Student Management Online Portal

Project Stage 2 Report

Submitted By

Aman Behl - 1714110716

Pooja Ghadge - 1500001213

Sourabh Singh - 1714110715

In partial fulfilment of the requirement

For the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER ENGINEERING

Under the guidance of

Prof. S.B.Nikam

DEPARTMENT OF COMPUTER ENGINEERING

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY), COLLEGE OF ENGINEERING, PUNE- 43

2020-21

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY), COLLEGE OF ENGINEERING, PUNE- 43

CERTIFICATE

This is to certify that the Project Stage1 report titled Student Management Online Portal, has been carried out by the following students in partial fulfilment of the degree of BACHELOR OF TECHNOLOGY in Computer Engineering of Bharati Vidyapeeth (Deemed to be University) Pune, during the academic year 2018-2019.

Team:

1. Aman Behl
2. Sourabh Singh
3. Pooja Ghadge

Prof. S.B Nikam (Project Guide) Dept. of Computer Engineering
Place: Pune
Date:

Bharati Vidyapeeth (Deemed to be University), College of Engineering, Pune- 43

ACKNOWLEDGEMENTS

Before we commence forward with our Project, we would collectively like to express our gratitude to the College and to the Computer Department of Bharati Vidyapeeth University, College of Engineering for providing the students with their constant support and guiding us forever on the path towards knowledge.

I would also like to take this special opportunity to thank, first and foremost, Dr. S.B.Vanjale, our head of department for being a constant source of encouragement and mentoring us in authoring our Project.

We would also like to thank Prof. S.B.Nikam, project guide for helping us in every step along the way. Your insightful remarks really challenged us to think and helped us bring out the best in us by constantly reviewing our reports and providing invaluable feedback.

We would also like to express our gratitude to the Principal of our college, Prof. Dr. Anand Bhalerao for giving us this wonderful opportunity of creating this dynamic Project by giving us access to the resources at all times and allowing us to use the College infrastructure and enlightening us with his kind words.

Lastly, we would like to thank our friends and family for constantly supporting us in our endeavours and being with us, gracing us with their blessings and affection. The completion of this project would never have been possible without the help of any of the above mentioned people.

Bharati Vidyapeeth (Deemed to be University), College of Engineering, Pune- 43

ABSTRACT

The mission of the Student Information Online Portal is to create an integrated information technology environment for students and administration. Our goal is to focus on services and integration for end users. It is a web based self service environment for students, prospective students , an informative environment for all levels of faculty and staff to take the reports and the information of the student , data extraction and performance analysis. It is mainly useful for educational establishments to manage student data which also facilitates all individual associated information. It provides capabilities for entering student results and other assessment scores, and managing many other student-related data needs in a college. Our easy-to-use, integrated college student portal application would be used to reduce time spent on administrative tasks, as to concentrate on other skillful practical activities. It can accept, process and generate reports at any given point of time accurately.

Table of Contents

CHAPTER 1
INTRODUCTION

The design and implementation of the student online portal system is to replace the current paper-based records . College Faculties are able to directly access all aspects of a student's academic progress and various activities of students through a secure, online interface . 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 a student user interface, allowing users to access information and submit requests online thus reducing processing time. All data is stored on servers managed by the college administrator and ensures the highest possible level of security. The system features a 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. This system provides a simple interface for the maintenance of student information. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming. All these problems are solved using student information management system:-Providing the online interface for students, faculty etc., Increasing the efficiency of college record management, Decrease time required to access and deliver student records, To make the system more secure, Decrease time spent on non-value added tasks. Students are the main asset for various universities. Universities and students play an important role in producing graduates of high qualities with academic performance achievement. Academic performance achievement is the level of achievement of the students' educational goal that can be measured and tested through examination, assessments and other forms of measurements. However, the academic performance achievement varies as different kinds of students may have different levels of performance achievement.

The student academic performance is usually stored in the student management system, in different formats such as files, documents, records, images and other formats. These available students' data could be extracted to produce useful information. However, the increasing amount of students' data becomes hard to be analysed by using traditional statistical techniques and database management tools . Thus, a tool is necessary for universities to extract useful information. This useful information could be used to predict the students' performance.

The project is designed to help the teachers and students manage their college activities. It consists of relational databases of students courses of the entire university. Using these databases, various functions that include marks management,performance analysis are provided. Within attendance management, a teacher can enter marks and see the performance status of each student for each course with their respective PRNs. Similar to login details Internal and Semester end marks can also be entered for each student.

This student performance analysis system provides an easy way to students in searching the details of projects, academic report and marks/percentage details with the graph. All the details of the projects and details of student’s attendance and marks are added by the Student respectively . It has Modules namely Student and teacher. Students are asked to register and then login, fills the academic details,view marks in graph-based format, events and notice. Teachers can login, approve students, view assigned events and notice. Basically, this system can help to maintain and manage the records of students, teachers, events of college, academics and more.

1.1 SECURITY ISSUES WITH EXISTING AUTHENTICATION MODELS

Normal Email Login authentication based on the DB where we store the email and send the unique ID according to that but we can’t verify the user if it is coming with the correct email or not , if the user is verified or not .

So we came up and tried a Firebase google authentication where users can simply login with Google .

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

To sign a user into your app, you first get authentication credentials from the user. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Then, you pass these credentials to the Firebase Authentication SDK. Our backend services will then verify those credentials and return a response to the client.

After a successful sign in, you can access the user's basic profile information, and you can control the user's access to data stored in other Firebase products. You can also use the provided authentication token to verify the identity of users in your own backend services.

1.1.2 POSSIBLE ATTACK STRATEGIES

If you allow user to Login through the database if some one can access your Database which is not secured can do phishing attacks.

1. Phishing

Phishing is when a hacker posing as a trustworthy party sends you a fraudulent email, hoping you will reveal your personal information voluntarily. Sometimes they lead you to fake "reset your password" screens; other times, the links install malicious code on your device.

2. Credential stuffing

If you've suffered a hack in the past, you know that your old passwords were likely leaked onto a disreputable website. Credential stuffing takes advantage of accounts that never had their passwords changed after an account break-in. Hackers will try various combinations of former usernames and passwords, hoping the victim never changed them.

3. Keyloggers

Keyloggers are a type of malicious software designed to track every keystroke and report it back to a hacker. Typically, a user will download the software believing it to be legitimate, only for it to install a keylogger without notice.

1.1.2 Security and Usability Issues with Existing Authentication Technique

Security of our authentication system is a Token based login System if User comes up with the authorized token Provided by the firebase then the user will be allowed to enter the site and make requests and stuff .

If we have our own Database hosted server, we can be assured that every personal data is safe and we can use the normal login with the DB id’s login and we have to generate the criteria with the token provided by our backend technologies.

If we are moving forward with the Google Login authentication system it will provide the normal google authentication system, which will provide its own token.

1.2 MOTIVATION

Main motivation in developing this system. The most important one in that the process of a non virtual application portal is a redundant and very time-consuming process that will take place many times during each semester, which forms a challenge to the administration especially with all other commitments. Also, teachers requires extra time and patience for all this process. The necessity to change and update the information knowledge bases to guarantee versatilities is very important for examiners and the process of keeping old questions registered to avoid redundancy is a very important task. The storing and utilization of generating work by adding all new updated and modified questions to questions knowledge base and will help saving organization experience and knowledge.

1.3 AIM OF RESEARCH

The objective of Student Information Management System is to allow the administrator of any organization the ability to edit and find out the personal details of a student and allows the student to keep up to date his profile. It'll also facilitate keeping all the records of students, such as their id, name, mailing address, phone number, DOB etc. So all the information about a student will be available in a few seconds. Overall, it'll make Student Information an easier job for the administrator and the student of any organization.

The main purpose of this project is to illustrate the requirements of the Student Information Management System and is intended to help any organization to maintain and manage personal data. It is a comprehensive project developed from the ground up to fulfill the needs of colleges as they guide their students. This integrated information management system connects daily operations in the college environment ranging from performance management to communicational means among students and teachers. This reduces data error and ensures that information is always up-to-date throughout the college. It provides a single source of data repository for streamlining your processes and for all reporting purposes. It has a simple user interface and is intuitive. This insures that the users spend less time in learning the system and hence, increase their productivity. Efficient security features provide data privacy and hence, increase their productivity.

1.5 Problem with authentication

Problem of Authentication-

Passwords in the clear

    a. HTTP Basic authentication was designed when telnet passwords-in-the-clear were the norm
    b. it protects against casual eyeballing by base64-encoding credentials,
    c. this provides no obstacle to an attacker that can write even trivial amounts of software

- TLS/SSL is compute-intensive and hence cost-prohibitive for many applications
    a. encrypting all the content is overkill if the problem is just user authentication
- HTTP authentication headers are not expressive enough to effectively convey a request for credentials
    a. they provides only a cryptic "realm" string to convey the identity of the relying party; this is an unacceptable way to communicate a trusted identity or brand
- HTML forms with <input type="password"> obscure the password from view and suggest to the user that their password is being kept confidential when (unless TLS/SSL is also in use) it is not.
- Each service provider manages credentials separately
    a. Users are burdened with managing separate credentials for each service

        - Many choose not to

   b. New services are difficult to launch without endorsement of an established brand, concentrating power and stifling innovation.

1.4 PROBLEM STATEMENT

Student Performance Analysis provides a solution to choose among multiple students and their record and it makes it easy for the organization to access multiple students at the same time on the same platform. To keep records and access of all the students in the organization is a very conventional manner.This way is very time consuming and makes all the teachers and administrators do these same activities frequently. Our project removes these drawbacks. Administration will have a track on all the records of the students and in a professional and systematic manner.

1.5 RESEARCH METHODOLOGY

Nowadays, in schools and colleges, it is very difficult to manage each and everything manually. Supervising and maintaining the whole database of a school or college can be time-consuming and challenging especially if it’s done on a regular basis. So, we need to handle and manage everything smartly.

To solve this problem Student database analysis is used. This software makes it easy to track the progress of every department of college and automate different functions. With this application everything can be seen on a single dashboard. The administrator can manage the college from anywhere. The possibilities of maintaining the whole database of a college with this software are endless.

Some of the prominent roles of this system are:

- Manages and automates different functions.
- Helps in long-term management and planning of all departments of college.
- Eliminates the need for having multiple management software for each department.
- Activities like performance and login details of every student can be digitalized and automated.

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE REVIEW

SR.NO	YEAR	AUTHOR NAME	PAPER NAME	PAPER DESCRIPTION	PAPER LIMITATIONS
1	2018	Dipin bhudhrani	Student information management system	To create an integrated information technology environment for students and administration.	Security management
2	2013	Robert G.Rittenhouse , Junaid Aheseli Chaudry	Graphical Authentication Systems	Potential Replacement or supplement for conventional Authentication systems	Graphical password Graphical user authentication
3	2019	Hong Zhang	Information management system	Analysis of student status management	Archives management
4	2019	Abuka Victor Deju	Management system for student .Design and implementation of SIS	Eliminating difficulties encountered by the students .	Accessibility issues,hosting services
5	2015	L.Joshi	Developing an online intranet college	Intranet based application that is of	Big data , learning analytics.

			management system	importance to either an educational institution or a college administration.	
6	2012	Ismail duru	Learning analytics in MOOC	Latest development and studies about students.	Special user groups.
7	2017	Disha kambale	Academic performance of student using data mining	To access the academic performance of student using WEKA	Student login only.
8	2015	Amirah mohamed shahiri	Prediction of student performance	Performing systematic relational review	Prediction Students performance
9	2018	U.Bin Mat	Research paper on inventory management system	Operates business hardware stores	Backup inventory in limited time
10	2020	Anupam khan	Student performance analysis and prediction in classroom learning	Performance in non linear ways	Educational datasets.

3.2 REQUIREMENT SPECIFICATION :

3.2.1 Hardware Requirement:

- i3 Processor Based Computer or higher
- Memory: 4 GB
- Hard Drive: 50 GB
- Internet Connection

3.2.2 Software Requirement:

- Windows 7 or higher
- Visual Studio Code or any IDE you are comfortable with.
- Node Latest version installed
- MongoDB cloud Atlas or MongoDB compass installed on the local system
- Any Browser (Chrome or Firefox or any other to Run it on LocalHost)

Advantages -

• It helps in maintaining students' records

•It helps teachers to get the Student data .

•Easy way of getting student secured data , we don't want to write manually our data and fill it in.

Limitation

•It only works on internet

Application

•This system can be used by multiple peoples to get the Courses and Recommendation online and as well as to track their records.

3.3 Analysis of Graphical based Authentication Systems

RECOGNITION-BASED AUTHENTICATION SCHEMES :

1.) Déjà vu algorithm

In 2000 Dhamija and Perrig proposed a new graphical authentication scheme called Déjà vu algorithm, which is based on the perception of hash visualization technique. In The Act Registration phase the user is asked to choose a certain number of images from a collection of random non-describable abstract pictures generated by a system. Later, the user will be required to identify previously selected images in order to be authenticated . The average registration and login time of this approach is much longer than in the traditional text-based approach. Also the server needs to store a large number of pictures that may delay the authentication process while transferring over the network. Furthermore, the process of selecting and identifying a set of images from the picture database can be time consuming for the user.

2.) Passface algorithm

In 2000 Brostoff and Sasse from Real User Corporationproposed a new graphical authentication scheme that is calledPassface algorithm. To create a password the user will be asked to choose a certain number of images of human faces from the picture database. At the authentication phase users will be required to identify previously chosen faces in order to be authenticated. The user recognizes and clicks on the knownface, and then the procedure repeats several times. This Technique is very memorable over long time periods. However,the majority of the users tend to choose faces of people based on the obvious behavioral pattern, which makes this authentication scheme kind of predictable and vulnerable to various attacks. Also it takes longer for login and registration than in traditional text-based password schemes.

3.) TRI-PASS: OUR PROPOSED ALGORITHM

The objective of this paper is to propose a new algorithm based on two previously discussed techniques namelyPassPoint and Triangle algorithms. In our proposed algorithm we want to focus on the features and advantages of these two algorithms and join them together to get as much as possible security and usability.Registration phase: To create a password the user should choose one image from the library of pictures and then choose any three points as his future password by clicking on the image. For better perception we will call these points“password points”.

3.3.1 According to working mechanism:

Login phase: The process of verification is based on our newly proposed scheme called Tri-Pass algorithm. In Tri-Pass algorithm to authenticate, the user has to imagine an invisible triangle around the area of the first “password point” and click on any three points that will form a triangle around this “password point”. It means that in order to be authenticated the “password point” should be inside the area of invisible triangle. Then the user should do the same for the second “password point” and third “password point”.

3.3.2 According to user approach:

According to the user Approach this is defined in the way that people have to be present for the biometric login authentication and for the face recognition algorithm techniques .
So we need to define the user in what way they can Approach this way where the thing is more convenient and more reliable .

3.4 Analysis of existing authentication techniques

Authentication is the process of identifying users that request access to a system, network, or device. Access control often determines user identity according to credentials like username and password. Other authentication technologies like biometrics and authentication apps are also used to authenticate user identity.

Cybercriminals always improve their attacks. As a result, security teams are facing plenty of authentication-related challenges. This is why companies are starting to implement more sophisticated incident response strategies, including authentication as part of the process. The list below reviews some common authentication methods used to secure modern systems.

1. **Password-based authentication**

Passwords are the most common methods of authentication. Passwords can be in the form of a string of letters, numbers, or special characters. To protect yourself you need to create strong passwords that include a combination of all possible options.

However, passwords are prone to phishing attacks and bad hygiene that weakens effectiveness. An average person has about 25 different online accounts, but only 54% of users use different passwords across their accounts.

The truth is that there are a lot of passwords to remember. As a result, many people choose convenience over security. Most people use simple passwords instead of creating reliable passwords because they are easier to remember.

The bottom line is that passwords have a lot of weaknesses and are not sufficient in protecting online information. Hackers can easily guess user credentials by running through all possible combinations until they find a match.

2. **Multi-factor authentication**

Multi-Factor Authentication (MFA) is an authentication method that requires two or more independent ways to identify a user. Examples include codes generated from the user’s smartphone, Captcha tests, fingerprints, voice biometrics or facial recognition.

MFA authentication methods and technologies increase the confidence of users by adding multiple layers of security. MFA may be a good defense against most account hacks, but it has its own pitfalls. People may lose their phones or SIM cards and not be able to generate an authentication code.

3. **Certificate-based authentication**

Certificate-based authentication technologies identify users, machines or devices by using digital certificates. A digital certificate is an electronic document based on the idea of a driver’s license or a passport.

The certificate contains the digital identity of a user including a public key, and the digital signature of a certification authority. Digital certificates prove the ownership of a public key and issued only by a certification authority.

Users provide their digital certificates when they sign in to a server. The server verifies the credibility of the digital signature and the certificate authority. The server then uses cryptography to confirm that the user has a correct private key associated with the certificate.

4. **Biometric authentication**

Biometrics authentication is a security process that relies on the unique biological characteristics of an individual. Here are key advantages of using biometric authentication technologies:

- Biological characteristics can be easily compared to authorized features saved in a database.
- Biometric authentication can control physical access when installed on gates and doors.
- You can add biometrics into your multi-factor authentication process.

Biometric authentication technologies are used by consumers, governments and private corporations including airports, military bases, and national borders. The technology is increasingly adopted due to the ability to achieve a high level of security without creating friction for the user. Common biometric authentication methods include:

Facial recognition—matches the different face characteristics of an individual trying to gain access to an approved face stored in a database. Face recognition can be inconsistent when comparing faces at different angles or comparing people who look similar, like close relatives. Facial liveness technology prevents spoofing.

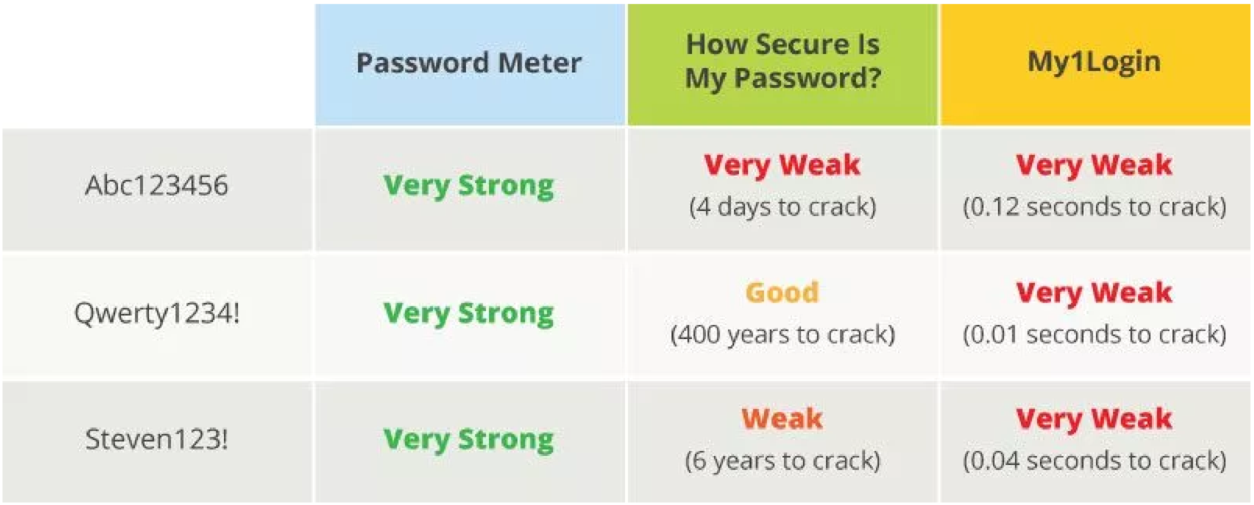
- Fingerprint scanners—match the unique patterns on an individual's fingerprints. Some new versions of fingerprint scanners can even assess the vascular patterns in people's fingers. Fingerprint scanners are currently the most popular biometric technology for everyday consumers, despite their frequent inaccuracies. This popularity can be attributed to iPhones.
- Speaker Recognition —also known as voice biometrics, examines a speaker's speech patterns for the formation of specific shapes and sound qualities. A voice-protected device usually relies on standardized words to identify users, just like a password.
- Eye scanners—include technologies like iris recognition and retina scanners. Iris scanners project a bright light towards the eye and search for unique patterns in the colored ring around the pupil of the eye. The patterns are then compared to approved information stored in a database. Eye-based authentication may suffer inaccuracies if a person wears glasses or contact lenses

5. **Token-based authentication**

Token-based authentication technologies enable users to enter their credentials once and receive a unique encrypted string of random characters in exchange. You can then use the token to access protected systems instead of entering your credentials all over again. The digital token proves that you already have access permission. Use cases of token-based authentication include RESTful APIs that are used by multiple frameworks and clients.

3.5 Problems with existing system

1. User-Generated Credentials
How do user-generated credentials pose a threat?
Since users have to create their own passwords, there's always a chance they won't create secure credentials. In fact, around 90% of user-generated passwords are considered weak and easily vulnerable to hacking.
Whether it's because users want to have a password that's easy to remember, they aren't up to date on password security best practices, or subconsciously (and even consciously) use patterns to generate their passwords, this type of authentication has its flaws.
Even if your website is equipped with a password strength-checking tool, (like the ones below) the results can be inconsistent and inaccurate, often leading users into a false sense of security.



	Password Meter	How Secure Is My Password?	My1Login
Abc123456	Very Strong	Very Weak (4 days to crack)	Very Weak (0.12 seconds to crack)
Qwerty1234!	Very Strong	Good (400 years to crack)	Very Weak (0.01 seconds to crack)
Steven123!	Very Strong	Weak (6 years to crack)	Very Weak (0.04 seconds to crack)

Creating complex passwords is difficult because our minds are drawn towards patterns. Think about the passwords you create. They likely all follow a similar formula, like using a word with numbers and a special character at the end.
While these patterns make it easier for us to remember our credentials, cybercriminals are also aware of the common formulas people use to create passwords. As a result, hackers can use this knowledge to adjust how their brute-force systems run through password combinations or crack your password making an educated guess.

And the above system makes sense only we have our own hosted server of thousands worth, so in order to avoid the things of the thing what we are doing is we are letting allow the user to come on login with the token only.
How do brute-force attacks pose a threat?

Simply put, a brute-force attack occurs when a computer program runs through every password combination until they find a match. The system will run through all one-digit combinations, two-digit combinations, and so forth until it cracks your password. Some programs specifically focus on combing through the most commonly used dictionary words, while others target popular passwords against a list of possible usernames.

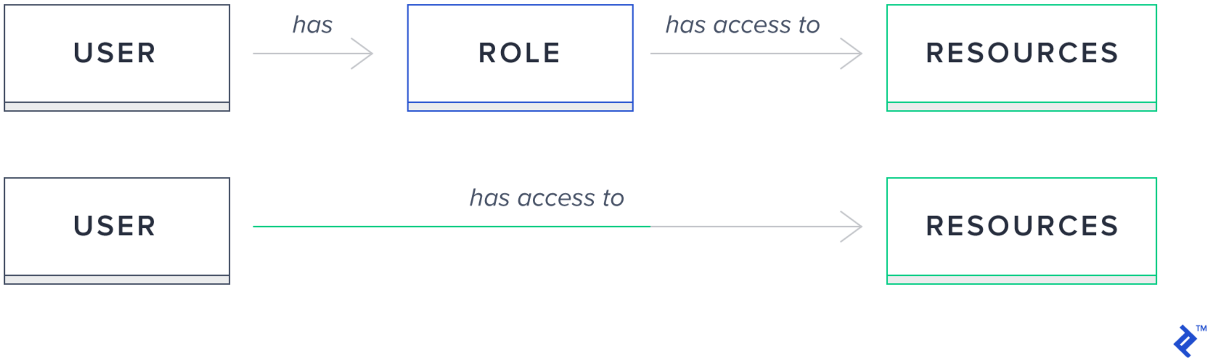
As technology evolves, so do the tools hackers use to crack people's credentials. Aside from merely guessing your password, a brute-force attack is the most common technique hackers use.

To make matters worse, these systems are able to run through thousands of combinations in less than a second, which means that shorter passwords can be cracked in a very short time frame.

Almost every app requires some level of authorization system. In some cases, validating a username/password set with our Users table is enough, but often, we need a more fine-grained permissions model to allow certain users to access certain resources and restrict them from others. Building a system to support the latter is not trivial and can be very time consuming. In this tutorial, we'll learn how to build a role-based auth API using Firebase, which will help us get quickly up and running.

Role-based Auth

In this authorization model, access is granted to roles, instead of specific users, and a user can have one or more depending on how you design your permission model. Resources, on the other hand, require certain roles to allow a user to execute it.



Firebase Authentication

In a nutshell, Firebase Authentication is an extensible token-based auth system and provides out-of-the-box integrations with the most common providers such as Google, Facebook, and Twitter, among others.

It enables us to use custom claims which we'll leverage to build a flexible role-based API.

We can set any JSON value into the claims (e.g., { role: 'admin' } or { role: 'manager' }).

Once set, custom claims will be included in the token that Firebase generates, and we can read the value to control access.
It also comes with a very generous free quota, which in most cases will be more than enough.

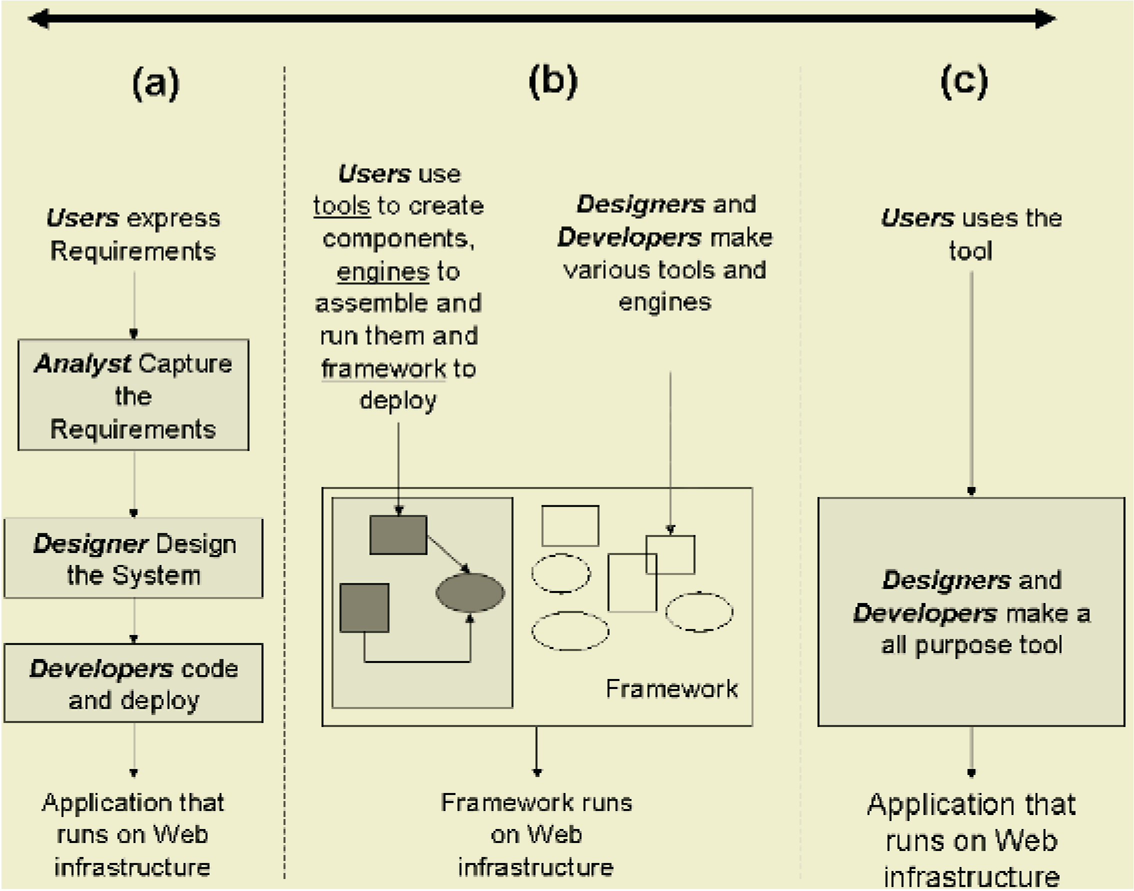
3.6 Analysis of proposed system

We analyze that when a user approaches a very feasible and maintainable content , they will be given time in that authentication system in which they require less-time with a simple Login popup and he is authorized .

3.6.1 Analysis of problem statement

3.6.2 System position statement

3.6.3 User Environment


(a)
(b)
(c)
Users express Requirements
Analyst Capture the Requirements
Designer Design the System
Developers code and deploy
Application that runs on Web infrastructure
Users use tools to create components, engines to assemble and run them and framework to deploy
Designers and Developers make various tools and engines
Framework
Framework runs on Web infrastructure
Users uses the tool
Designers and Developers make a all purpose tool
Application that runs on Web infrastructure

Development environments

A development environment is where most of the technical programming takes place. These environments are used by developers to build applications and write their code, from platform-specific native apps to progressive web-apps and cross-platform apps. Generally, development environments are set up on local PC's and facilitated by a Git repository. A development environment is generally not accessible to the end-users and customers.

Staging environments

The staging environment is the place to be after most of the technical development has taken place in the development environment. This type of environment is typically used by developers to test out the code they've written and see if the application, including all its functionalities, works properly in a realistic setting. The staging environment is also often called the testing environment. Besides testing the code themselves, the staging environment is also suitable for demonstrating newly finished work to the product owner, or as a way to let end-users test the product and receive valuable feedback before pushing it to the production environment. Although the staging environment is officially already live and running, only people with the correct URL can access the site. Usually, these pages are protected by a login wall, in case the link falls into the wrong hands.

Production environments

When the development team, product owners, and end-users have thoroughly tested the product in the testing environment, the product is pushed to the production environment. The production environment is the final environment in the software development process, and should be used for the official launch of the app, web-app, or website. Generally, the production environment isn't much different than the staging environment. The main difference is that the production environment contains live data and is actively used by end-users. It is therefore easier to access for the public than the staging environment.

3.7 System perspective

A. Desktop Applications vs. Web Applications

- A desktop application is a program that runs standalone in a computer device, like a desktop computer, laptop, or even a mobile device (PCMag, n.d.). All the components of these applications are located within the device, and these components are executed within a single address space of the system memory.

- The components of the web applications are distributed; hence the system is executed in multiple address spaces. PCMag's definition for the web application says that a web application is "an application in which all or some parts of the software are downloaded from the Web each time it is run. It may refer to browser-based apps that run within the user's Web browser, or to 'rich client' desktop apps that do not use a browser, or to mobile apps that access the Web for additional information" (PCMag, n.d.). We think that the "rich client applications" or the "Rich Internet Applications" have their own domain, step above the domain of the web applications. Therefore, according to the definition, we can interpret that the scope of the web applications is limited to the browser-based applications.

B. Evolution of Web Applications

Early systems, which utilized the service of web were called “Web sites”, and they were limited to a collection of documents with static content, interconnected via hyperlinks (Jazayeri, 2007). These documents were located in a web server, and the user can request for these documents using a web client application, mainly the web browser. Upon users' needs the web browser sends requests to the server, and displays the received responses to the user.

Later these web sites incorporated server-side development languages like PHP or JAVA, and related TTs; as well as client-side processing languages such as JavaScript (JS), and associated TTs. With all these TTs web sites were evolved to web applications, with server-side and/or client-side application components, which are capable of processing and producing dynamically tailored information (Shklar & Rosen, 2003). With continuous evolvements, the web applications nowadays provide a platform to deliver features of some other services – which can be gained from data networks, such as e-mails, and file transferring – via web protocols. With all these features, the service of web and the web applications have become advanced, powerful, and a complex set of entities, with a high demand.

A web application can serve multi-users in parallel, with less hassle compared to the desktop applications. As Conallen (Conallen, 1999) says, “One of the most significant advantages of a web application is its deployment”. He continues saying that “deploying a web application is usually a matter of setting up the server-side components on a network. No special software or configuration is required on the part of the client”. In such setting, web applications are easy to manage, maintain, and modify, with a low workload compared to the higher number of users it can serve simultaneously. In clients' perspective, no special software or configuration is required; and for the users, the web applications provide both platform and location independent access. In early stages, there were some device dependencies for the web applications regarding the mobile devices; however, the modern TTs are minimizing these barriers and evolving rapidly.

New types of system like mobile applications (Techopedia, n.d.), service oriented systems (Bianco, et al., 2011), and Internet of Things (Agenda, 2016) have become popular, which use the service of web for communication. We noted that these types of applications are not covered by the definition of the “web application” and they can be seen as much complex systems than the standard traditional web applications. However, they all share similar architectural characteristics within the domain of web.

There are several architectural styles available for the systems, which utilize the service of web: client-server or two-tier architecture, three-tier architecture, multi-tier architecture, and Service Oriented Architecture(SOA) are some regularly used styles. All these styles are based on the basic client-server model and then are extended up to complex multi-tier or SOA models, to serve the needs of the advanced user requirements. The main characteristic of these styles is, they partition the system into different layers, where these layers contains different components. These different components in the system, run in multiple locations and environments (mainly – and at least – in the client and the server), and the development of these components are heavily TTs dependent. This setting initiates generating the complexity in web applications; these facts – location and technology dependencies in development, complex structures, variable size, and advanced features and capabilities – together have made the systems, which use the service of web, a very advanced and complex.

III. THE CONCEPT OF WEB-BASED APPLICATIONS

It was discussed in the section II.A, the term “web application” covers the basic notion of browser based client-server applications. As discussed in the section II.B, in present, the use of the service of web has expanded its limits, beyond the generic web applications, and beyond the web browser. Therefore, we think that the term “web application” or its definition are not capable of addressing all these systems within a wider range. However, as discussed in the later part of the section II.B, still the architecture of these systems are based on the client-server style, using the request-respond model over HTTP; thus, they can be grouped into a single domain. We propose the term “Web-based Applications” to refer all these applications, which are based on the client-server model and utilize the service of web for communication. This term can be seen as an umbrella term, which covers a wider range of systems than the term “web application”, both in and beyond the browser, addressing a superset of standard web applications.

During the literature survey, we identified an available definition for the term “web-based applications”. The Techopedia (inc., 2017) defines the web-based applications as: “A web-based application is any program that is accessed over a network connection using HTTP, rather than existing within a device’s memory. Web-based applications often run inside a web browser. However, web-based applications also may be client-based, where a small part of the program is downloaded to a user’s desktop, but processing is done over the internet on an external server. Web-based applications are also known as web apps.” They further explain that “there is a lot of confusion created by the use of terms like web-based, internet-based and cloud-based when referring to applications. Web-based applications actually encompass all the applications that communicate with the user via HTTP. This includes light applications like Flash games, online calculators, calendars, and so on, as well as more intensive applications such as word processors and spreadsheet applications" (inc., 2017). This definition and the explanation widen the general notion of the web applications, and we think that this definition can be improved by constructing it in an architectural point of view, allowing it to have more space to be extended by merging variety of related concepts into the domain.

A. A Contemporary Definition for the Web-based Applications

Before defining the web-based applications, we had better understand the formalism of them. As stated in section II.A, a desktop application is an application that runs standalone in a desktop/laptop computer, mobile device (PCMag, n.d.), or even in any other device as an embedded application. Remote components can be added to the standalone applications, utilizing web technologies for communication, data processing, storing, etc. Such systems, which utilizes the service of the web, can be considered as the "Web-based Applications". Note that the web browser itself is a desktop application, which utilizes the service of web.

Based on this discussion, we propose to define the web-based applications as follows. This definition also specifies the scope of the capabilities of web-based applications.

Web-based application is a system, with application component(s) in client-side [client-component(s)], which communicate(s) with application component(s) in a web server [server-component(s)], for processing data. They utilize the service of web, based on the client-server architecture, request-response model, standard HTTP, and other related techniques and technologies.

Elaborating this definition further, we suggest to consider this basic model of a web-based application as a "Standalone Web-based Application". This concept is presented by the formula below.

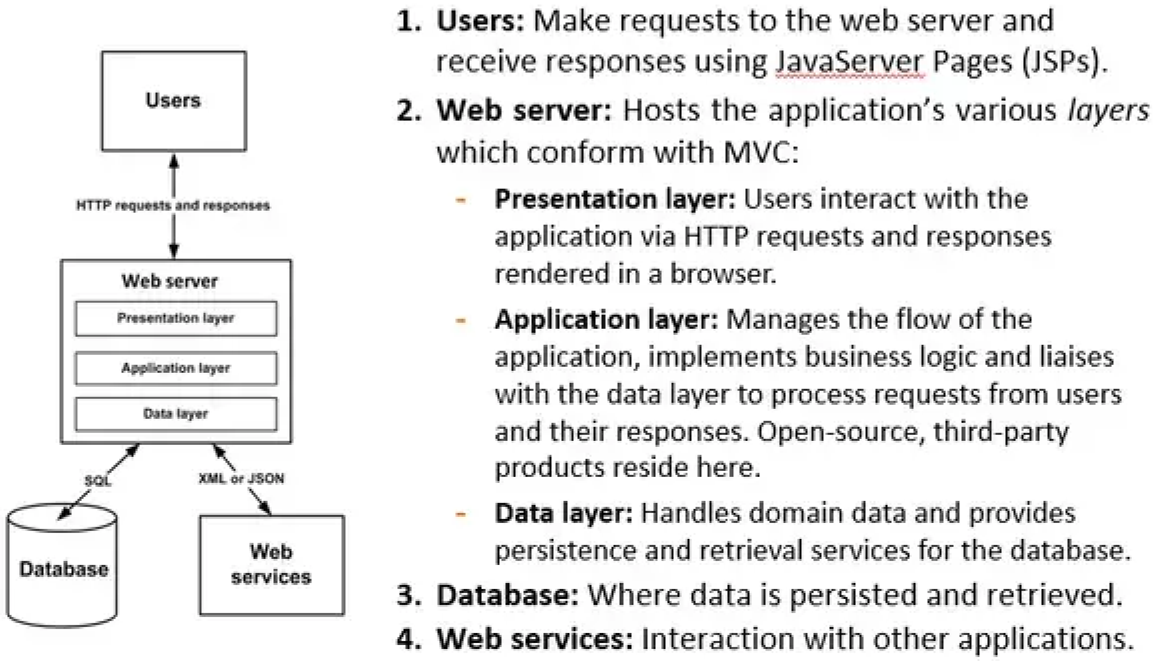
This standalone web-based application can be extended by adding more layers/partitions/tiers – mainly to the server-side – and the system can be developed into a multi-tier web-based application. This concept is specified in the formula below.

The need for introducing this concept is to cater the possibilities of extending the technological and technical capabilities of the web-based applications.

B. Types of Web-based Applications

This section discusses the types of web-based applications towards providing a good understanding of the scope of the proposed term "web-based applications" and its definition.

Taivalsaari and Mikkonen had also come up with a similar idea, and they have introduced a taxonomy for the concept they call “Web-based Software” (Taivalsaari & Mikkonen, 2011). In their taxonomy, they have used the dimension in three categories: “those that are related to the characteristics of the (1) applications, (2)execution environment, and (3) end to end architecture.” The key focal point of their discussions is Cloud Computing, and they have discussed very little beyond the standard browser-based web applications.



1. **Users:** Make requests to the web server and receive responses using JavaServer Pages (JSPs).
2. **Web server:** Hosts the application’s various *layers* which conform with MVC:
   - **Presentation layer:** Users interact with the application via HTTP requests and responses rendered in a browser.
   - **Application layer:** Manages the flow of the application, implements business logic and liaises with the data layer to process requests from users and their responses. Open-source, third-party products reside here.
   - **Data layer:** Handles domain data and provides persistence and retrieval services for the database.
3. **Database:** Where data is persisted and retrieved.
4. **Web services:** Interaction with other applications.

Project Planning

4.1 Purpose of document

A. Toward a Student Information System

This paper basically focuses on providing a simple interface for the easy collation and maintenance of all manner of student information. The creation and management of accurate, up-to- date information regarding students’ academic careers is critical for students and for the faculties and administration of Bharati Vidyapeeth University, Pune and for any other educational institution. A student information system deals with the proper student information, assessment scores and other student related fields which are required by the administration.

All these data need to be made available through an online interface.

B. A Study of Student Information Management Software

This paper focuses on providing information to support the operation, management and decision-making functions of enterprises or organizations. In the face of a huge amount of information, it is required to possess the student information management system to improve the efficiency of student management. Through this system, the standardized management, scientific statistics and fast query of student information can be realized, and thus the workload of management can be reduced. In this paper, a typical student information management system will be established to realize the systematization, standardization and automation of student information relationships.

C. Web Based Student Information System

This paper focuses on a simple interface for maintenance of student information. The creation and management of accurate, up-to- date information regarding a student's academic career is critically important in the university as well as colleges. Student information system deals with all kinds of student details, academic related reports, college details, course details, curriculum, batch details and other details too. It tracks all the details of a student which can be used for all reporting purposes, tracking of progress in the course, completed semesters, years. Different reports and Queries can be generated based on vast options related to students, batch, course, exams, semesters.

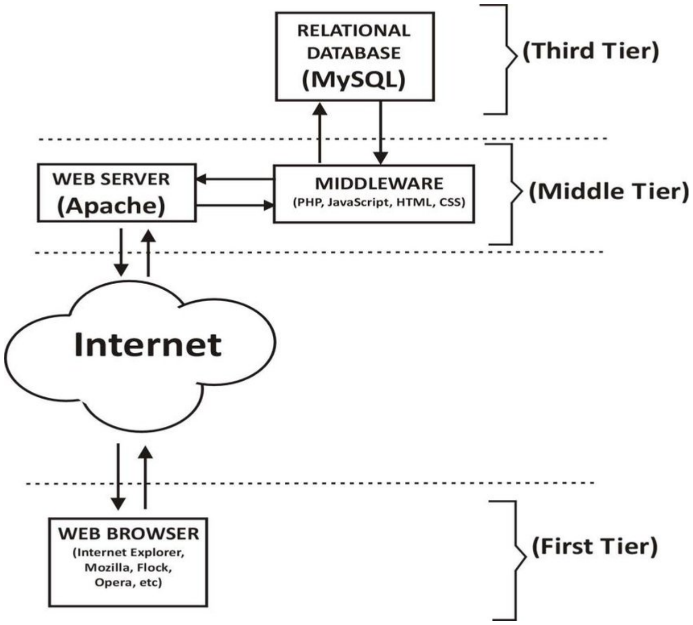
Chapter-5
SYSTEM MODELING

5.1 UML Modeling -

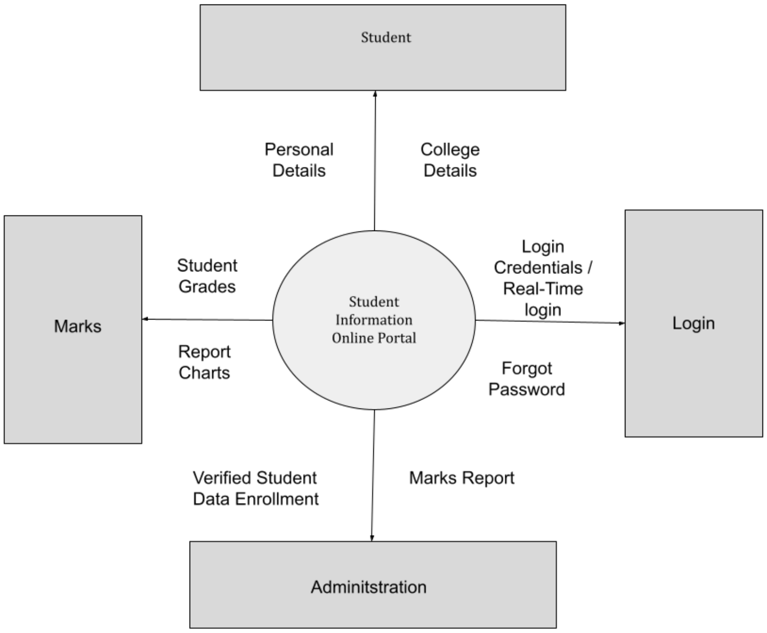
UML class diagrams are mostly used in software engineering for modeling the static structure of applications. They help business analysts to analyze a business domain and depict core business elements, which lay the foundation for a future app. Prepared by business analysts UML class diagrams provide developers with a clear understanding of the whole system structure.
The class diagram is the only UML diagram type that can be mapped directly with object-oriented languages including JavaScript. These diagrams make things much easier for developers in complex projects. That is why this visualization tool is extremely popular in the dev community.

UML class diagrams illustrate the following key elements of a system:

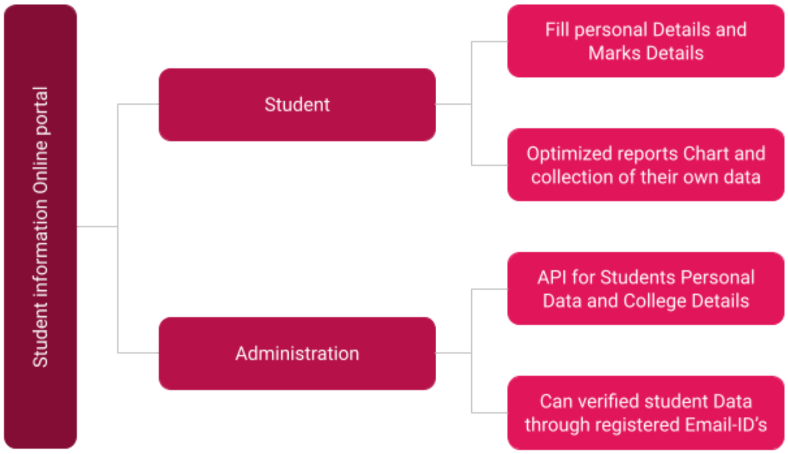
- Classes
- Class attributes
- Relationship


RELATIONAL DATABASE (MySQL)
(Third Tier)
WEB SERVER (Apache)
MIDDLEWARE
(PHP, JavaScript, HTML, CSS)
(Middle Tier)
Internet
WEB BROWSER
(Internet Explorer, Mozilla, Flock, Opera, etc)
(First Tier)

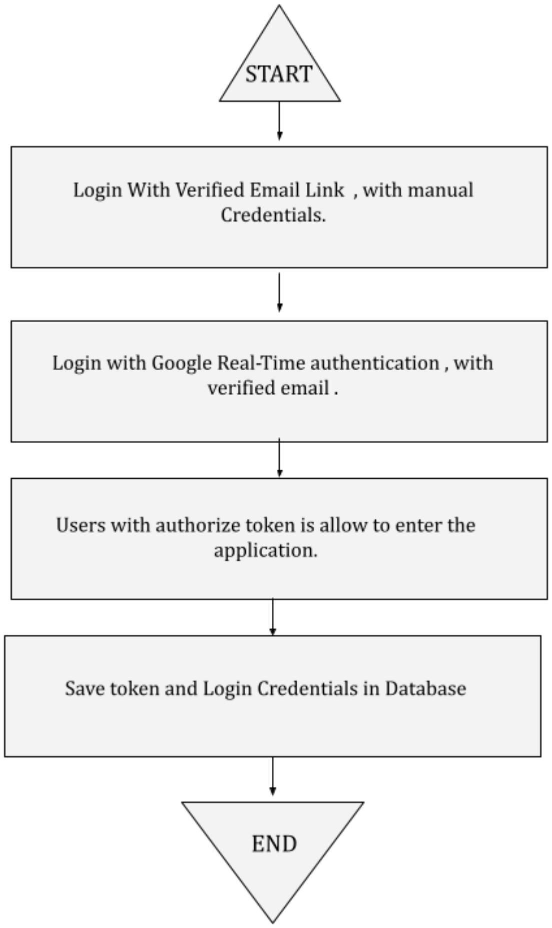
5.1.1 Structural Modeling



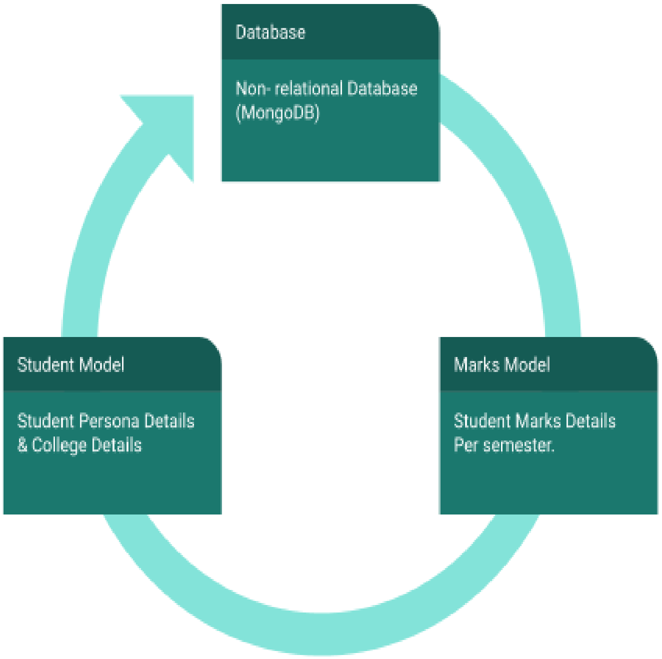
5.1.2 Behavioral Modeling

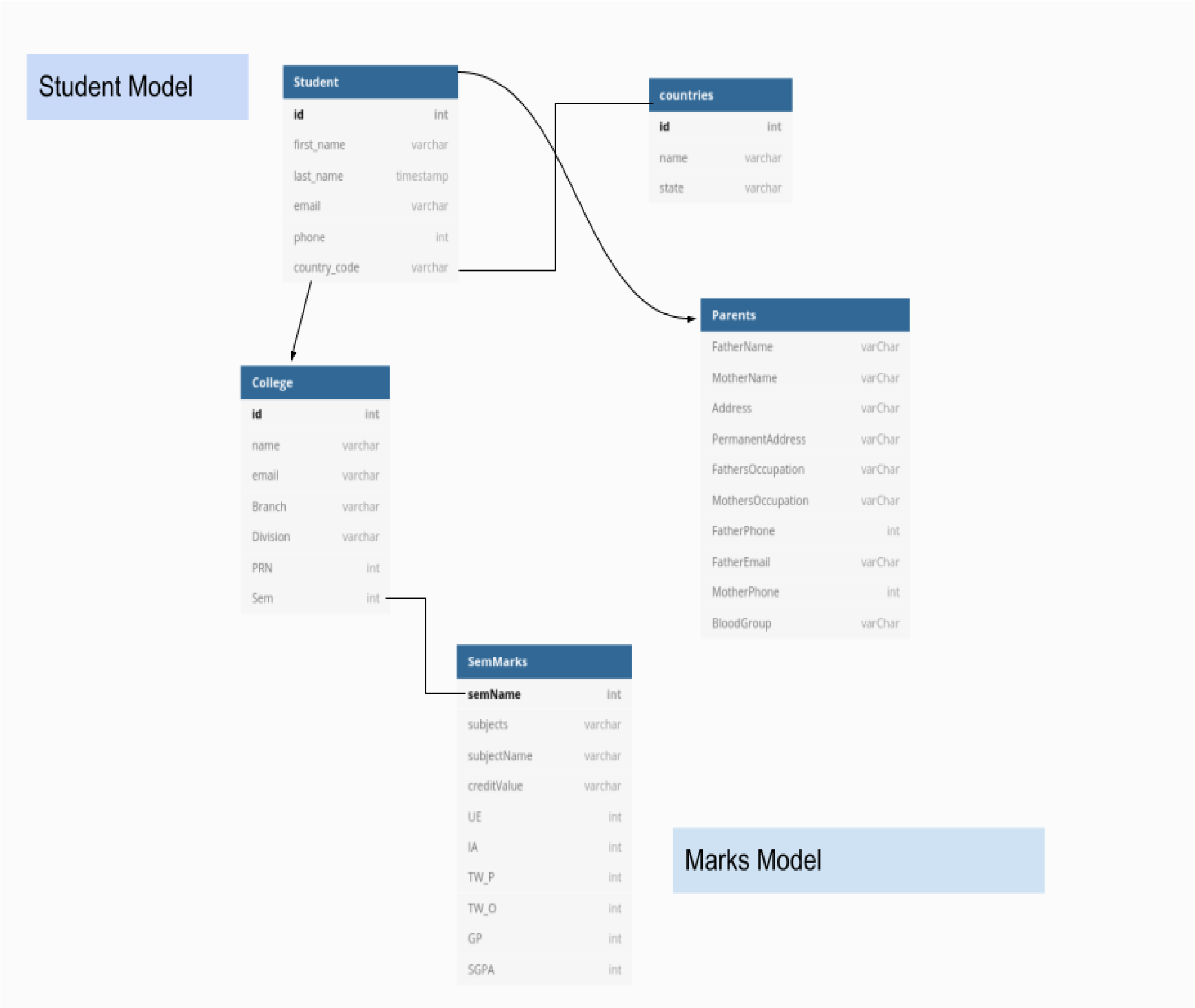

Student information Online portal
Student
Administration
Fill personal Details and Marks Details
Optimized reports Chart and collection of their own data
API for Students Personal Data and College Details
Can verified student Data through registered Email-ID's

5.1.3 State Modeling


START
Login With Verified Email Link , with manual Credentials.
Login with Google Real-Time authentication , with verified email .
Users with authorize token is allow to enter the application.
Save token and Login Credentials in Database
END

5.1.4 Architectural Modeling


Database
Non- relational Database (MongoDB)
Student Model
Student Persona Details & College Details
Marks Model
Student Marks Details Per semester.



Student	
id	int
first_name	varchar
last_name	timestamp
email	varchar
phone	int
country_code	varchar

countries	
id	int
name	varchar
state	varchar

Parents	
FatherName	varChar
MotherName	varChar
Address	varChar
PermanentAddress	varChar
FathersOccupation	varChar
MothersOccupation	varChar
FatherPhone	int
FatherEmail	varChar
MotherPhone	int
BloodGroup	varChar

College	
id	int
name	varchar
email	varchar
Branch	varchar
Division	varchar
PRN	int
Sem	int

SemMarks	
semName	int
subjects	varchar
subjectName	varchar
creditValue	varchar
UE	int
IA	int
TW_P	int
TW_O	int
GP	int
SGPA	int

Chapter-6

SYSTEM IMPLEMENTATION

6.1 Architecture of proposed system -

We have used JAVASCRIPT and React-JS a javascript library as our primary teck stack to build this system and this system we have used Material-UI frameworks and Bootstrap for the UI development.

And for the backend we have used NodeJS framework Express for the API integration and all the required token security system in the backend itself .

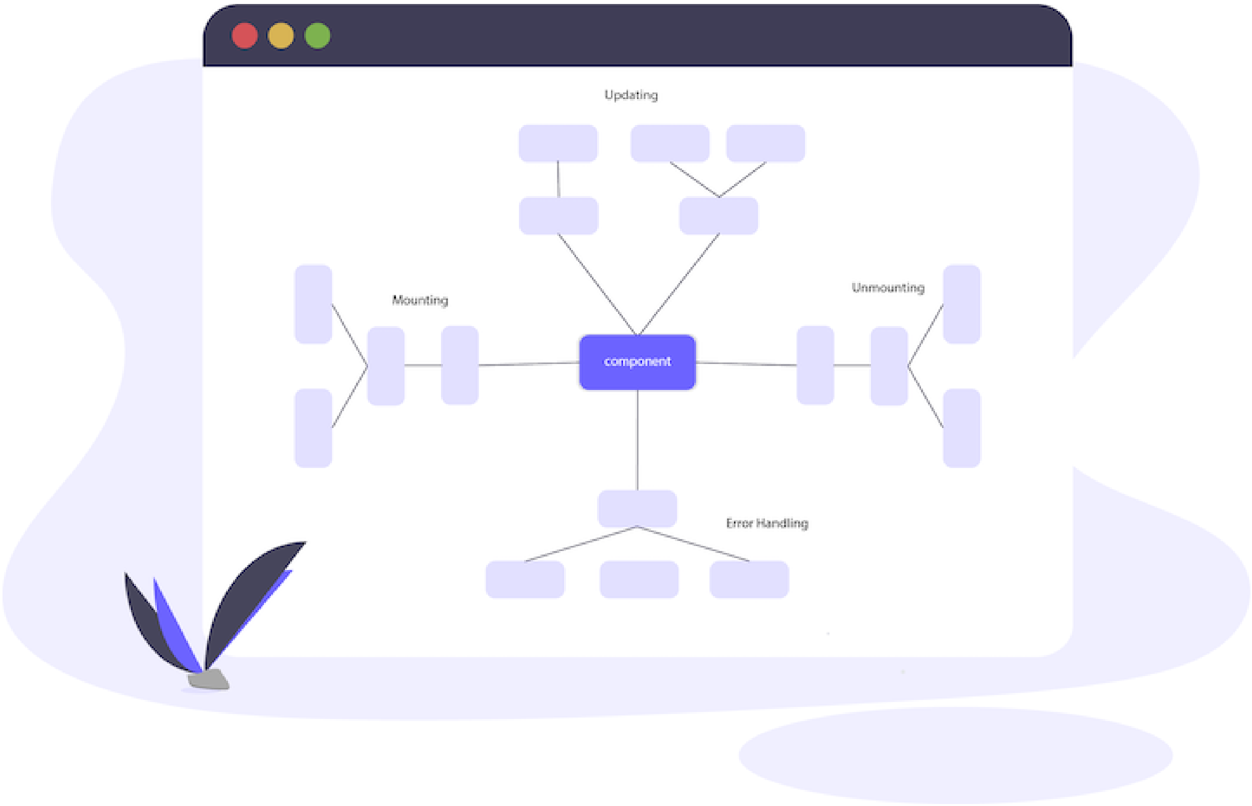
We have used NoSql database MongoDb for storing the user details .

So in term of our project tech stack we are using one lifecycle method for our project.

PROJECT LIFECYCLE:

Description

For a visual overview of what we'll cover in this tutorial, the following diagram illustrates how the React component lifecycle works:



What is the React component lifecycle?

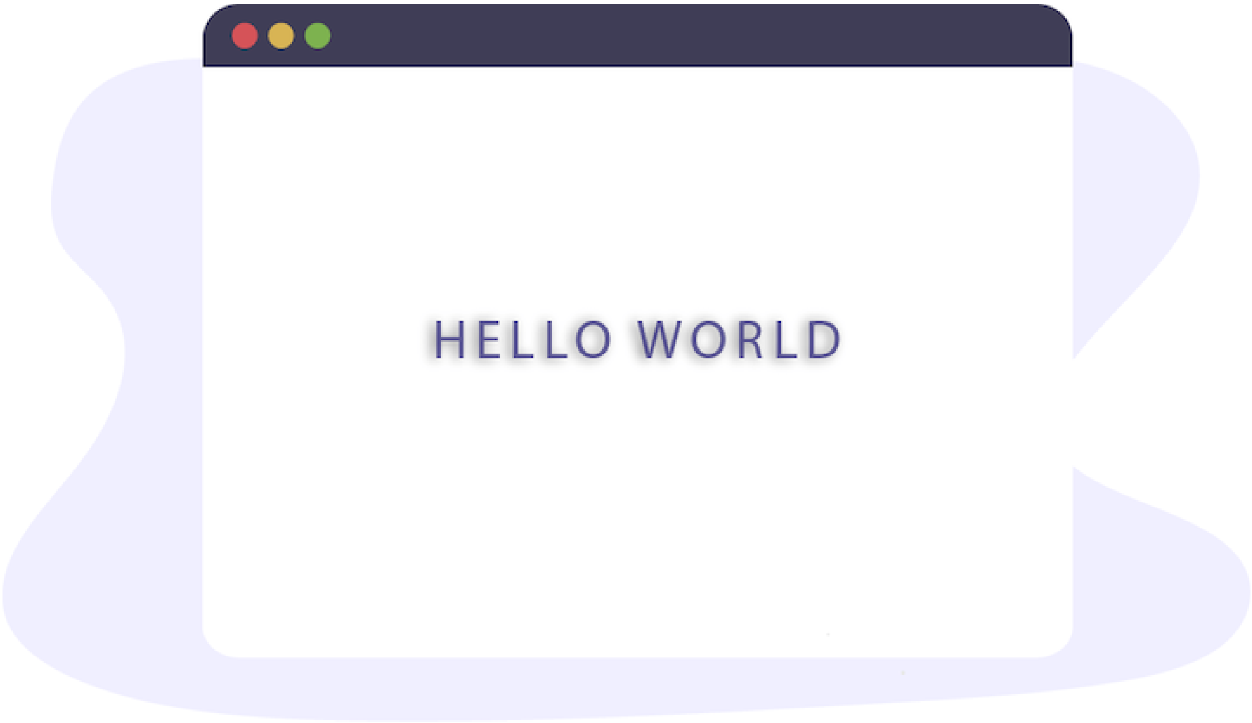
In React, components go through a life cycle of events:

1. Mounting (adding nodes to the DOM)
2. Updating (altering existing nodes in the DOM)
3. Unmounting (removing nodes from the DOM)
4. Error handling (verifying that your code works and is bug-free)

You can think of these events as a component's birth, growth, and death, respectively. Error handling is like an annual physical.

Let's look at a simple example. If I told you to build a Hello World component, you might write something like this:

```
class HelloWorld extends React.Component {
  render() {
   return <h1> Hello World </h1>
  }
}
```



HELLO WORLD

Before rendering, the component will have gone through its mounting, updating, and unmounting phase. Let's break it down further.

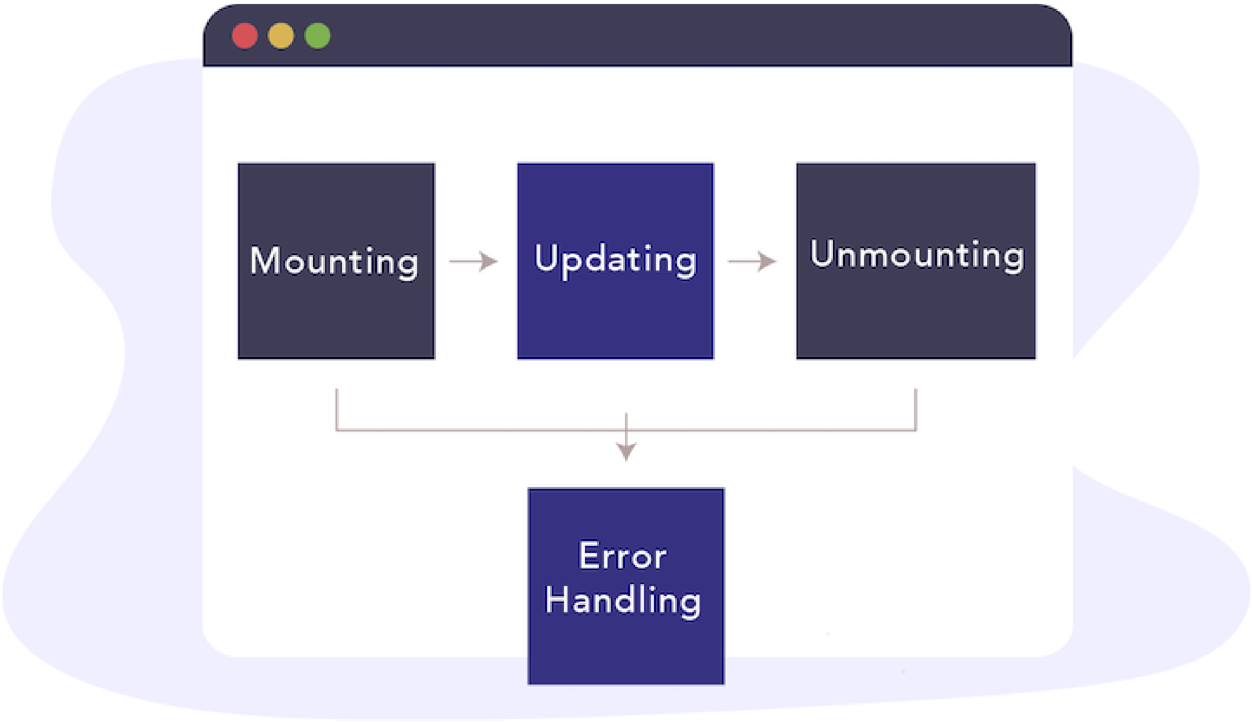
Mounting a component is like bringing a newborn baby into the world. This is the component's first glimpse of life. At this phase, the component, which consists of your code and React's internals, is then inserted into the DOM.

After the mounting phase, the React component "grows" during the updating phase. Without updates, the component would remain as it was when it was initially created in the DOM. As you might imagine, many of the components you write till need to be updated, whether via a change in state or props. Consequently, they go through the updating phase as well.

The final phase is called the unmounting phase. At this stage, the component "dies". In React lingo, it is removed from the DOM.

There's one more phase a React component can go through: the error handling phase. This occurs when your code doesn't run or there's a bug somewhere. Think of it like an annual physical.

The following diagram shows the React lifecycle:

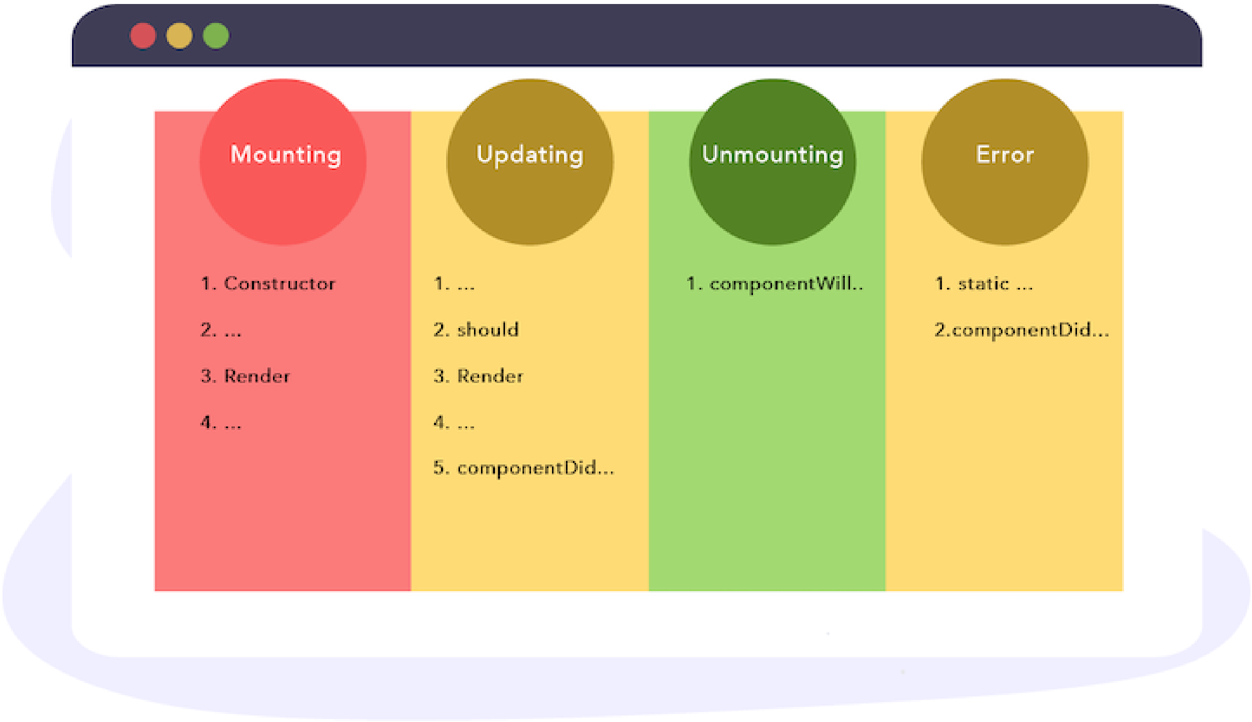


Note that a React component may not go through every phase. For example, a component could be mounted one minute and then unmounted the next without any updates or error handling. The illustration and example described above assume that the component goes through all four phases.

React lifecycle methods

Each React lifecycle phase has a number of lifecycle methods that you can override to run code at specific times during the process. These are popularly known as component lifecycle methods.

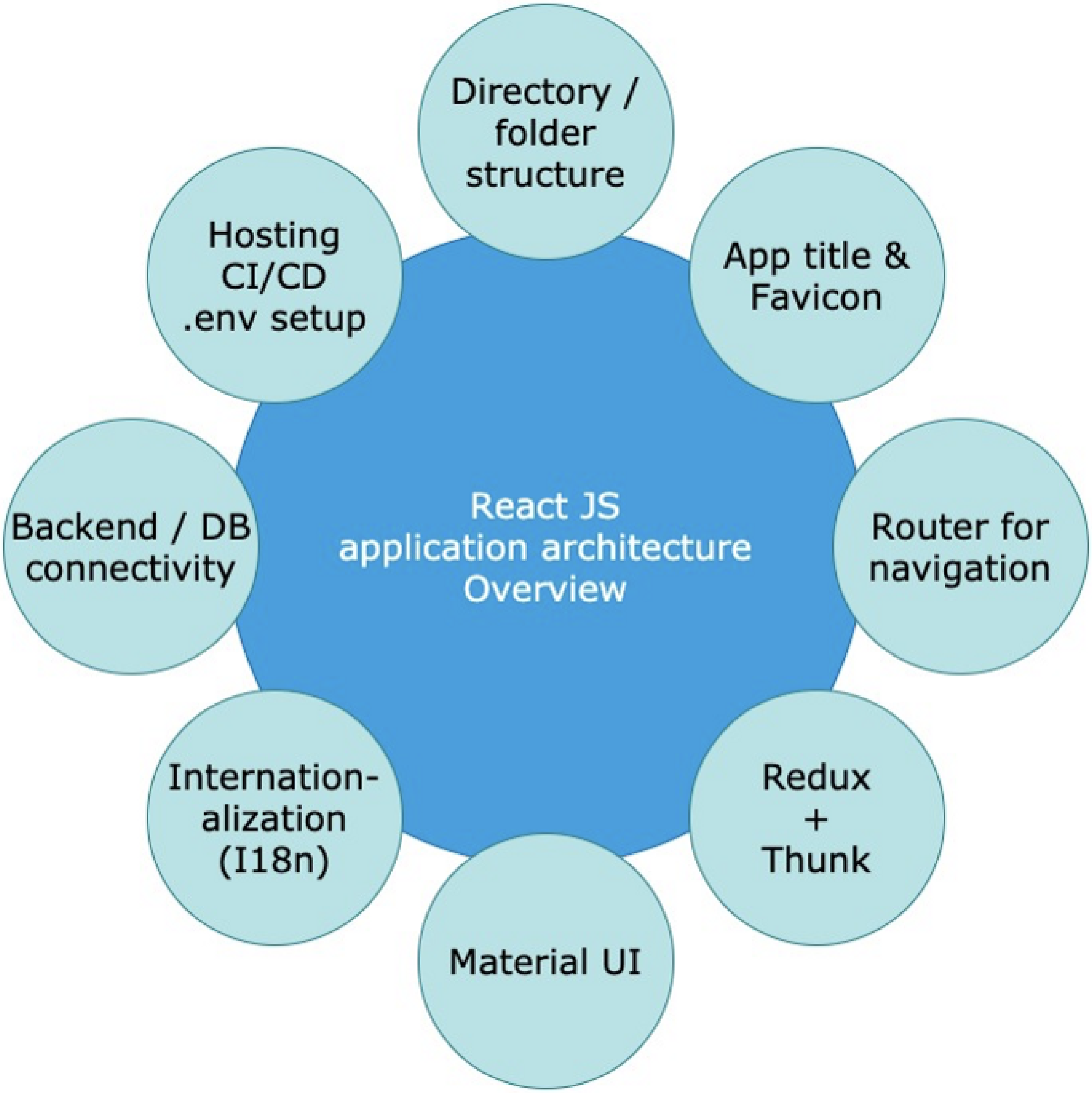
The diagram below shows the React lifecycle methods associated with the mounting, updating, umounting, and error lifecycle phases:



Mounting	Updating	Unmounting	Error
1. Constructor	1. ...	1. componentWill..	1. static ...
2. ...	2. should		2.componentDid...
3. Render	3. Render		
4. ...	4. ...		
	5. componentDid...		

Mounting lifecycle methods
The mounting phase refers to the phase during which a component is created and inserted to the DOM.

Structure of the file system

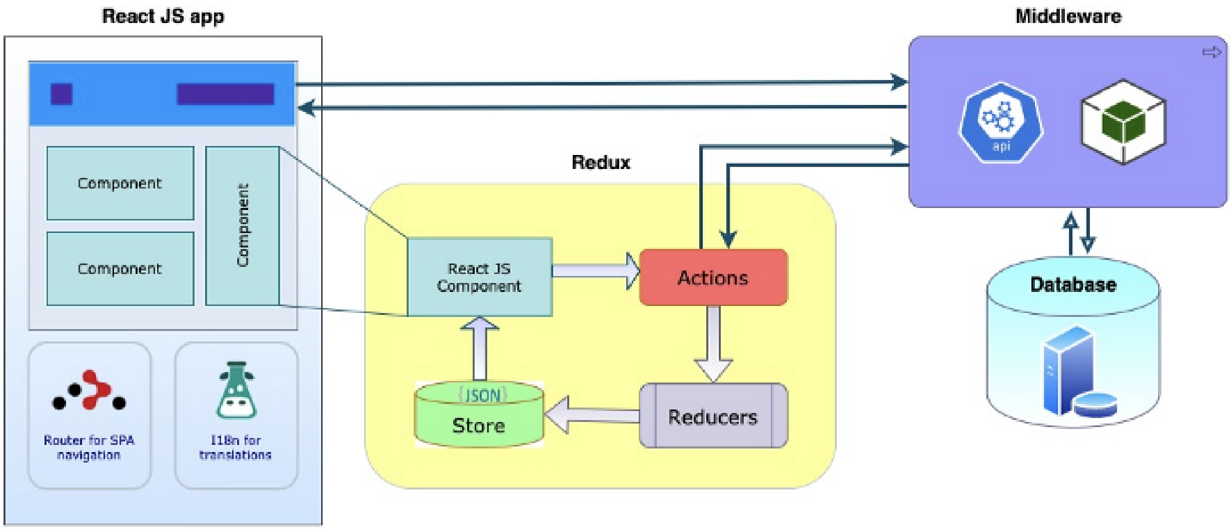

Directory / folder structure
App title & Favicon
Router for navigation
Redux + Thunk
Material UI
Internation-alization (I18n)
Backend / DB connectivity
Hosting CI/CD .env setup
React JS application architecture Overview

Use the following command to add new react project without installing react CLI globally:

```
$ npx create-react-app myapp
$ cd myapp
$ npm start
```

This will create a react js web app in a directory called “myapp” and run the project in the development environment on your default browser. It will also watch for the changes and perform hot-reloading for the best development experience.

React js application that is created via ‘create-react-app’ ships with some default code. Let’s remove almost everything (logo, images, content and styling) and keep the bare skeleton application.



You can run this on local host by doing **npm run start** make sure Node is installed in your system by going into the client side and to run the backend first we have to do :-

Npm install

Npm start-

Your server will run at localhost:5000

And your React server will run on **localhost:3000**

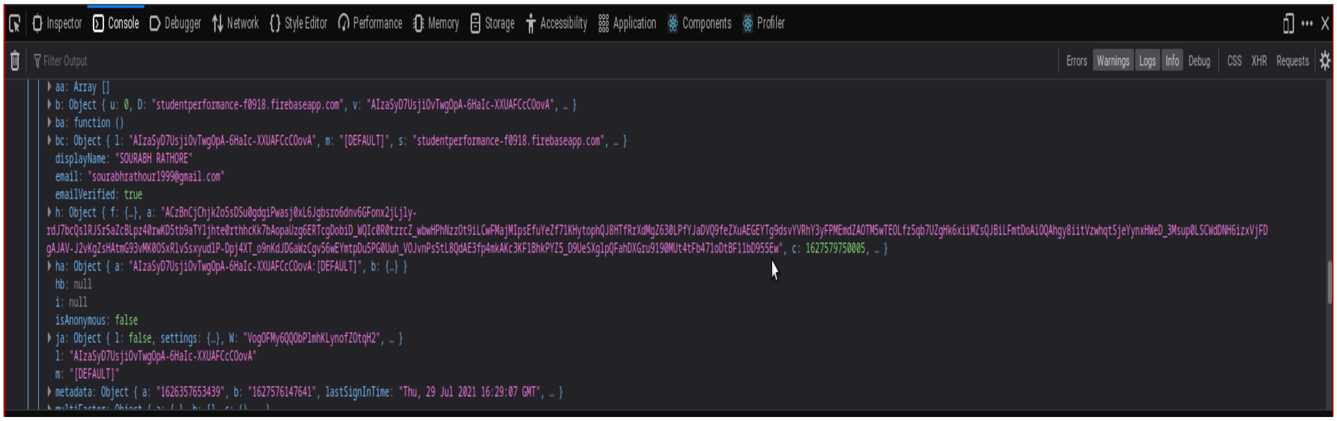
6.2 Algorithm for system

Our algo of system works with the term of security authentication if the user is verified and comes up with the token then only we allow him to enter otherwise we will say it is unauthorized Our Algo on server side works like this -

```
//verify user firebase token AUTH

app.use(function (req, res, next) {
 if (req.token) {
   admin
     .auth()
     .verifyIdToken(req.token)
     .then(function (user) {
       req.user = user;
       console.log(user);
       next();
     })
     .catch(function (error) {
       res.sendStatus(401);
     });
 } else {
   res.sendStatus(401);
 }
});
```

If we get in console like this then we will move forward the user -



```
▸ aa: Array []
▸ b: Object { u: 0, D: "studentperformance-f0918.firebaseapp.com", v: "AIzaSyD7UsjiOvTwgOpA-6HaIc-XXUAFCcCOovA", … }
▸ ba: function ()
▸ bc: Object { l: "AIzaSyD7UsjiOvTwgOpA-6HaIc-XXUAFCcCOovA", m: "[DEFAULT]", s: "studentperformance-f0918.firebaseapp.com", … }
  displayName: "SOURABH RATHORE"
  email: "sourabhrathour1999@gmail.com"
  emailVerified: true
▸ h: Object { f: {…}, a: "ACzBnCjChjkZo5sDSu0gdgiPwasj0xL6Jgbsro6dnv6GFonx2jLjly-
rdJ7bcQslRJSr5aZcBLpz40rwKD5tb9aTYljhte0rthhcKk7bAopaUzg6ERTcgDobiD_WQIc0R0tzrcZ_wbwHPhNzzOt9iLCwFMajMIpsEfuYeZf71KHytophQJ8HTfRrXdMgZ630LPfYJaDVQ9feZXuAEGEYTg9dsvYVRhY3yFPMEmdZAOTM5wTEOLfz5qb7UZgHk6xiiMZsQJBiLFmtDoAiOQAhgy8iitVzwhqtSjeYynxHWeD_3Msup0LSCWdDNH6izxVjFD
gAJAV-J2vKgZsHAtmG93vMK0OSxRlvSsxyudlP-Dpj4XT_o9nKdJDGaWzCgv56wEYmtpDu5PG0Uuh_VOJvnPs5tL8QdAE3fp4mkAKc3KF1BhkPYZ5_D9UeSXglpQFahDXGru9190MUt4tFb471oDtBF11bD955Ew", c: 1627579750005, … }
▸ ha: Object { a: "AIzaSyD7UsjiOvTwgOpA-6HaIc-XXUAFCcCOovA:[DEFAULT]", b: {…} }
  hb: null
  i: null
  isAnonymous: false
▸ ja: Object { l: false, settings: {…}, W: "VogOFMy6QQObPlmhKLynofZOtqH2", … }
  l: "AIzaSyD7UsjiOvTwgOpA-6HaIc-XXUAFCcCOovA"
  m: "[DEFAULT]"
▸ metadata: Object { a: "1626357653439", b: "1627576147641", lastSignInTime: "Thu, 29 Jul 2021 16:29:07 GMT", … }
▸ multiFactor: Object { a: {…}, b: [], c: {}, … }
```

Here we will be having the Token and user Details and as well as the Email-verified by the Google is true -

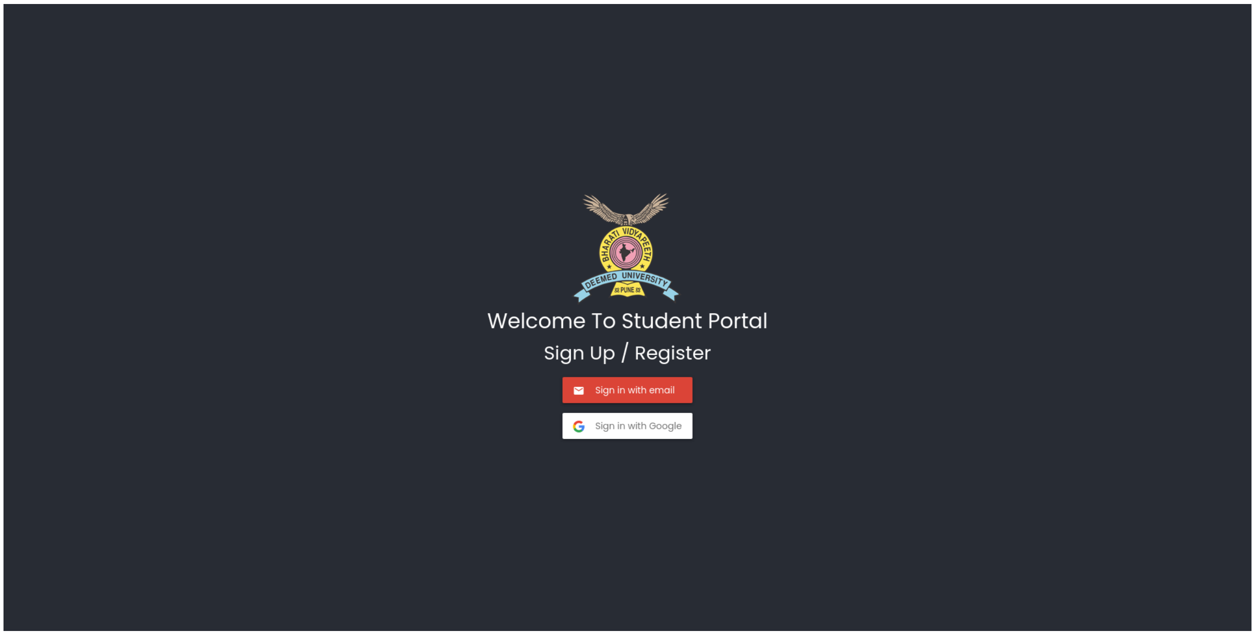
```
// Verifying and storing the user

let UserContext = React.createContext();
function App() {
 let [user, setUser] = useState();

 useEffect(function () {
   const authObserver = firebase.auth().onAuthStateChanged(function (user)
{
      setUser(user);
      if (user) {
        user
          .getIdToken(true)
          .then(function (idToken) {
            axios.defaults.headers["Authorization"] = `Bearer ${idToken}`;
          })
          .catch(function (error) {
            // handle error
          });
        return authObserver;
      }
   });
 });
 console.log("user", user);
```

User interface of our system: -
Since it is a web based application so it must be simple and hence worth We have to say is good in looking.

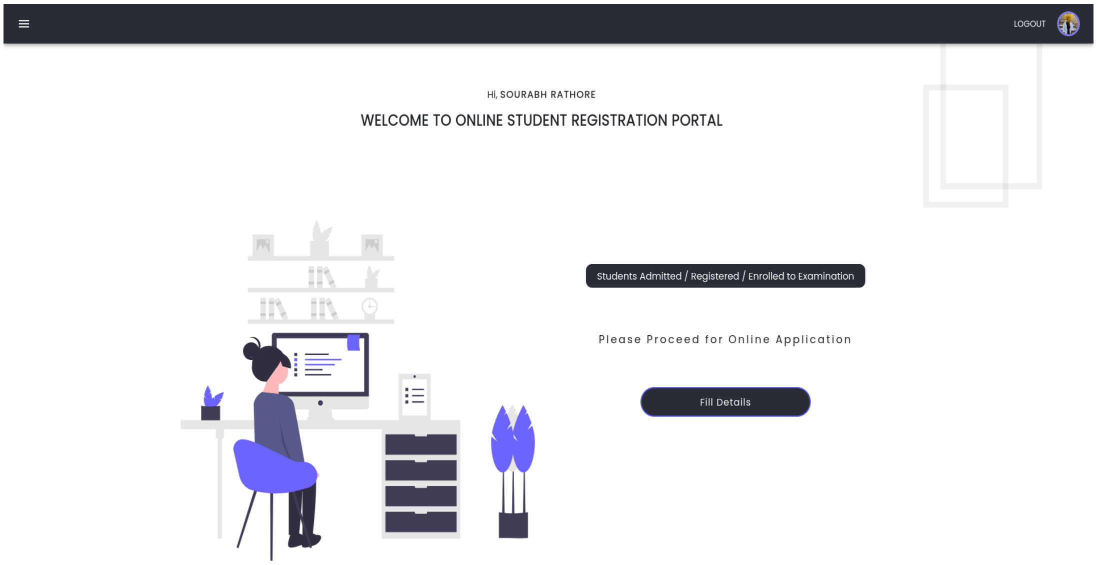
Login Page -



Welcome To Student Portal

Sign Up / Register

Home Page -



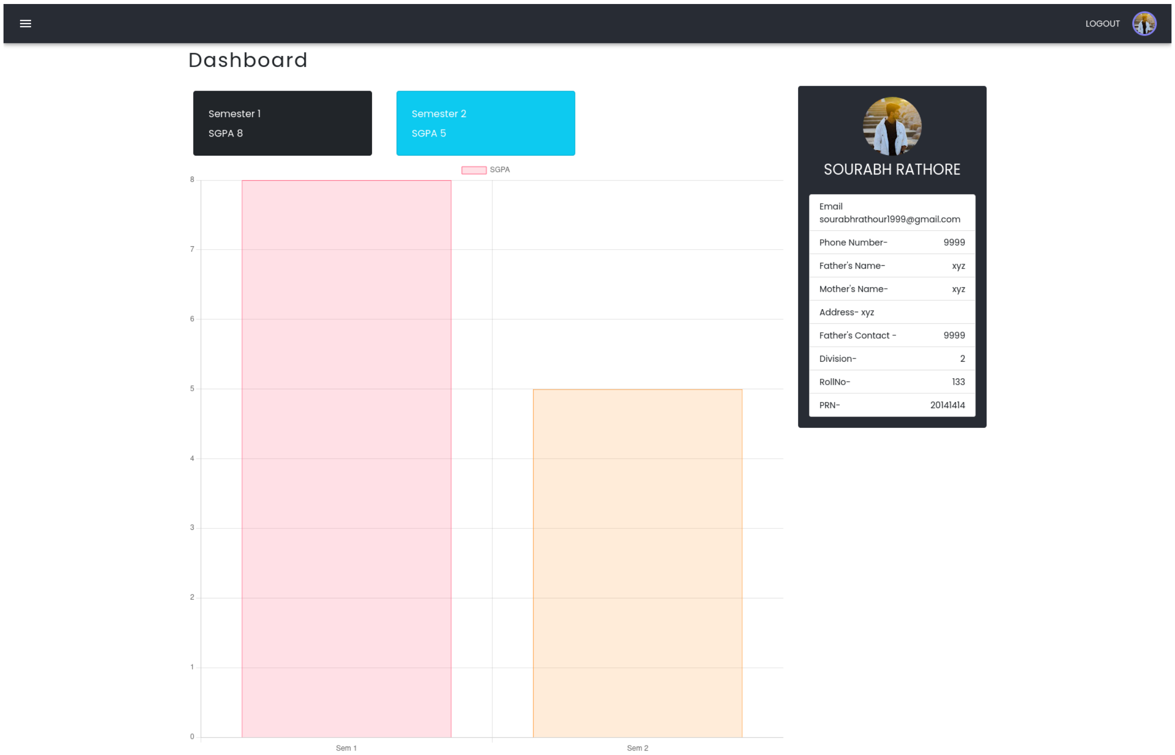
Hi, SOURABH RATHORE

WELCOME TO ONLINE STUDENT REGISTRATION PORTAL

Students Admitted / Registered / Enrolled to Examination

Please Proceed for Online Application

Dashboard of student-



Dashboard

Semester 1
SGPA 8

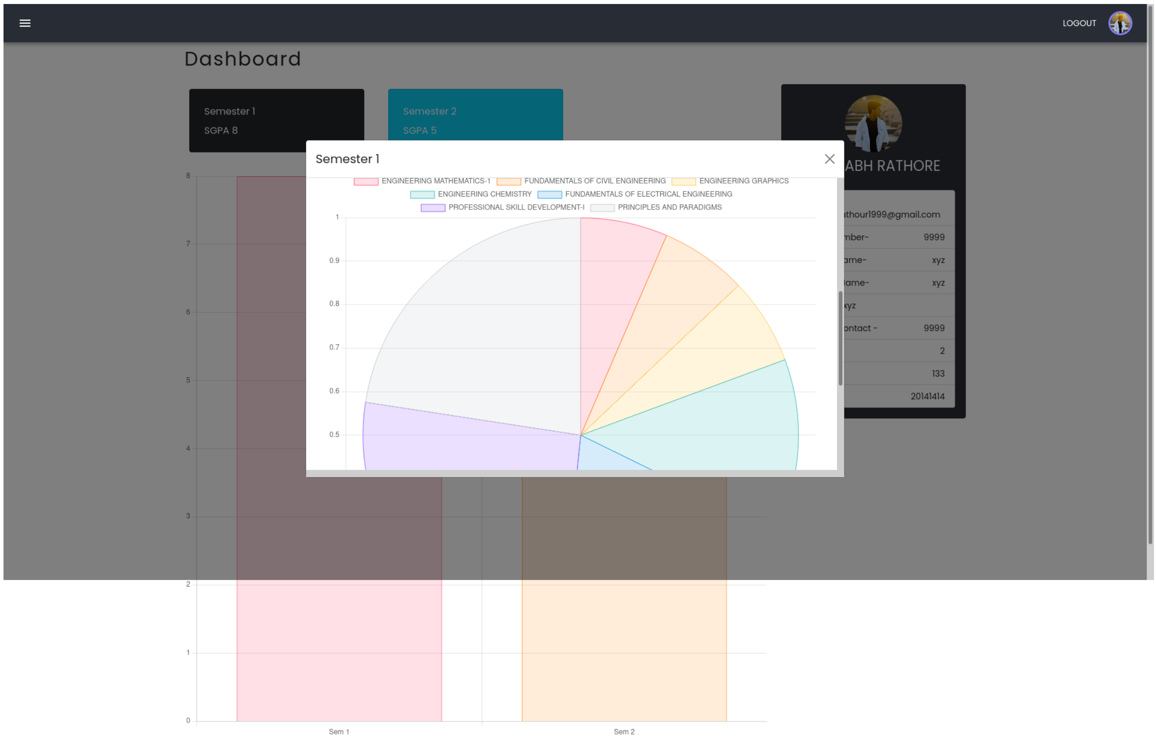
Semester 2
SGPA 5

SOURABH RATHORE

Email
sourabhrathour1999@gmail.com

Phone Number-	9999
Father's Name-	xyz
Mother's Name-	xyz
Address- xyz	
Father's Contact -	9999
Division-	2
RollNo-	133
PRN-	20141414

Dashboard of semester Marks -


LOGOUT
ENGINEERING MATHEMATICS-1
FUNDAMENTALS OF CIVIL ENGINEERING
ENGINEERING GRAPHICS
ENGINEERING CHEMISTRY
FUNDAMENTALS OF ELECTRICAL ENGINEERING
PROFESSIONAL SKILL DEVELOPMENT-I
PRINCIPLES AND PARADIGMS
1
0.9
0.8
0.7
0.6
0.5
8
7
6
5
4
3
2
1
0
Sem 1
Sem 2

Dashboard

Semester 1
SGPA 8

Semester 2
SGPA 5

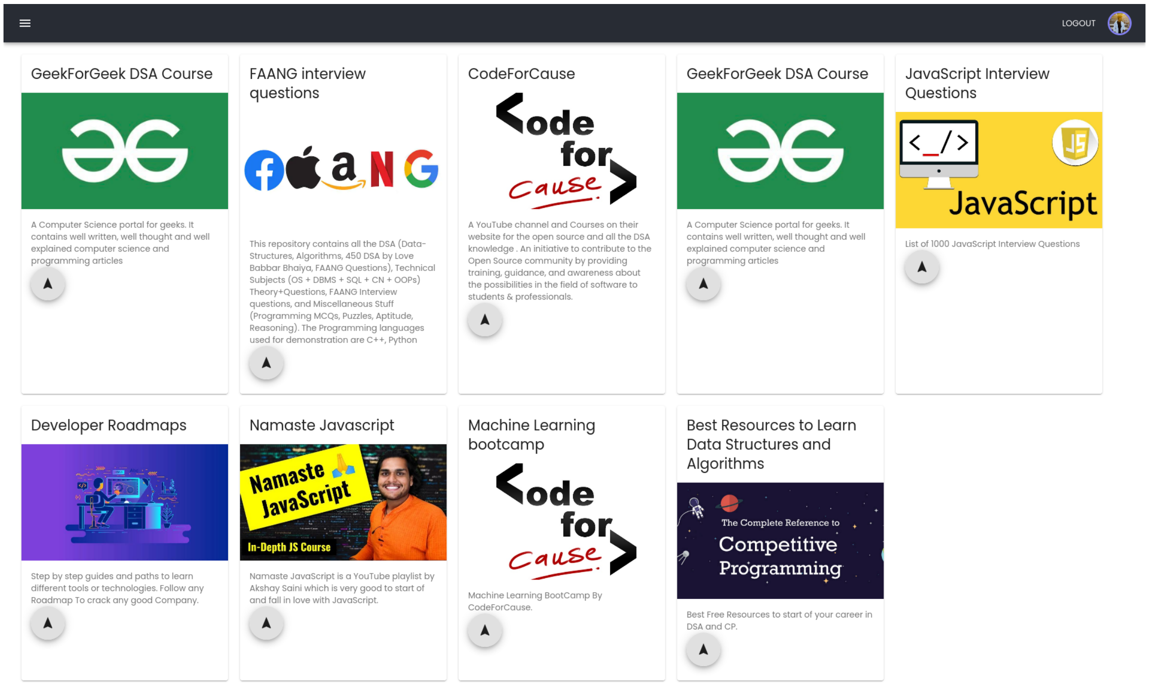
Semester 1

ABH RATHORE

athour1999@gmail.com

mber-	9999
ame-	xyz
ame-	xyz
xyz	
ontact -	9999
	2
	133
	20141414

Course Recommendation -



GeekForGeek DSA Course

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles

FAANG interview questions

This repository contains all the DSA (Data-Structures, Algorithms, 450 DSA by Love Babbar Bhaiya, FAANG Questions), Technical Subjects (OS + DBMS + SQL + CN + OOPs) Theory+Questions, FAANG Interview questions, and Miscellaneous Stuff (Programming MCQs, Puzzles, Aptitude, Reasoning). The Programming languages used for demonstration are C++, Python

CodeForCause

A YouTube channel and Courses on their website for the open source and all the DSA knowledge . An initiative to contribute to the Open Source community by providing training, guidance, and awareness about the possibilities in the field of software to students & professionals.

GeekForGeek DSA Course

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles

JavaScript Interview Questions

List of 1000 JavaScript Interview Questions

Developer Roadmaps

Step by step guides and paths to learn different tools or technologies. Follow any Roadmap To crack any good Company.

Namaste Javascript

Namaste JavaScript is a YouTube playlist by Akshay Saini which is very good to start of and fall in love with JavaScript.

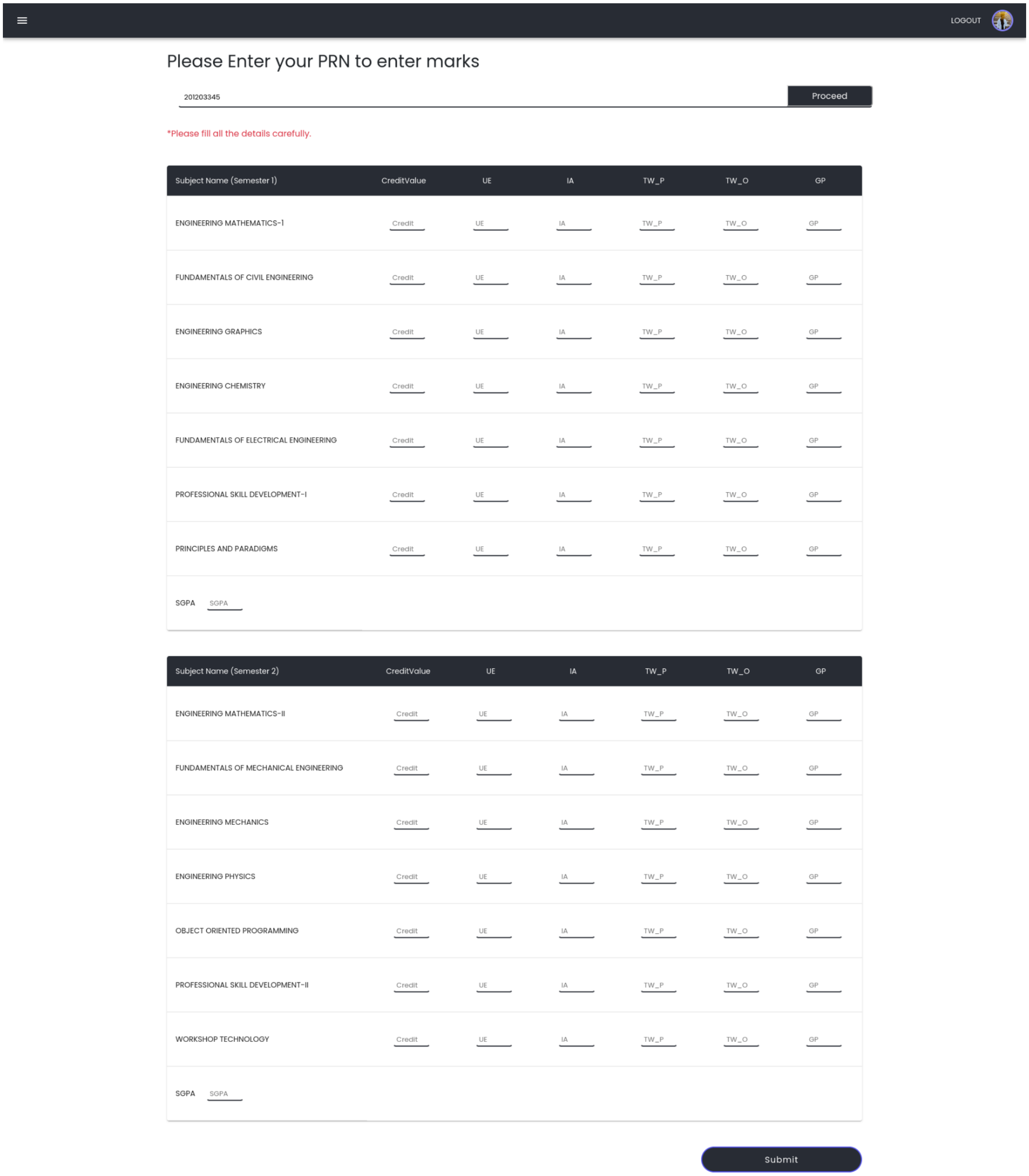
Machine Learning bootcamp

Machine Learning BootCamp By CodeForCause.

Best Resources to Learn Data Structures and Algorithms

Best Free Resources to start of your career in DSA and CP.

Semester Marks Details in Marksheet Format -



Please Enter your PRN to enter marks

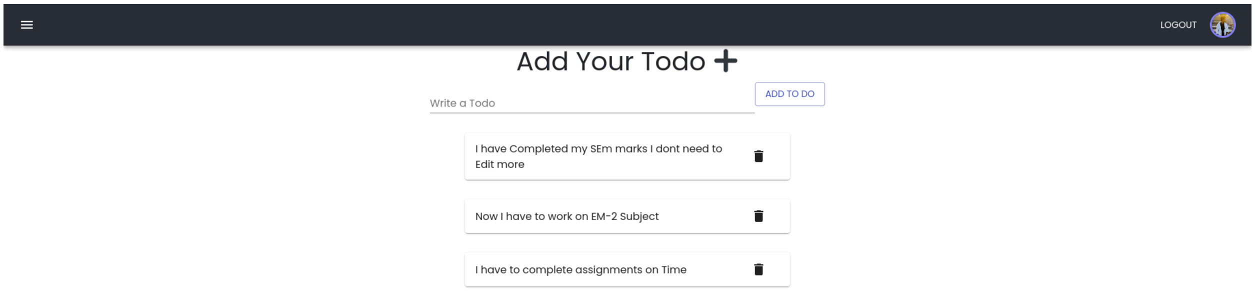
201203345

*Please fill all the details carefully.

Subject Name (Semester 1)	CreditValue	UE	IA	TW_P	TW_O	GP
ENGINEERING MATHEMATICS-1	Credit	UE	IA	TW_P	TW_O	GP
FUNDAMENTALS OF CIVIL ENGINEERING	Credit	UE	IA	TW_P	TW_O	GP
ENGINEERING GRAPHICS	Credit	UE	IA	TW_P	TW_O	GP
ENGINEERING CHEMISTRY	Credit	UE	IA	TW_P	TW_O	GP
FUNDAMENTALS OF ELECTRICAL ENGINEERING	Credit	UE	IA	TW_P	TW_O	GP
PROFESSIONAL SKILL DEVELOPMENT-I	Credit	UE	IA	TW_P	TW_O	GP
PRINCIPLES AND PARADIGMS	Credit	UE	IA	TW_P	TW_O	GP
SGPA SGPA						

Subject Name (Semester 2)	CreditValue	UE	IA	TW_P	TW_O	GP
ENGINEERING MATHEMATICS-II	Credit	UE	IA	TW_P	TW_O	GP
FUNDAMENTALS OF MECHANICAL ENGINEERING	Credit	UE	IA	TW_P	TW_O	GP
ENGINEERING MECHANICS	Credit	UE	IA	TW_P	TW_O	GP
ENGINEERING PHYSICS	Credit	UE	IA	TW_P	TW_O	GP
OBJECT ORIENTED PROGRAMMING	Credit	UE	IA	TW_P	TW_O	GP
PROFESSIONAL SKILL DEVELOPMENT-II	Credit	UE	IA	TW_P	TW_O	GP
WORKSHOP TECHNOLOGY	Credit	UE	IA	TW_P	TW_O	GP
SGPA SGPA						

Add Notes for the user in order to stay up to the college stuff -



I have Completed my SEm marks I dont need to Edit more

Now I have to work on EM-2 Subject

I have to complete assignments on Time

Chapter-7
Testing Approach

7.1 Introduction -

React Testing Library-
React Testing Library builds on top of DOM Testing Library by adding APIs for working with React components.

> Projects created with Create React App have out of the box support for **React Testing Library**. If that is not the case, you can add it via npm like so:

```
npm install --save-dev @testing-library/react
```

- React Testing Library on GitHub

The problem

You want to write maintainable tests for your React components. As a part of this goal, you want your tests to avoid including implementation details of your components and rather focus on making your tests give you the confidence for which they are intended. As part of this, you want your testbase to be maintainable in the long run so refactors of your components (changes to implementation but not functionality) don't break your tests and slow you and your team down.

The React Testing Library is a very light-weight solution for testing React components. It provides light utility functions on top of react-dom and react-dom/test-utils, in a way that encourages better testing practices. Its primary guiding principle is:

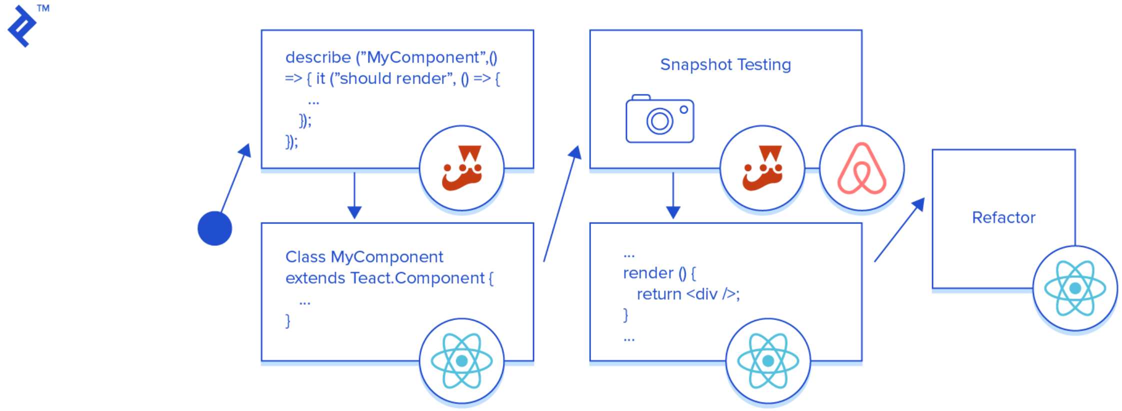
The more your tests resemble the way your software is used, the more confidence they can give you.

So rather than dealing with instances of rendered React components, your tests will work with actual DOM nodes. The utilities this library provides facilitate querying the DOM in the same way the user would. Finding form elements by their label text (just like a user would), finding links and buttons from their text (like a user would). It also exposes a recommended way to find elements by a data-testid as an "escape hatch" for elements where the text content and label do not make sense or are not practical.

This library encourages your applications to be more accessible and allows you to get your tests closer to using your components the way a user will, which allows your tests to give you more confidence that your application will work when a real user uses it.

This library is a replacement for Enzyme. While you *can* follow these guidelines using Enzyme itself, enforcing this is harder because of all the extra utilities that Enzyme provides (utilities which facilitate testing implementation details).

7.3 Test Strategy Test Strategy



```
describe ("MyComponent",()
=> { it ("should render", () => {
    ...
   });
});
```

```
Class MyComponent
extends Teact.Component {
   ...
}
```

```
...
render () {
   return <div />;
}
...
```

7.3.1 Unit testing

Node.js is a widely used javascript library based on Chrome's V8 JavaScript engine for developing server-side applications in web development.

Unit Testing is a software testing method where individual units/components are tested in isolation. A unit can be described as the smallest testable part of code in an application. Unit testing is generally carried out by developers during the development phase of an application.

In Node.js there are many frameworks available for running unit tests. Some of them are:

- Mocha
- Jest
- Jasmine
- AVA

Unit testing for a node application using these frameworks:

Mocha: Mocha is an old and widely used testing framework for node applications. It supports asynchronous operations like callbacks, promises, and async/await. It is a highly extensible and customizable framework that supports different assertions and mocking libraries.

To install it, open command prompt and type the following command:

```
# Installs globally

npm install mocha -g


# installs in the current directory

npm install mocha --save-dev
```

How to use Mocha?

In order to use this framework in your application:

1. Open the root folder of your project and create a new folder called **test** in it.
2. Inside the test folder, create a new file called test.js which will contain all the code related to testing.

open package.json and add the following line in the scripts block.

```
"scripts": {

"test": "mocha --recursive --exit"
```

3. }

7.3.2 Integrated testing

Requirements for React App Integration Tests

Here are some of the things React developers *want* to do when writing integration tests:

- Test application use-cases from the user's perspective. Users access information on a web page and interact with available controls.

- Mock API calls do not depend on API availability and state for passing/failing tests.
- Mock browser APIs (for example, local storage) since they simply do not exist in the test environment.
- Assert on React DOM state (browser DOM or a native mobile environment).

Now, for some things we should try to *avoid* when writing React app integration tests:

- Test implementation details. Implementation changes should only break a test if they indeed introduced a bug.
- Mock too much. We want to test how all the parts of the app are working together.
- Shallow render. We want to test the composition of all the components in the app down to the smallest component.

Sample App Setup

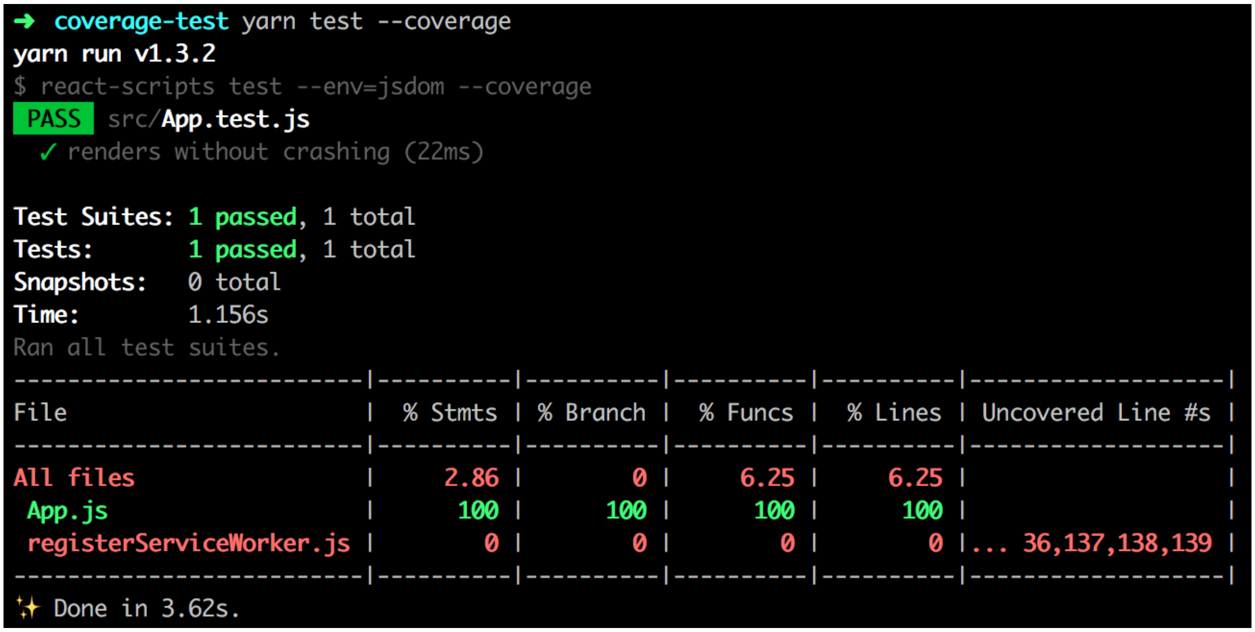
The app for which we are going to write sample integration tests implements a simple scenario:

- The user enters a GitHub username.
- The app displays a list of public repositories associated with the entered username.

How the above functionality is implemented should be irrelevant from an integration testing perspective. However, to keep close to real-world applications, the app follows common React patterns, hence the app:

- Is a single-page app (SPA).
- Makes API requests.
- Has global state management.
- Supports internationalization.
- Utilizes a React component library.

7.3.4 Test case and test results

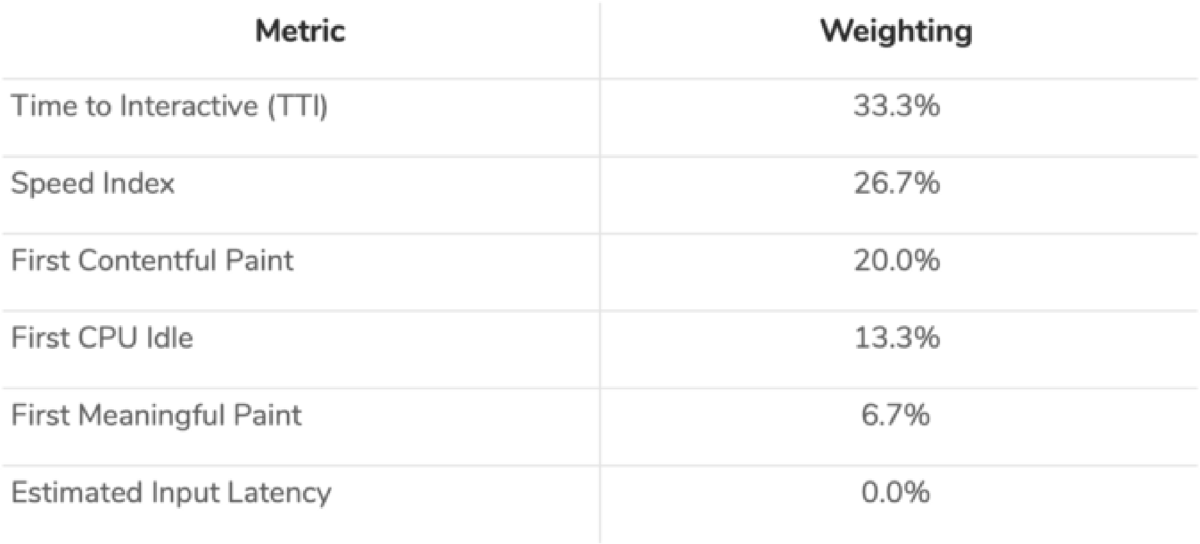


```
➜ coverage-test yarn test --coverage
yarn run v1.3.2
$ react-scripts test --env=jsdom --coverage
 PASS  src/App.test.js
  ✓ renders without crashing (22ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.156s
Ran all test suites.
--------------------------|----------|----------|----------|----------|-------------------|
File                      |  % Stmts | % Branch |  % Funcs |  % Lines | Uncovered Line #s |
--------------------------|----------|----------|----------|----------|-------------------|
All files                 |     2.86 |        0 |     6.25 |     6.25 |                   |
 App.js                   |      100 |      100 |      100 |      100 |                   |
 registerServiceWorker.js |        0 |        0 |        0 |        0 |... 36,137,138,139 |
--------------------------|----------|----------|----------|----------|-------------------|
✨ Done in 3.62s.
```

What are the key Google PageSpeed metrics:

- **First Contentful Paint** — The point at which the very first piece of content is delivered to your page via the browser
- **First Meaningful Paint** — The point at which the user can see a substantial, meaningful piece of content in the browser window
- **First CPU idle** — The point at which your device has loaded enough of the webpage content to be able to handle user inputs such as scrolling, clicking buttons or typing inputs into fields
- **Time to interactive** — The point at which the site is fully loaded and can be interacted with by the user
- **Estimated Input Latency** — A measure of how well the javascript code translates to a responsive webpage (i.e. not laggy in the browser)



Metric	Weighting
Time to Interactive (TTI)	33.3%
Speed Index	26.7%
First Contentful Paint	20.0%
First CPU Idle	13.3%
First Meaningful Paint	6.7%
Estimated Input Latency	0.0%

CONCLUSION :

It is always prudent to opt for a student information system that is designed using modern system architecture to cope with changing requirements. This system should encompass very solid information coding and distinctly outlined business applications. The overview of the system elaborates the ease of information delivery at the tip of your fingers with precise data and increases the retention rate of students and teaches them how to manage their time efficiently.In the future we can add more features like one-to one communication between faculty and students and we can add custom attendance system with face recognition. In all the school management system is bringing a great difference in the lives of students, teachers, parents, and the admin. Good management offers better productivity and hence more progress towards development. Seeing its demands and benefits, we have come forward with best-featured school management Online portal. It helps the college to achieve the target, reduce work, increase efficiency, eliminating error, and monitoring progress.

REFERENCES:

[1] S.R.Bharamagoudar, Geeta R.B, S.G.Totad, "*Web service api for student information and course management systems*"International Journal of Advanced Research in Computer and Communication Engineering Vol. June 2013 [4] Hanan A. Al-Souly, Abeer S. Al-Sheddi, Heba A. Kurdi "*Enhanced TSFS Algorithm for Secure Database Encryption*" Science and Information Conference 2013. -p328-335

[2] D.Manivannan, R.Sujarani "*LightWeight and Secure Database Encryption using TSFS*

Algorithm” [6] Li Qian, Jun Hu, Shuying Liu “*SQL Injection Attack and Prevention Technology*” International Conference on Estimation, Detection and Information Fusion(ICEDIF 2015) -p-303-307

[3] System and method for communicating student information among students, parents, guardians and educators.(US 20060127870 A1)

[4] Web service pi for student information and course management systems(US20080085502A1) [5] Student specific Information System.(US20120237917A1)