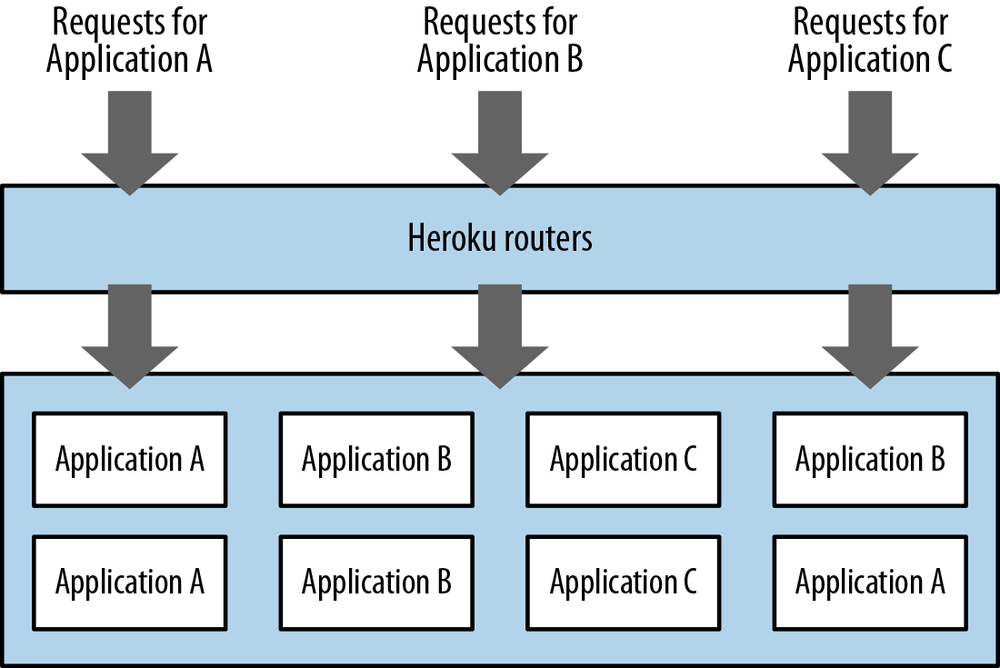
1. Benefits Derived

* It takes very less time to get started. Create a config file and start integrating
* Flexible to use
* Integration with GitHub and cloud
* Unlimited open source projects with full functionality
* Extensive project configuration via “.travis.yml” file
* Allowed cluster tests and ran them in parallel

1. Tool Used

**Heroku** – It is a cloud based application deployment and management service. The Heroku platform is broken up into several key segments; the most important among these are: routers, which ensure your application receives the requests from users on the Web and **dynos**. These are smart containers that make up the basis of Heroku’s container based design system. It runs application inside various dynos and each dyno is separated from each other.

1. Features of Tool Used



**Runtime** – The apps runs inside smart containers in a fully managed runtime environment, we handle everything critical for production – configuration, orchestration, load balancing, failovers, logging, security and more.

**Scale** – It scales in an instant, both vertically and horizontally. We can elegantly run everything.

**Code and Data Rollback** – It’s build system service lets us roll back our code or the database to a previous state in an instant.

**GitHub Integration** – The seamless GitHub integration meant every pull request spun up a disposable Review App for testing, and any repo could be set up to auto – deploy with every GitHub push to the branch of our choosing.

**Data Clips** – It made it easy to keep everyone in the loop with up-to-the-second data insights from our project by sharing query results.

1. Benefits Derived

* Allowed us to focus on code instead of infrastructure
* Enhanced the productivity of cloud app development team
* Offered single billing for all projects broken down by team
* Monitor and enhance performance through rich application monitoring
* Helped our development
* Simple horizontal and vertical scalability
* Leading platform tools and services ecosystem
* Fast application lifecycle management and permissions
* Allowed us to remove friction from the development
* A bunch of supportive tools
* Beginner and start-up friendly
* Medium learning curve

­­­­­­­­­­­­­­­­­­­­­­

* 1. **TEST MANAGEMENT**

1. Tool Used

**Selenium** – Selenium is a portable framework for testing web applications. Selenium provides a playback tool for authoring functional tests without the need to learn a test scripting language (Selenium IDE).

It also provides a test domain-specific language (Selenese) to write tests in a number of popular languages like C#, Java, PHP, Python, Ruby and Scala. The tests can then be run against modern web browsers.

It is basically used to automate the testing across various web browsers. It supports various browsers like Chrome, Mozilla, Firefox, Safari, and IE, and you can very easily automate browser testing across these browsers using Selenium WebDriver.

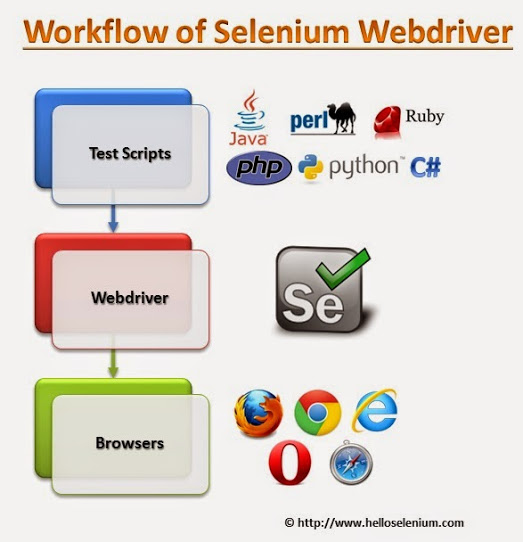
B)Features of Tool Used

\*Selenium is an open source and portable Web testing Framework.

\*Selenium IDE provides a playback and record feature for authoring tests without the need to learn a test scripting language.

\*It can be considered as the leading cloud-based testing platform which helps testers to record their actions and export them as a reusable script with a simple-to-understand and easy-to-use interface.\*Selenium supports various operating systems, browsers and programming languages.

Following is the list: Programming Languages{C#, Java, Python, PHP, Ruby, Perl, and JavaScript}; Operating Systems{Android, iOS, Windows, Linux, Mac, Solaris} and Browsers{Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari}.



\*It also supports parallel test execution which reduces time and increases the efficiency of tests and requires fewer resources as compared to other automation test tools.

It can be integrated with frameworks like Ant and Maven for source code compilation as well as with testing frameworks like TestNG for application testing and generating reports.

\*Selenium web driver does not require server installation, test scripts interact directly with the browser.

\*Selenium commands are categorized in terms of different classes which make it easier to understand and implement.

\*Selenium Remote Control (RC) in conjunction with WebDriver API is known as Selenium 2.0. This version was built to support the vibrant web pages and Ajax.

C)Benefits Derived

* Language and Framework Support
* Multi-Browser Support
* Support Across Various Operating Systems
* Ease Of Implementation
* Reusability and Integrations
* Flexibility
* Parallel Test Execution and Faster Go-to-Market

­­­­­­­­­­­­­­­­­­­­­­

**7.0 CONCLUSION AND FUTURE SCOPE**

In conclusion, we have readied a web application that shall contain up to date information of the schools and colleges/universities where this application may be deployed. It would substantially improve the efficiency of the educational institution’s record management ethics.

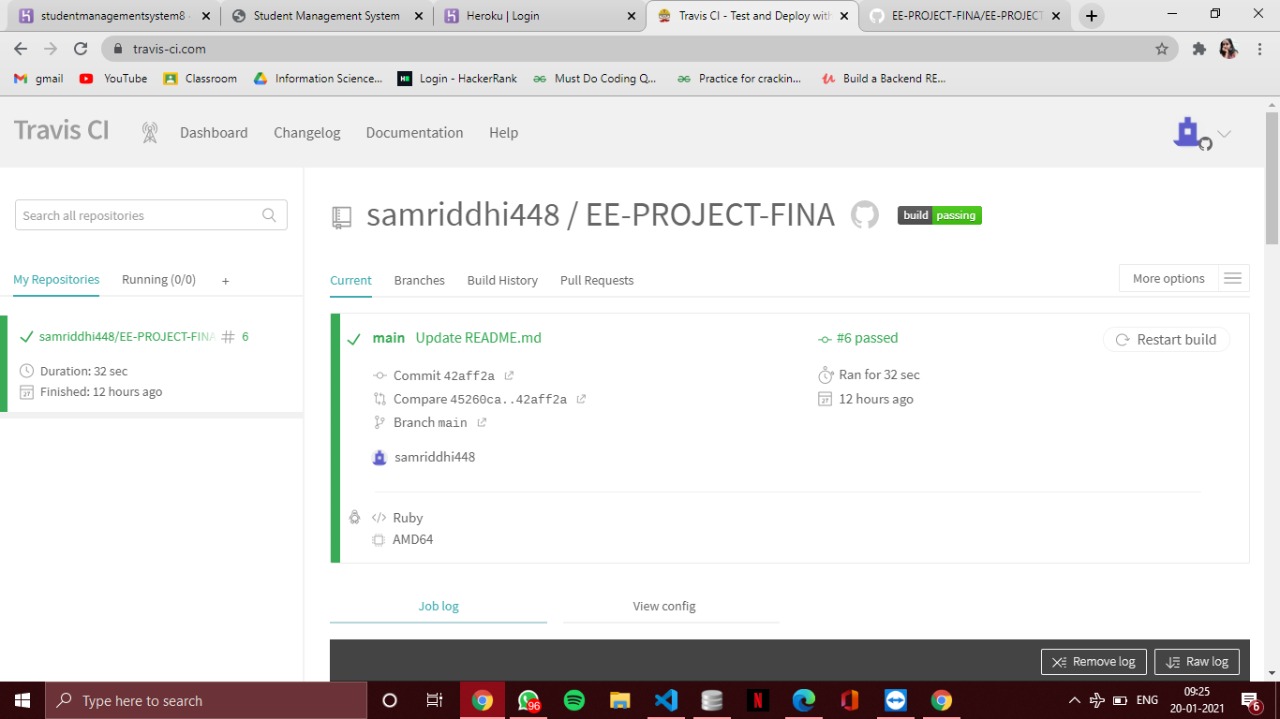
Moving forward, it will aid the institute with:

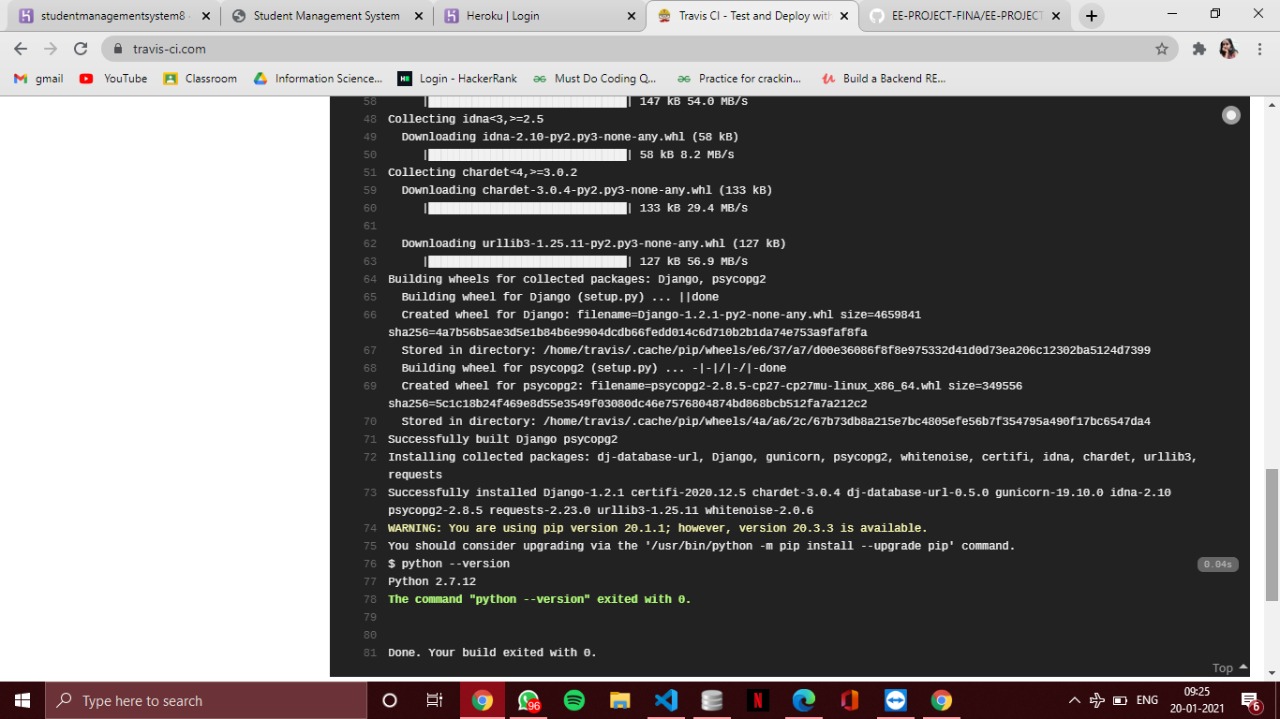
* Providing the online interface for students, faculty etc.
* Increasing the efficiency of college record management
* Decreasing the time required to access and deliver the student records
* Making the system more secure
* Decreasing the time spent on non-value tasks.

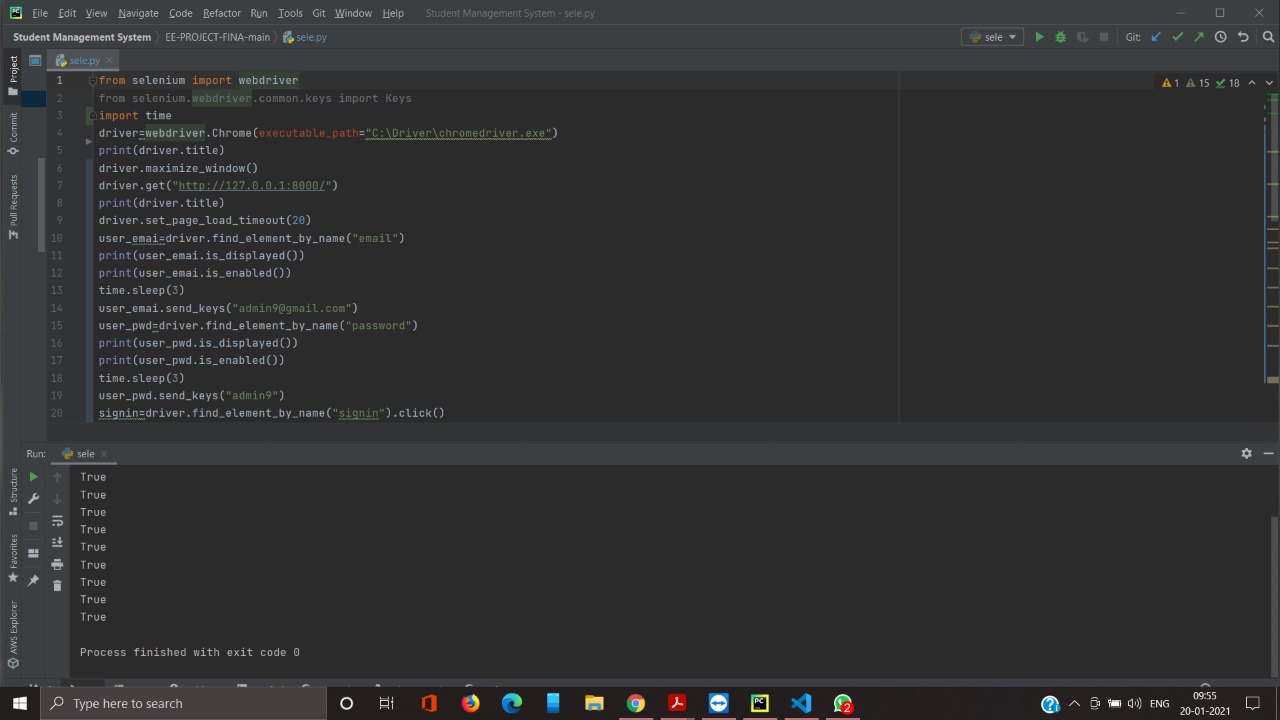
The features that are desired to be integrated and are in our future scope include:

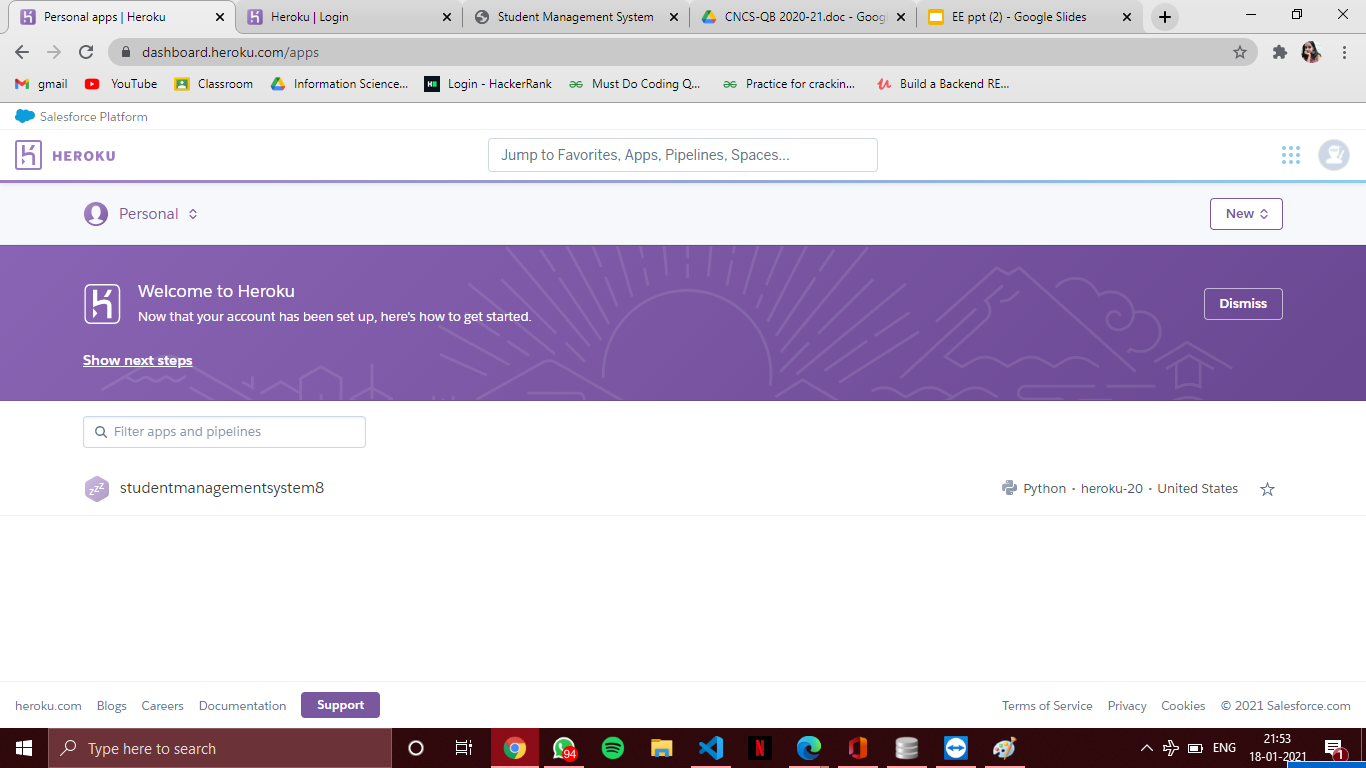
* Dynamic chat-bot that interacts with users of our webapp and solves their queries by providing relevant and accurate results.
* Video-conferencing feature that will allow students, staff and the heads of various departments to communicate on a tailor made platform.
* Display of students’ certificates at the end of each course and the downloading of the same for official purposes.

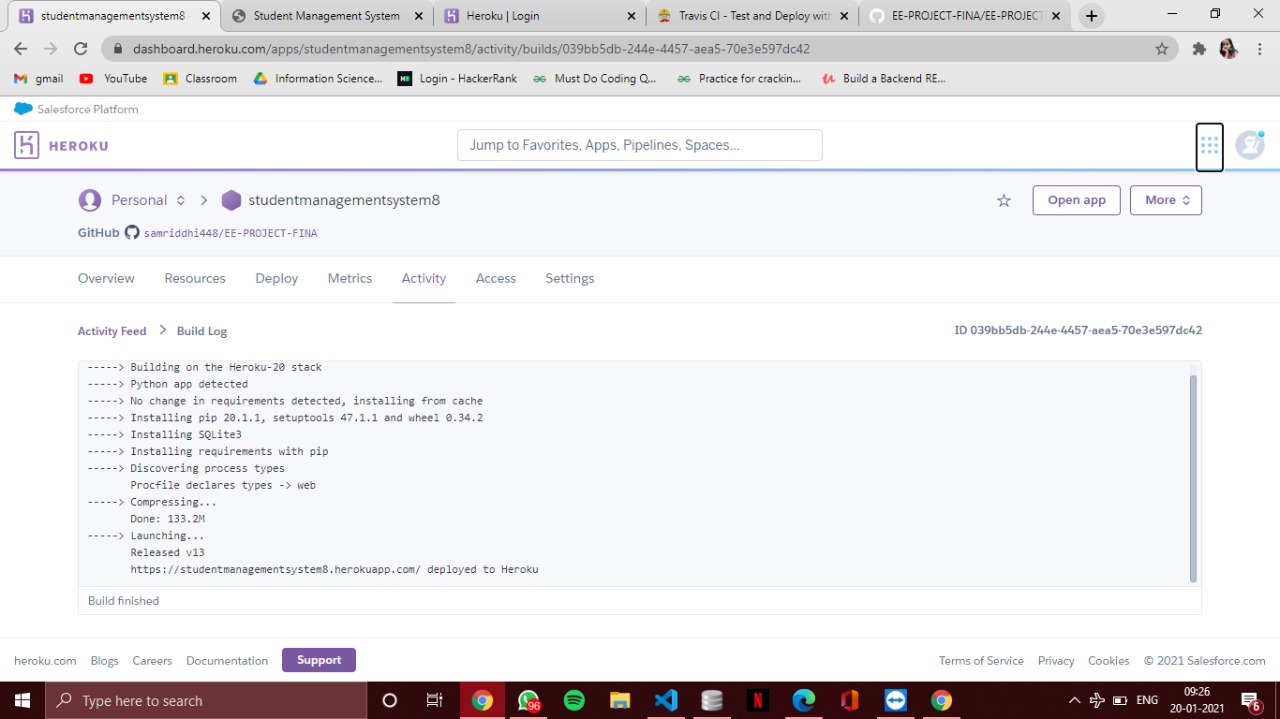
CODE OVERVIEW:











**ANNEXURE**