Lecture Management System

“Lecture Stream”

A Project Report Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Master of Computer Application

Submitted By

Name: Suraj Kumar

Roll No: 21MCA008

Faculty of Computing and Information Technology

Usha Martin University, Ranchi

January, 2023

BONAFIDE CERTIFCATE

Certified that this project report titled Lecture Management named “Skill - up” is the bonafide work of suraj Kumar who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

(Signature of the concerned Supervisor of the Organization with Organization Seal)

(Certificate to be countersigned by the HOD.)

DECLARATION

I do hereby declare that this report entitled Lecture Management "E-Learning App", submitted by Mr. Suraj Kumar, bearing Roll No: 21MCA008 in the fulfillment of the requirement for the degree of Master of Computer Application to Usha Martin University, Ranchi, is my own and it is not submitted to any other institute.

Name: Suraj Kumar

Roll No: 21MCA008

CERTIFICATE

This is to certify that entitled E-Learning Application "Lecture Management System" being submitted by Suraj Kumar, bearing Roll No- 21MCA008, in the fulfillment of the requirement for the degree of Master of Computer Application to Usha Martin University, Ranchi, is a bonafide work carried out under my/our supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

(Dr. Mohan Prakash) **(Dr. Sharmistha Roy)**

Co-Guide **Guide**

(Dr. Sharmistha Roy)

HOD **External Examiner**

ACKNOWLEDGEMENTS

I express my deepest sense of gratitude to my guide **Dr. Sharmistha Roy** and co-guide **Dr. Mohan Prakash** Faculty of Computing and Information Technology, Usha Martin University, Ranchi, for suggesting the subject of work and constant supervision throughout this work. His/her co-operation and timely suggestions have been unparalleled stimuli for me to travel eventually towards the completion of this project report. Indeed his/her continuous involvement has helped me in bringing of this project work which otherwise would have remained a distant dream.

I am indeed thankful to **Dr. Sharmistha Roy**, HOD, Faculty of Computing and Information Technology, Usha Martin University, Ranchi, for giving me permission to carry out my project work. I would like to express my gratitude to all teaching and non-teaching staff members of Faculty of Computing and Information Technology, Usha Martin University, Ranchi, for their co-operation in my work.

Name: Suraj Kumar

Roll No: 21MCA008

Abstract

Online communication is ruling the world today. Especially during the COVID times we have seen a drastic closing of educational institutions, offices and other places of employment and exchange. There was an increased challenge of conducting discussions and communicating with people of all sections, increased problems of discourse like taking classes in schools, colleges and Universities due to lack of interactive methods accessible to the masses. Of course we need to say that there were some online tools for meeting with each other like Skype, Google duo and some other paid tools. However the use of these tools were restricted to a few people and it was difficult to organize big lengthy sessions using such methods. The first lock downs promoted software developers throughout the world to organize and develop better tools and methods for online communication among a large number of people. This led to the development of many user friendly software applications and the research to improve these continue till date. In this project we have tried to build a software that aims to provide streaming video for and users. This streaming videos maybe weed life or even maybe recorded for viewing later. The development work has mainly three parts the first one which consists of the front end of the software and has been built upon the very modern and popular JavaScript framework React. The second part is basically JSON server which stores the data about the different video-streams and which allows the admin to add, delete or modify streams as per the requirement. The third part consists of a video streaming server which actually implements the streaming part and enables the admin to stream a live video or recorded one for future requirements. The other users can view the videos as per the requirement but only the admin is allowed to make changes. The authentication system that has been used for this software is the Google auth API which allows authentication uniquely using email ID and Google password. The objective is to provide a simple and lightweight environment for one to many communication which may be suitable for purposes like taking classes or addressing gatherings

Table of Contents

Chapter 1

INTRODUCTION

1.1 Background

An e-learning history timeline

Long before the internet was launched, distance courses were being offered to provide students with education on particular subjects or skills. In the 1840′s Isaac Pitman taught his pupils shorthand via correspondence. This form of symbolic writing was designed to improve writing speed and was popular amongst secretaries, journalists, and other individuals who did a great deal of note taking or writing. Pitman, who was a qualified teacher, was sent completed assignments by mail and he would then send his students more work to be finished using the same system.

In 1924, the first testing machine was invented. This device allowed students to test themselves. Then, in 1954, BF Skinner, a Harvard Professor, invented the “teaching machine”, which enabled schools to administer programmed instruction to their students. It wasn’t until 1960 however that the first computer based training program was introduced to the world. This computer-based training program (or CBT program) was known as PLATO-Programmed Logic for Automated Teaching Operations. It was originally designed for students attending the University of Illinois, but ended up being used in schools throughout the area.

The first online learning systems were really only set up to deliver information to students but as we entered the 70s online learning started to become more interactive. In Britain, the Open University was keen to take advantage of e-learning. Their system of education has always been primarily focused on learning at a distance. In the past, course materials were delivered by post and correspondence with tutors was via mail. With the internet, the Open University began to offer a wider range of interactive educational experiences as well as faster correspondence with students via email etc.

Online learning today

With the introduction of the computer and internet in the late 20th century, e-learning tools and delivery methods expanded. The first MAC in the 1980′s enabled individuals to have computers in their homes, making it easier for them to learn about particular subjects and develop certain skill sets. Then, in the following decade, virtual learning environments began to truly thrive, with people gaining access to a wealth of online information and e-learning opportunities.

By the early 90s, several schools had been set up that delivered courses online only, making the most of the internet and bringing education to people who wouldn't previously have been able to attend a college due to geographical or time constraints. Technological advancements also helped educational establishments reduce the costs of distance learning, a saving that would also be passed on to the students – helping bring education to a wider audience.

In the 2000′s, businesses began using e-learning to train their employees. New and experienced workers alike now had the opportunity to improve upon their industry knowledge base and expand their skill sets. At home, individuals were granted access to programs that offered them the ability to earn online degrees and enrich their lives through expanded knowledge.

1.2 Objectives

There are certain goals when it comes to eLearning and some of these are to:

Enhance the quality of learning and teaching

Meet the learning style or needs of students

Improve the efficiency and effectiveness

Improve user-accessibility and time flexibility to engage learners in the learning process

eLearning is vast and an expanding platform with huge prospective in higher education. Since there are many challenges in making eLearning effective, it is important to know how to manage it and access to the resources. Take a minute and just imagine if one is not having the roadmap to guide from start to finish is actually like plunging into eLearning without an effective strategy because learners would be lost in the learning content.

The Objectives of the App:

1. The first, and main objective of the app is to get online lecture and with related notes.
2. To offer students a flexible learning schedule without any time limit to complete the course.
3. To allows instructors to build online courses on their preferred topics.
4. To provide access to thousands of courses produced by independent instructors to millions of students.

1.2.1 Scope of the project

It may help in collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to E-learning Management System. It will also reduce a cost of collecting the management and collection procedure will go on smoothly

Our project aims at Learning process automation, i.e I have tried to computerize various process of E-learning management System

- In this many student can take class in single time.
- Real time progress report of student can be tracked via one to one communication
- Students or User can read article on problem which decrease the unintelligence
- It satisfy the user requirement
- Be easy to understand by user and operator
- Be easy to operate, Have a good user interface
- It is open source and free to access

1.2.2 Reports of the Lecture Management system

- With the help of this application Mentor or teacher can generate report of student performance
- Teacher can take QUIZ and VIVA on One –To-One
- User can also take notes which is shared by the mentor
- Application also provide blog or article related to topics

1.2.3 Modules of the project

- Service Module: One to one mentorship
- Class Module: One to many mentorship or Class
- Contact Module: Contact any time with contact options
- Chat Module: You can do chat with chat option during class
- Register Module: For managing the registered-User details
- Login Module: For managing the login details

Focused Modules

- Registration
  In this, first the interested student get registered by filling their details like name, email, profession, phone, password
  Then each user profile will be maintained which can be edited by the user when desired. Each person can be registered only one time. Details of each person with their username and password is saved permanently in the database.
- Login
  After providing the correct username and password, the user log's in to the Lecture management system's Home page. There the user can watch videos, read article or blog and can get a notes which is provided by the mentor.

  If user enter wrong username and password then they cannot get the access of feature of application
- Dashboard
  After providing the correct username or email and password, the user redirected to dashboard page and there they can get access of all feature of app.
  Here, at this page user can learn about many technologies like Java, Spring-Boot, Node-js, React-js and many more

Chapter 2

TECHNOLOGIES

The following are the system requirements to run our project.

2.1 Software Technologies:

Front End: React-js, Web Browser (Internet explorer, Mozilla Firefox or Chrome), Command Prompt, Visual Studio Code.

Back End **: Node-js**

Server side **: Express-js**

Database **:Mongo-Db**

Platform : Website.

2.2 Hardware technologies:

For Computers:

RAM : 2 GB or more

PC : **Intel® Core™ i3**

HDD : **500MB or more**

Monitor : 15" at least VGA/SVGA/PGA (color preferred)

For Android:

Ram: 2GB or more

Processor: MediaTek P22 or above(recommended)

Operating System: Above Android 4.0

3.1 Requirement analysis:

As the literature review indicated, there is a strong leaning towards positive outcome from using tablets in educational institutions, even when provided with little or no customization. In fact, the team could not find one academic publication, which has dismissed the use of tablets in academia as having negative or highly undesirable outcome. Some of the negative publicity has been reported in the media and the team considered this part in their review. As such, the team decided to focus on redesigning the tablet that would make it better fit for purpose in an educational context. The team looked at the three key layers of the system: Application/Apps layer, Operating System layer, and Machine layer.

- Application/Apps layer:

  The primary purpose of the tablet is to support educational apps and apps of educational value. In addition, full running application that has an educational endeavor should be a suitable application to be considered. The focus group reported variety of educational applications that could be categorized as into two: vertical support and horizontal support applications: - Vertical support applications being specific to courses, example such as Microsoft Access

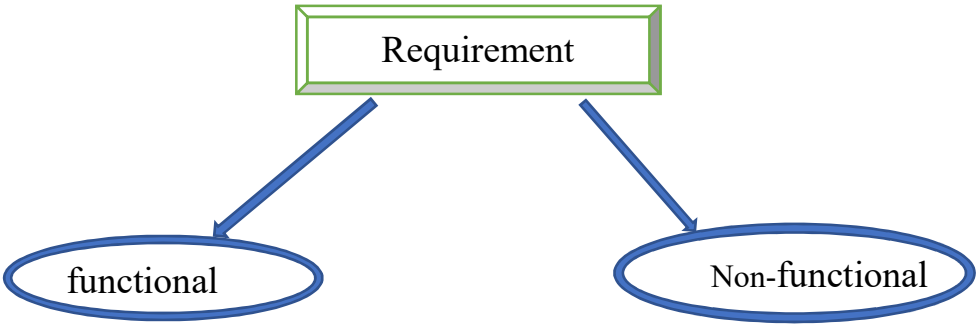
(database design), financial calculator, and Adobe Flash (Animation). In addition, a vertical application can be developed for the sole purpose of the course. - Horizontal support applications being useful and in use across variety of courses department, notable examples are such as Microsoft Word, standard calculator, and English dictionary. Built-in access to educational applications or applications for educational purposes is seen as essential for the success of the tablet as an educational tool. This includes having built-in shortcuts to the institution's own Virtual Learning Environment, electronic email, calendar, electronic library, and even direct access to highly academic online sources

- Operating System & System Administration layer
  The tablet's operating system has to be robust, and provide support for the apps/applications mentioned above so that it would truly support the needs of the students as a replacement for their desktops, laptops, and mobile phones. The focus group noted that neither Google's Android nor Apple's iOS operating systems are able to support the variety of vertical support applications needed. In reviewing Android and Apple iOS, both operating systems are designed from an entertainment point of view and some of their built-in applications present risks for misuse of the tablets in particularly download of games. Neither system had built-in administration tools, although some applications are available to create custom restrictions but these were not well tested. In addition, the operating system should support the creation of users with editable permissions. Microsoft Windows operating system appeared to respond more favourably in the review. The operating system already supported much of the applications the educational institution has and could have an administration level that matches already what the educational institution has for their current computer labs. As a result, the recommendation is that the tablet should have an operating system that matches the one already in use by the educational institution. When using the same operating system the university is using on its desktops in the library or in the labs, students would be more familiar with that use. Being more familiar will reduce the time for getting accustomed to the use and would not require extensive training. Furthermore, an operating system that would allow the IT department of the institution to push applications and push updates would ensure future changes in the requirements of the institutions go much more efficiently. Here the team noted, and still not tested, Microsoft Windows 10 pro as having the right potentials. The team reported an operating system requirement that should be part of future designs for educational operating system and that has to do with device tracking, locking, and erasing content

- . Machine layer:
  The choice of machine has to be robust, and provide true replacement for using desktops, laptops, and even mobile phones for educational purposes. The focus group noted recommendations from the literature on providing tablets stylus pen and training on using the machine. The team recommended the tablets to have

some key features: USB port for transfer of assignments, extended memory option, headphone port, long battery life (minimum one academic day while in use and standby of three academic day when not in use from one full charge), Bluetooth or lock-in keyboard, Near Field Communication, light weight (comparable to the average weight of a book), smooth scrolling, wireless network and mobile network access, be tested thoroughly for reliability with the list of identified applications, and comes with at least one year warranty. Other key recommendations would depend on the institution's needs. The team noted some features that are not essential and could be removed. These features are not required for academic purposes and, if removed, could in fact reduce the cost and weight of the tablet: the GPS, the outward camera, and inward camera

Further The Requirement Is divided into Two types:



3.1.1 Functional Requirement of application:

Students can choose courses, attend lectures, take viva and oral quizes, take feedback on their progress records, etc. as per their convenience.

Attend lectures either at the scheduled time or on request view lecture at a later time.

Faculties can take lectures, upload assignments, announcements, evaluate answer sheets and also can upload lectures and other discussions in various formats as in videos, power point presentation etc.

Upload of various assignments, University notices, student' snotices, journals, videos.

Real time collaboration among users via chat rooms, shared and interactive whiteboards.

Asynchronous communication in the form of Emails, discussion boards that enable communication to occur at "convenient-times" that suit student schedules and are not accessed at simultaneous or prearranged times.

There can be forums, blogs etc. to discuss various queries and to put up suggestions posted both by students and teachers.

Administrators can generate reports, log files, backup/recovery of data at any time.

Shared documents and media library that can help in active learning of a student.

Image library.

One-to-many, many-to-one and many-to-many information sharing.

Availability of voice mail box to allow faculties to get the descriptive messages left by the students.

Provisions of resources to arouse the interest of students in extra-curricular activities like public speaking and grasp the chance to enhance their personalities.

Students can take up various quizzes which can help them realize their inbuilt talents in various fields.

3.1.2 Non-Functional Requirement of application:

Performance Requirements:

The application should be able to operate on all major web-browsers with all of its fundamental functions. It should not slow-down the system even at peak hours without affecting the quality of service of the system.

Safety and Security Requirements

The server on which the E-learning Web-application will have its own security to prevent unauthorized write/delete access.

The system should provide a secure login to the users by using advanced secure login algorithms and provide access only to the authorized users as security is the key requirement of this system.

The user ID and the password should not be shared with anyone (students/faculty/or anyone else)

3.2 Definition:

E-learning is "utilizing electronic technologies to access educational curriculum outside of a traditional classroom," according to North Carolina's eLearning education initiative. E-learning courses or programs are generally based online.

E-learning is a type of learning conducted digitally via electronic media, typically involving the internet.

It can be accessed via most electronic devices including a computer, laptop, tablet or smartphone, making it a versatile and easy way for students to learn wherever they are. E-learning resources come in a variety of forms – from software programmes and digital courses to interactive online platform and apps.

Below are some common types of e-learning methods and the differences between them.

DIGITAL MATERIALS

E-learning can be carried out via the consumption of videos, PDF documents, slides-shows and word documents. Thanks to the availability of these resources, it's incredibly easy for anybody to teach themselves a new skill at their own pace.

ONLINE COURSES

Online courses are often provided by Learning Management Systems (LMSs) and allow learning material to be delivered at a steady rate, organized into sections and chunks to make it easier for the learner.

They often come with interactive materials to allow the learner to test and apply their own knowledge.

Popular online course providers include Skillshare and Udemy.

3.3 Purpose of e-learning:

The idea of e-learning is to empower learners to absorb personal accomplishment, basic schooling or to obtain a degree certificate, without actually attending the school or university or any other academic institute.

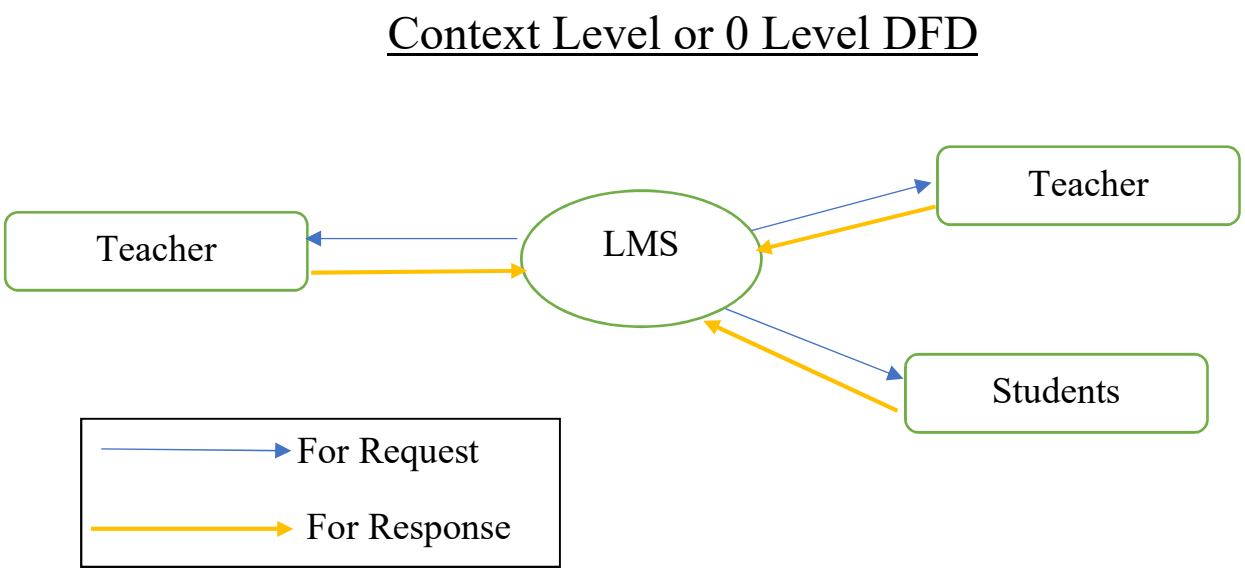
Another idea is to apply E-learning to all levels of schooling to ensure students grasp the lessons adequately.

Psychologists believe that audio-visual method of teaching creates a disciplined learning environment and fosters effective student engagement in the class.

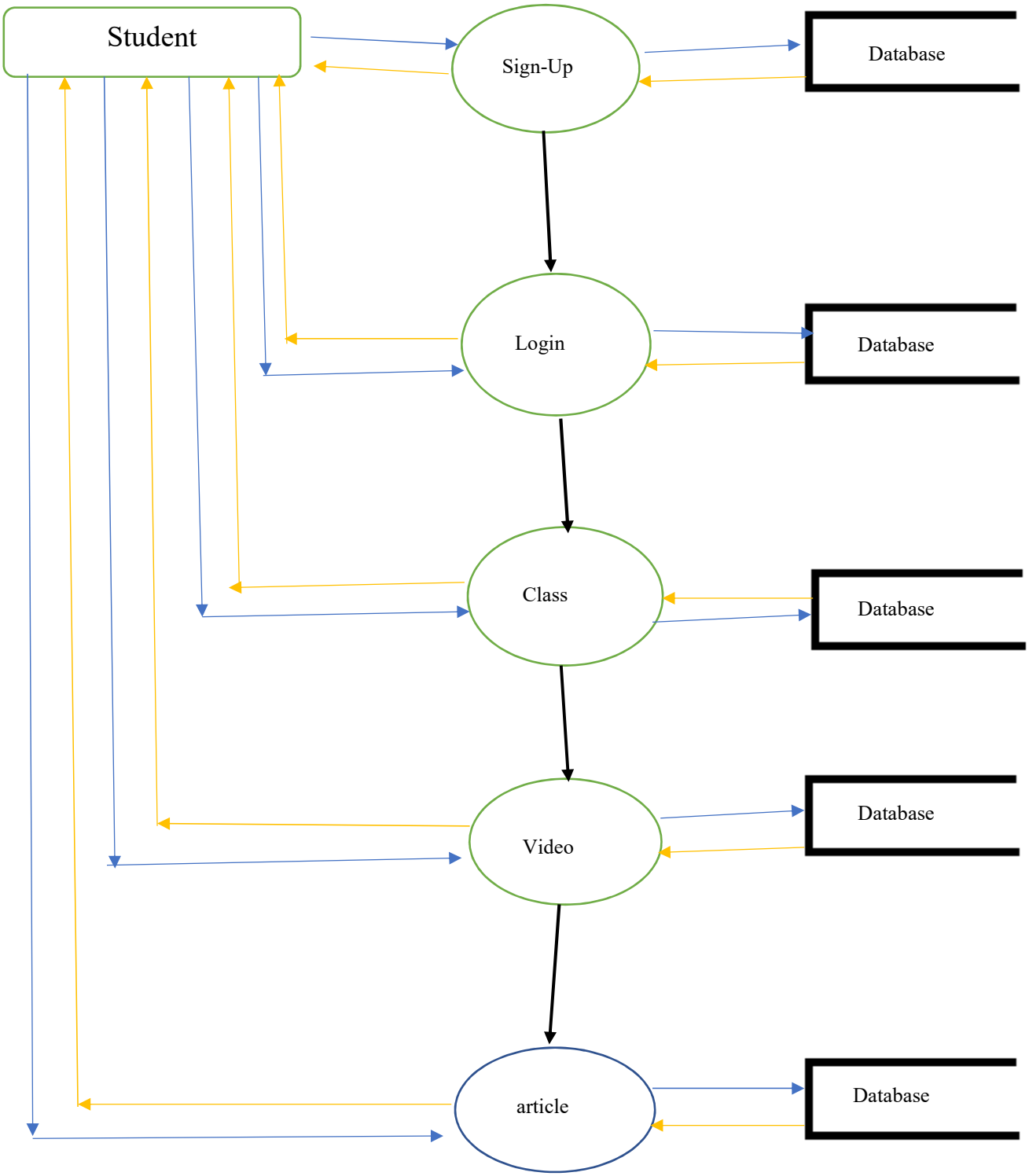
The next benefit of learning online or on electronic media assures you are in sync with advanced learners. Also, digital and self-initiated learning can be acquired at the desired locations. One need not wander in search of learning.

eLearning is versatile and ample to suit all learning methods.

3.2.1 DFD Diagram:



DFD for Student and Teacher

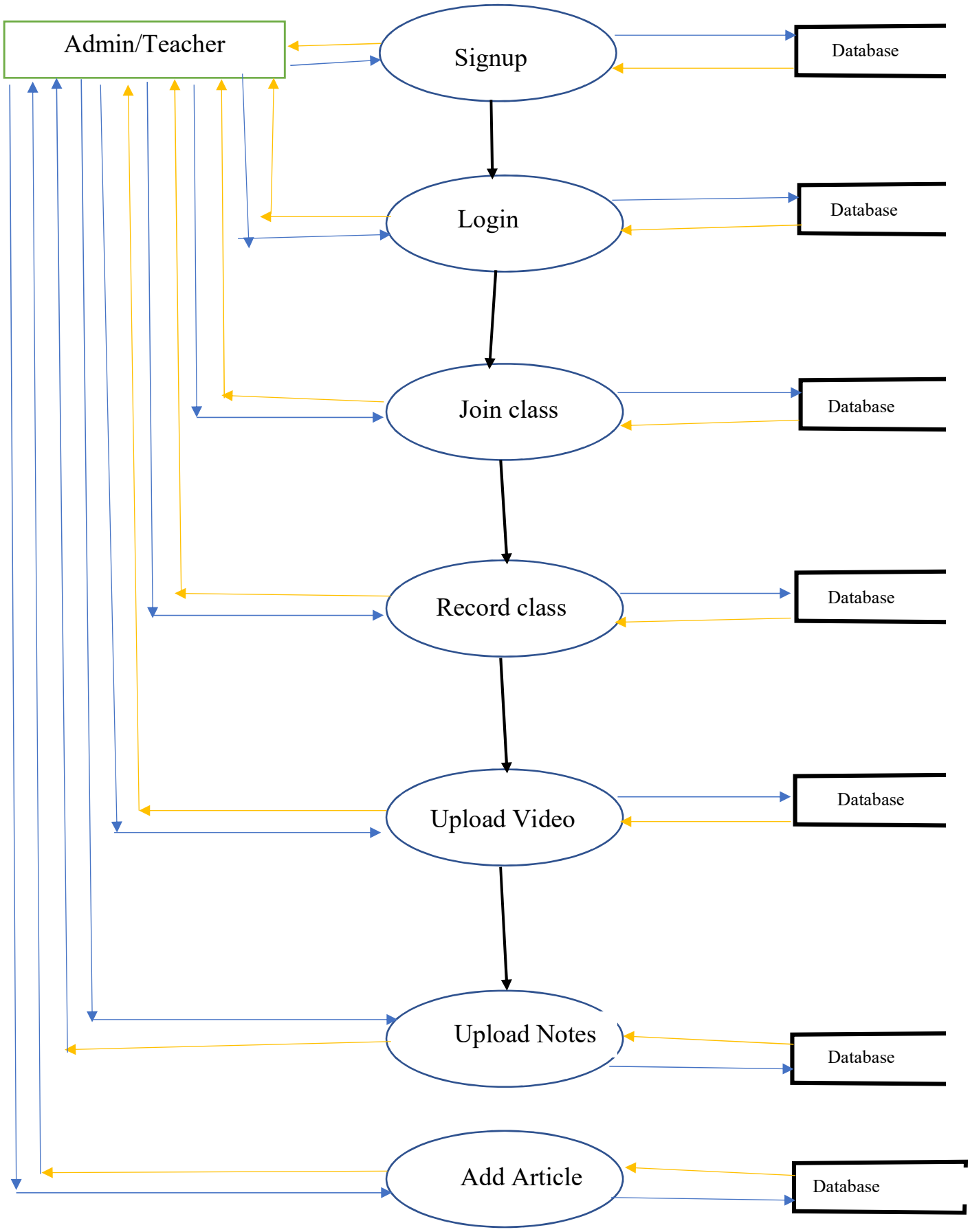


For Request

For Response

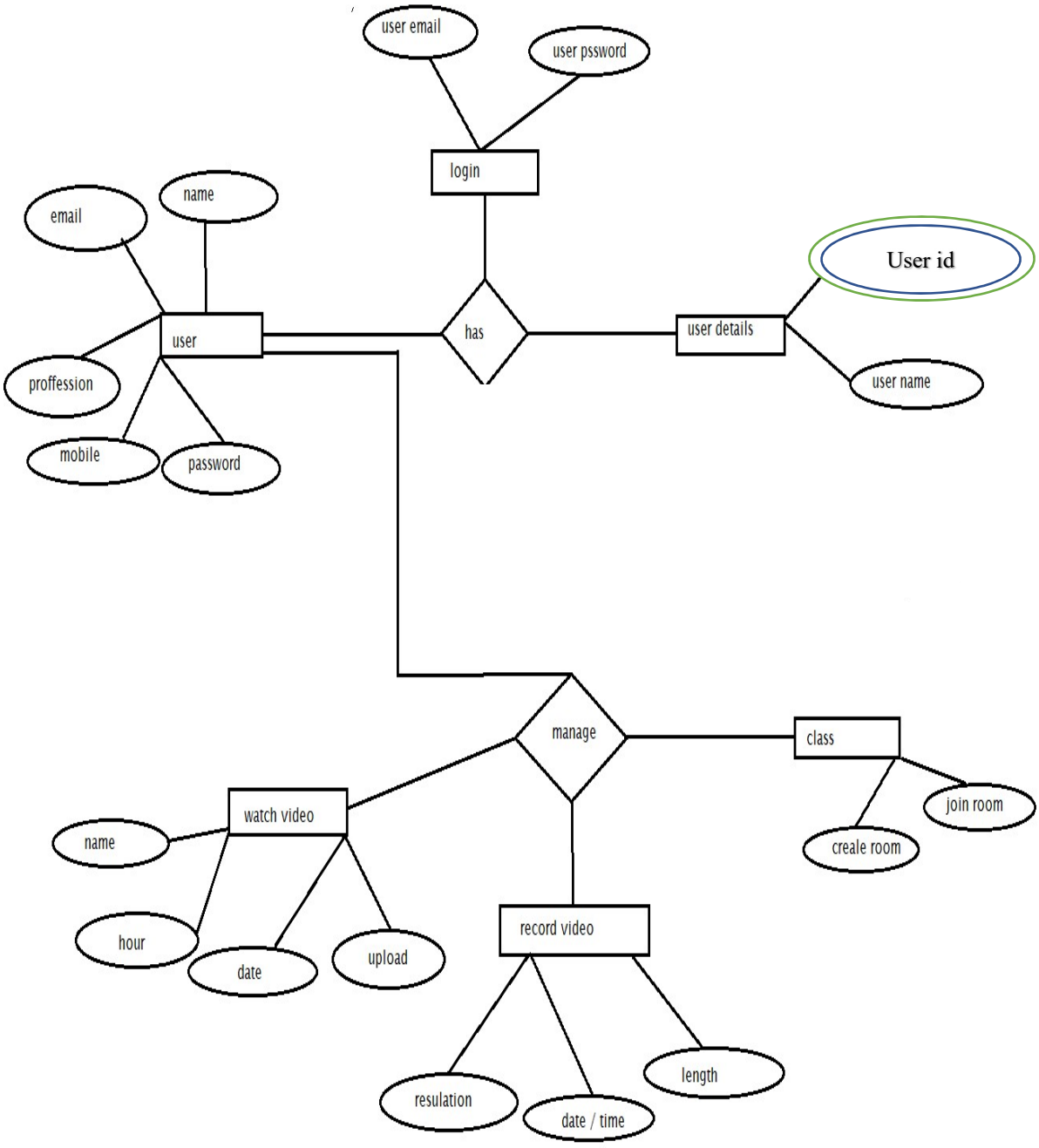
For Database

Database


Admin/Teacher
Signup
Database
Login
Database
Join class
Database
Record class
Database
Upload Video
Database
Upload Notes
Database
Add Article
Database

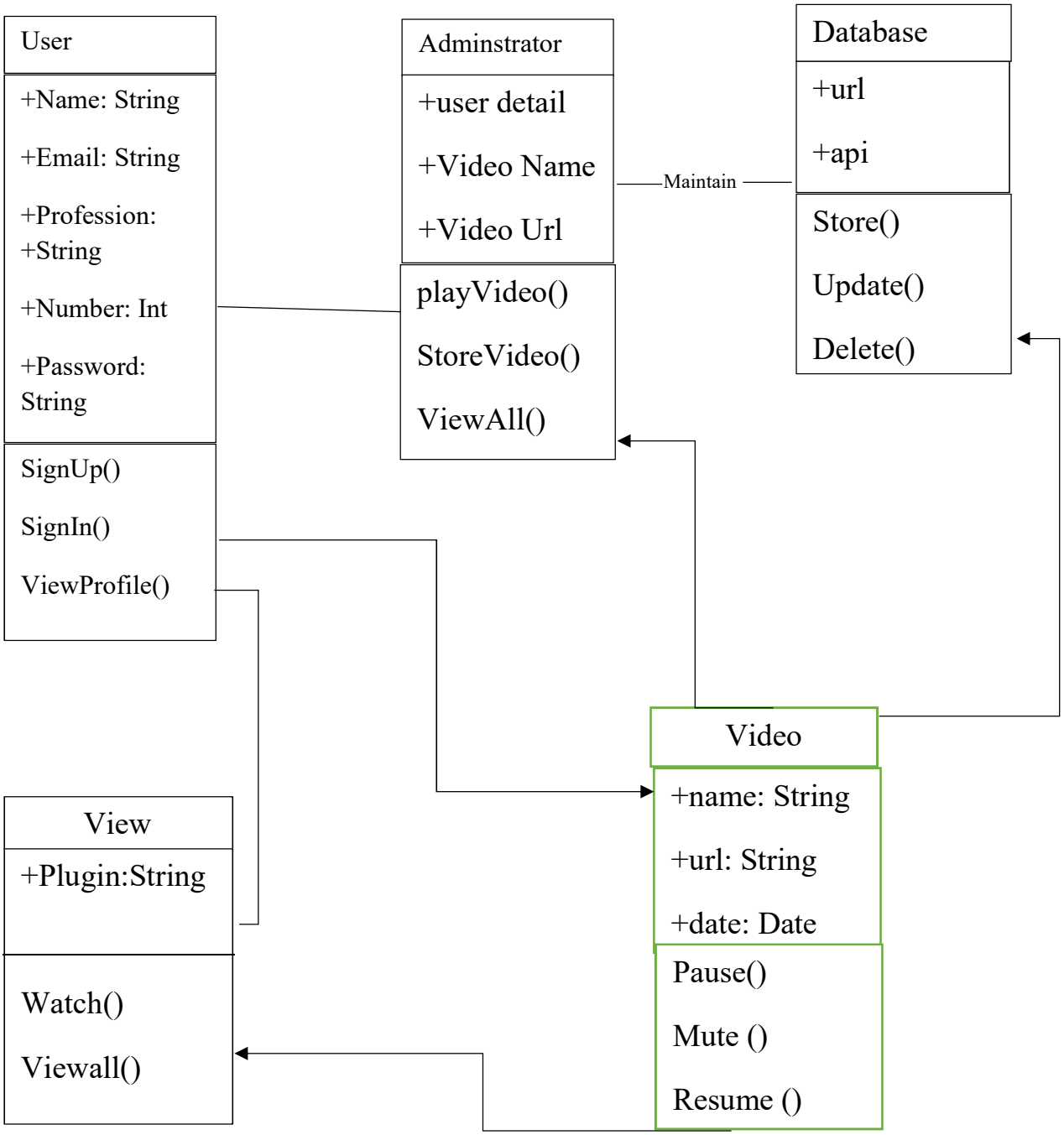
3.2.2 E.R Diagram:

E.R Diagram of application



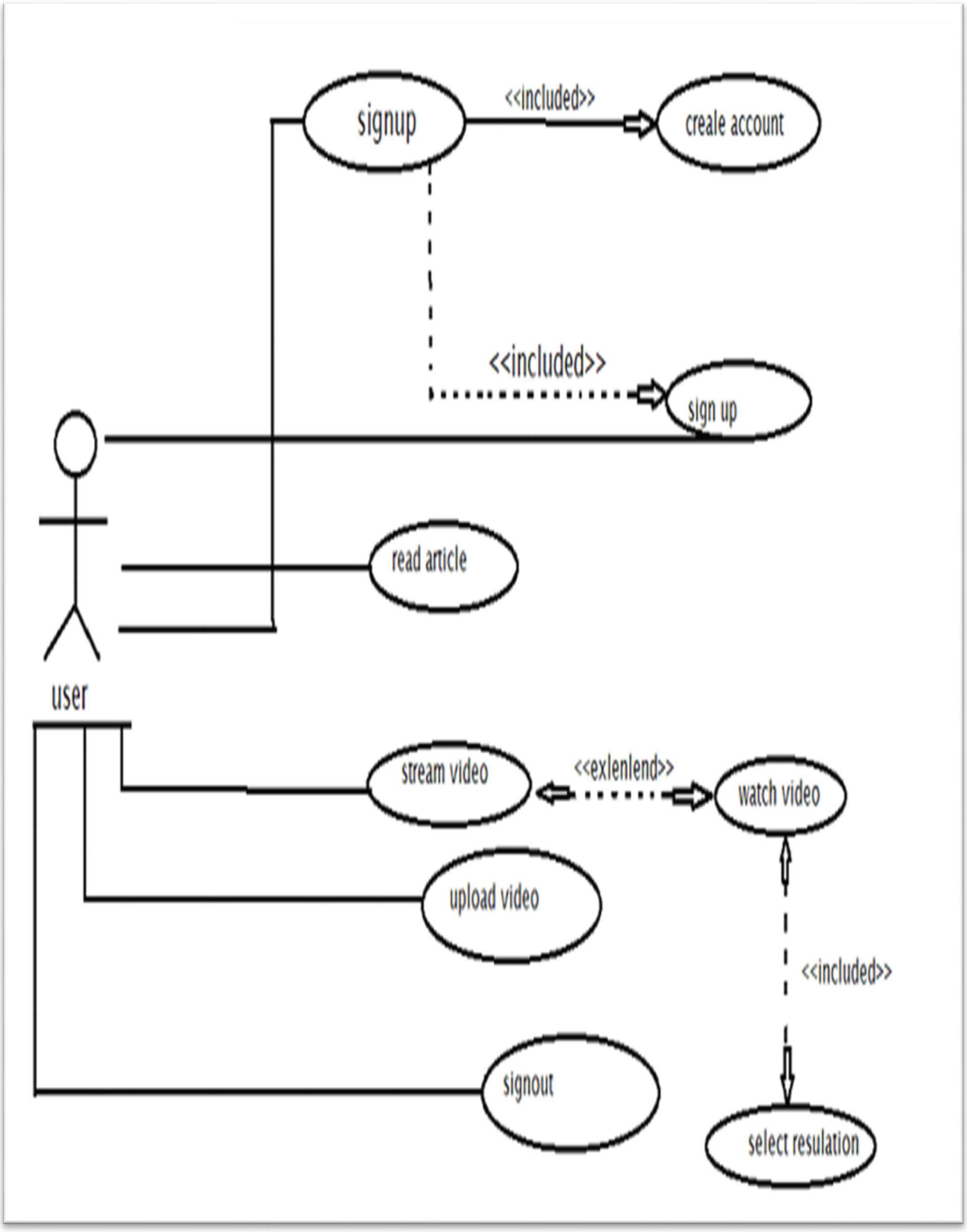
3.2.3 Class Diagram:

Class Diagram of application



3.2.4 U M L Diagram:

UML Diagram using Video Database



4.1 Development Phase of Application:

Here, there are 6 common phases that any eLearning development process must undergo. What are they, why are they important, and where in the development process do they fit in?

The Instructional Design (ID) world has been influenced by a fair share of Instructional Design models such as the ADDIE Model, D. Carey and J. Carey's Model, Gagne's nine-step model, and SAM - the latest one. While the Instructional Design model followed by each organization differs, the model itself is followed quite rigidly. This is not the case with the eLearning development process; each organization follows its own process – one that fits into its way of functioning. But even with each company following its own eLearning development process, there are some common phases that must be followed.

The 6 common phases that all eLearning development processes must go through are:

1. Creation of the Instructional Design Document (IDD).
2. Scripting.
3. Development of a prototype.
4. Development of the course without audio.
5. Development of the course with audio.
6. Creation of the Learning Management System version of the course.

Phase #1: Creation Of The Instructional Design Document (IDD)

The IDD is a high level plan of the overall instructional approach of the course. Stakeholders and the Instructional Design team sign off on the global instructional approach – some common approaches are problem solving, avatars, and scenarios.

It is also decided how the content must be broken down – into courses, modules, and even screens. If several courses are to be created, and each course is to be created using a different Instructional Design strategy, these must be dealt with in the IDD. The instructional and visual elements are finalized at this stage.

Phase #2: Scripting

The content to be used for each course is broadly finalized and divided into modules; and the content for each screen within a module is decided on.

The interactivities, assessments, activities, and resources for each module are finalized. The narrative script (if used) is finalized as well. The scripting document is usually a Word document, although a PowerPoint presentation works just as well.

Phase #3: Development Of A Prototype

This is a complete representation of the entire course in just a few slides. The onscreen content and the audio script are created and finalized. Important elements that will be used in the course are picked out and a functional prototype is created. Decisions are made and finalized regarding how content is going to be presented onscreen, the interactivities to be used, the activities/assessments to be added, colors, images, animations, and the additional resources to be provided.

A lot of writing, rewriting, and revisions take place during this stage. The prototype should be functional on the Learning Management System.

Phase #4: Development Of The course Without Audio

During this phase, a fully functional course without audio is presented to the stakeholders, for their approval. Even at this stage, stakeholders tend to make changes to the content, narration, animations, etc.

It is expensive to create audio and then recreate it with iterations. So the audio file is not created until the audio script is finalized.

Phase #5: Development Of The Course With Audio

Once the audio script is finalized (the previous step), a fully functional course with the audio is sent to the stakeholders for approval. When the audio is developed, it is synchronized with the animations and onscreen text; audio quality and pronunciations are important, and must be checked to make sure it is right.

Phase #6: Creation Of The Learning Management System Version

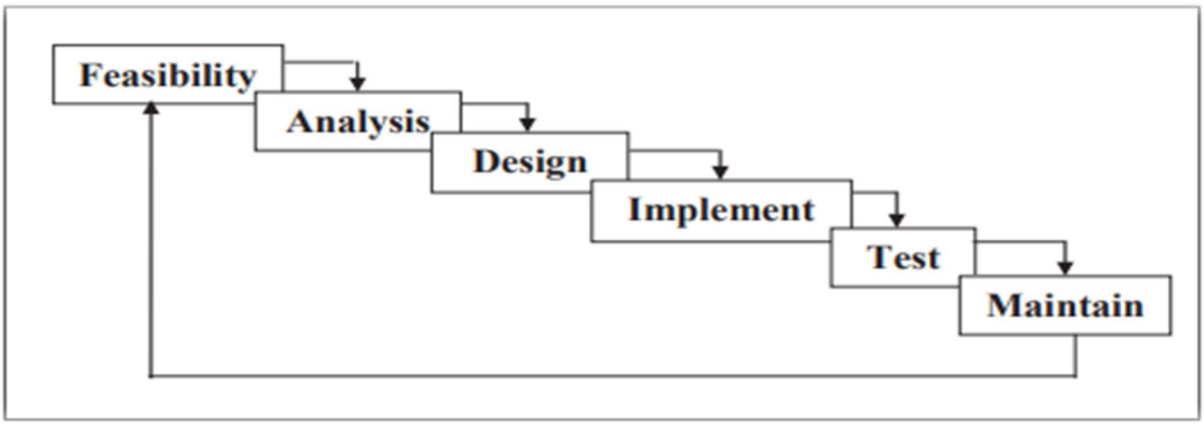
While the first 5 phases of the eLearning development process are mostly creative in nature, this final step is the mechanical and more technical part of creating an eLearning course. Once the stakeholders signoff on the course, the course (AICC/SCORM/XAPI compliant version) is ready to be hosted on the learning platform and go "live". This is taken care of by the technical team.

4.2 S D L C (software development Life Cycle):

A system approach was taken to develop information system which is known as System Development Life Cycle (SDLC).It is a step-by-step process which refers to a methodology for developing systems. It provides a consistent framework of tasks needed to develop systems.

The SDLC methodology can be reduced to include only those activities that are appropriate for a particular project, whether the system is automated or manual, whether it is a new

system or an enhancement to existing systems. SDLC methodology tracks a project from an idea developed by the user, through a feasibility study, systems analysis and design, programming, pilot testing, implementation, and post-implementation analysis. Documentation developed during the project development is used in future when the system is re-assessed for its continuation, modification, or deletion



The System Development Life Cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application. Various SDLC methodologies have been developed to guide the processes involved including the waterfall model (the original SDLC method), rapid application development (RAD), joint application development (JAD), the fountain model and the spiral model (Whitten, Bentley, & Dittman, 2004). Several models are combined mostly into some sort of hybrid methodology.

Documentation is crucial regardless of the type of model chosen or devised for any application, and is usually done in parallel with the development process. Some methods work better for specific types of projects, but in the final analysis, the most important factor for the success of a project may be how closely a particular plan was followed.

The system development life cycle (SDLC) is the entire process of formal, logical steps taken to develop a software product. The phases of SDLC can vary somewhat but generally include the following:

1. Conceptualizations

2. Requirements and cost/benefits analysis;

3. Detailed specification of the software requirements;

4. Software design;

5. Programming;

6. Testing;

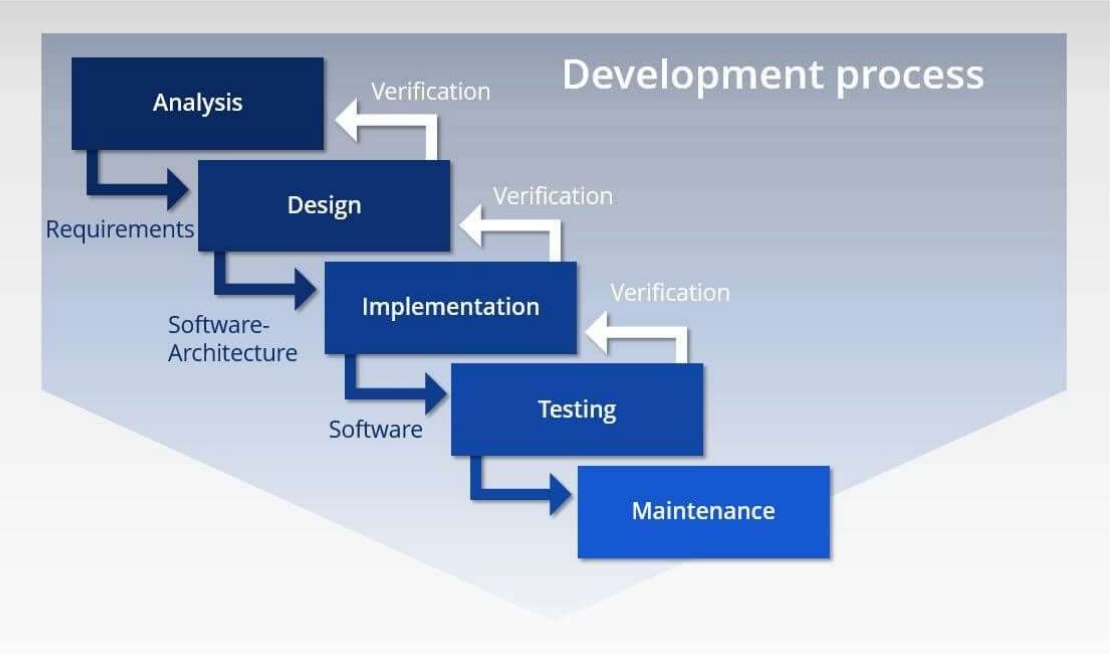
7. User and technical training;

8. And finally, maintenance.

The SDLC which is followed for the developing of this project is:

Waterfall-Model:

The Waterfall Model has been used for more than 40 years. Thus, it is a time-tested approach that has been finely tuned over the decades. As its name suggests, this development model centers on a step-by-step design process. Each stage must be completed before moving onto the next, which makes it ideal for Instructional Designers who prefer a more linear and organized strategy. The Waterfall Model relies heavily on the planning and research aspects of eLearning course design, so that you are already aware of your learner's needs and the problems that must be addressed before you create your first prototype. This allows you to create a learner-centered eLearning course that offers the best ROI.



Reason why I had implemented Waterfall Model !

1. **Cost effective.**
   One of the most notable benefits of using the Waterfall Model to create your eLearning course is that it can reduce development costs. You must devote a significant amount of time to researching needs, planning activities, and assessing knowledge gaps. As such, by the time you actually start the design process you have a great deal of data at your disposal. This means that you won't have to endure a lengthy and costly revision process. E-Learning professionals are able to detect any problems early on and remedy them right away.
2. **Streamlines the work process:**

All members of your eLearning team are on the same page from the very beginning of the development process. This is due to the fact that you must assess training needs and determine goals within the very first phase. You can meet with clients early in the process to identify their expectations, instead of designing the entire eLearning course only to discover that they are looking for something completely different.

3. **Improves collaboration**.

   The Waterfall Model allows all collaborators in your eLearning team to work on their aspect of the eLearning project autonomously. Once you've clarified what the eLearning program needs to accomplish, everyone can dive into their respective tasks in order to speed up the process. For example, your graphic designers can already begin creating the presentations and template, while the SME can gather all of the necessary information and identify the key takeaways.

4. **Ensures cohesiveness.**

   This approach is linear, which gives you the chance to work on every aspect of the eLearning project in order of importance. Rather than piecing all of the elements together at the end, you can figure out how all of the components are going to fit into the overall puzzle from day one. Thus, your eLearning program will be more cohesiveness and organized. Employees will be able to see how each of the ideas or concepts connect with one another, instead of having to sit through a disjointed eLearning experience.

4.4 Waterfall In Instructional Design:

This Instructional Design process involves six key stages that takes a linear approach. Here are the 6 phases that you should consider when creating your next eLearning course:

Needs Analysis

Assess the needs of your learners and identify the primary goals and objectives of the eLearning course. This may involve surveys, online tests, focus groups, interviews, and on-the-job observations. The training needs analysis should focus on your learner's strengths and areas for improvement.

Knowledge Analysis

After determining the needs of your learners you must identify how you are going to fill the performance and knowledge gaps. In other words, how are you going to get from point A to point B so that your eLearning program is truly effective? Conduct tasks and skills assessments to figure out how you can improve their productivity and give them the tools and resources they need.

Identify Limitations

Every eLearning program has its limitations. This may come in the form of a tight eLearning budget, technology constraints, or busy schedules. Once you identify these limitations you can decide whether they have a viable solution, or if you need to factor them into your eLearning course design. For example, if you dealing with learners who may have limited tech know-how, you can remedy this by offering tutorials that teach them how to use the LMS and use devices they are most familiar with.

Content Development

Identify the ideal eLearning activities and online resources that align with your goals and objectives. Branching scenarios, virtual presentations, eLearning simulations, serious games, and eLearning videos are just some of the multimedia elements you can incorporate into your eLearning strategy. You should also decide which Instructional Design models and theories are ideally suited for your learners' needs.

Prototyping

Create a rough draft of your eLearning course to serve as a prototype, then conduct thorough testing in order to determine its effectiveness. This is also the time to work out any issues that may hinder your goals and revise the aspects of your eLearning course that aren't living up to expectations. Focus groups can be an invaluable tool at this stage, as it gives you the opportunity to gather feedback from your target audience.

Deployment

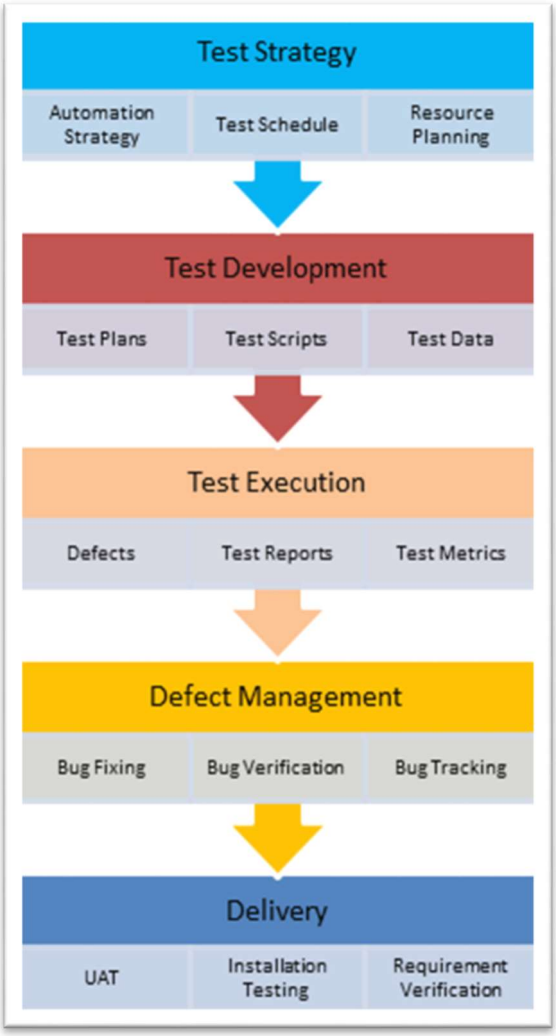
Launch your eLearning course and measure the results. This may also require revisions and eLearning course modifications. Fortunately, the Waterfall method involves a great deal of research and planning, which means that revisions should be minimal at this point.

The Waterfall model may not be for everyone, but it is a viable approach that yields a wide range of benefits for new and experienced eLearning professionals, alike. Use this article as a guide to keep your eLearning course design on track and maximize the results of your eLearning program.

At the other end of the Instructional Design spectrum is the Iterative Design approach. Read the article 5 Benefits Of Iterative Design In eLearning to discover an Instructional Design approach that allows you to focus on one eLearning course component or design phase at a time, rather than trying to develop the complete package from start to finish.

4.5 Testing Strategy Of application:

A Test Strategy is a plan for defining an approach to the Software Testing Life Cycle (STLC). It guides QA teams to define Test Coverage and testing scope. It helps testers get a clear picture of the project at any instance. The possibility of missing any test activity is very low when there is a proper test strategy in place.



Test Strategy:

It define the stage of planning were we take decision like,

Automation strategy:- where we see that how the software will work during on execution.

Test Schedule: It defines the when and how the test will be done

Resource Planning : Here we can gather our recourses for testing phase.

Test Development:

In this phase we organize the thing in such a way that it will give an effective results

Test plan: we do planning for test that when it to be done and how it to be done

Test Script: this is process were we use scripting method of SDLC

Test Data: in this we test the data (mean flow of data throughout the application)

Test Execution:

This is phase were we test the application on execution time for handling bugs and defect of the software.

Defect: in this we look for defect of software and bug on execution time of an software

Test report: while handling the defect on same time we generate the test report of software.

Defect management:

Here we manages a defect of software and remove it properly from the software.

Bug Tracking: we search for bug in application in systematic way so that it could be found easily.

Bug verification: once bug found we verify the source of bug and search for result.

Bug fixing: after getting source of bug then we remove and fix it properly and then once again we repeat process for defect management

Chapter 5

IMPLEMENTATION AND TESTING

Software Test Techniques are widely used in finance, space science, defense industry and insurance sectors. Software test techniques provide software usability, durability, optimum system performance and minimum cost. Different testing methods have been developed for E-learning Testing and Evaluation. In this study, we have developed an e-learning testing and evaluation system which is supported with software testing techniques. Validation and verification methods have been used for developing such a system. In e-learning testing, we have a checklist for page structure, question types and module transition standards. Then we validate these standards. White-box testing will be used for internal structure of the E-Learning System. Classes, functions and procedures will have been prepared for White-Box Testing. Unit testing is a procedure used to validate that individual units of source code are working properly. Overall system performance is generally assessed in terms of response time and throughput rates under differing processing and configuration conditions. E-learning system performance is measured by computing e-learning page response time and data load time. Our system is developed for measuring e-learning page load time, response time, and

average waiting time. We propose an E-learning system with test driven development. Results show that test techniques should be used for development of e-learning applications.

Keywords: E-Learning, Software Test, Evaluation, Testing, Test Techniques, V-model

5.1 Software testing technique

Software Testing Techniques

Static Test

Static tests are different at some aspects because don't demand application (Ricca &Tonella, 2001). Static test is not in-detail; instead it is interested in algorithm, document and code. Static test is applied for finding code syntax and code errors. This kind of test code is written by developer. While control process is being performed on static test, test control is performed from a preset control list. Static test control list can be seen on Table 1

Reliability	**Performance**	**Usability**
Are reliability requirements determined?	Are answers and delay time determined?	Is usability requirement determined?
Are robustness requirements determined?	Are output requirements determined?	Are color schemas and standards convenient?
Are service requirements determined?	Are data capacity requirements determined?	

Table 1: Static test control list

In the framework of this test, written code is read instead of executing. During reading, the code is examined and analyzed statically with review. Importance of static test is assuring of exception handling early in software lifecycle phase

System Test

In system test various topics are considered such as reliability, security and sustainability. How software working with different hardware is emphasized. At this point black box approach can be dealt with, while software internal structure is not considered. These tests evaluate performance, reliability, functionality of developed software. Developed

software is confirmed according to software's design in unit/class and integration tests. On the other hand, system test aims at confirming user demands.

Performance Test

Performance test is used for evaluating system test outputs that can be produced on determined and acceptable period of time .Performance test is examining system conformity of software transaction time. Performance examining is performed on every level of the test procedures. However system performance can be figured out after system integration (Software Testing - Testing Tutorials, 2012).

Unit Test

Unit testing is a method by which individual units of the software such as source code, a class, procedure or full program are taken into consideration. Unit test enhance the confidence and reliability of a system in which units of it works as expected. Unit testing can be conducted as creating suitable testing procedure. A unit is the smallest testable part of an application. In procedural programming, a unit may be an individual program, function and a procedure while in object-oriented programming; the smallest unit is a method; which may belong to a base/super class, abstract class or derived/child class. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits. Unit tests can be applied step by step manner. First of all, an essential procedure is selected and then the upper level procedures are tested respectively.

User Acceptance Test

It is performed whether a system satisfies its acceptance criteria, enable the customer to determine and inform whether accept the system. User Acceptance test should be performed on every intermediate product to validate acceptance of system. User acceptance test objectives are sorted as below:

- Validate system set-up for transactions and user access,

- Confirm use of system in performing in sequence processes,
- Verify performance on critical functions,
- Confirm integrity of converted and additional data and,
- Assess and sign off go-live readiness.

The Strength of Testing

Various test techniques allow user to develop robust software design. Static test is applied on system coding level. We can determine and remove defects according to static test results. E-learning system is composed of classes, procedures and statements. So, we can eliminate software defects via static testing. E-Learning system has not only several web pages but also has a database management system to hold required data. At this point, a system test should be used on overall system. System performance is generally assessed in terms of response time and throughput rates under differing processing and configuration conditions. The performance problems are most often come from the result of the client or server being configured inappropriately. An E-Learning System generally works based on client-server architectures. We can increase E-Learning System transaction efficiency via performance testing. E-Learning System is consisted of software modules. Each module should be tested with unit testing method. The E-Learning System has different roles. These roles use E-Learning with various authentications. To design a consistent E-Learning System, user acceptance test should be used.

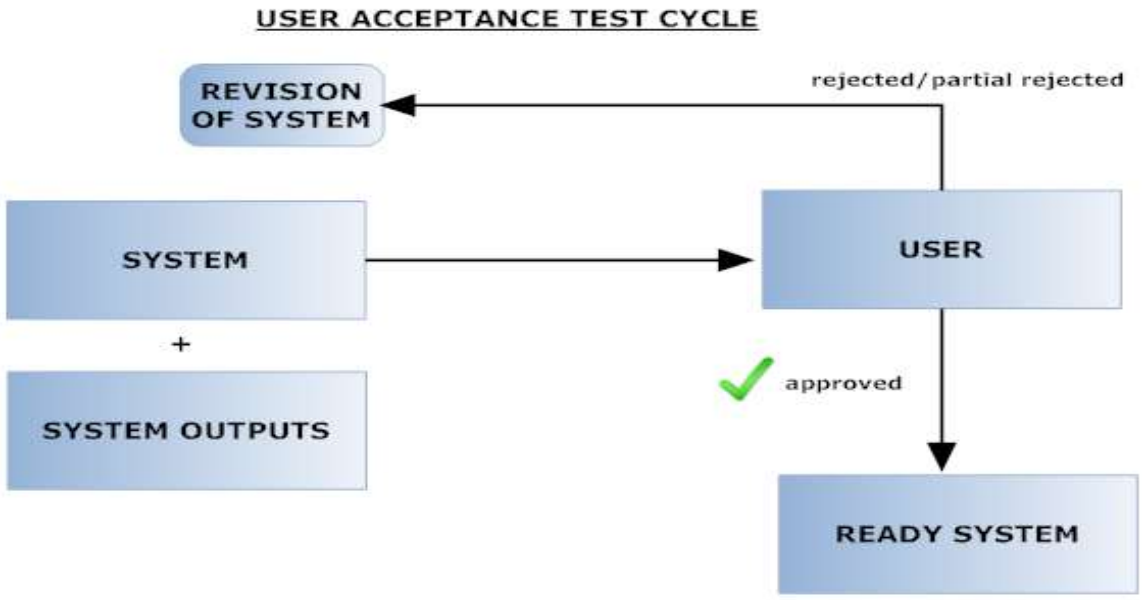


Figure 1: User Acceptance Test Cycle

5.2 About V model

V-Model

V-model as a software life-cycle model is a development of the waterfall mode. Testing activities are started with beginning from first level of the project. The process steps are determined after the coding phase to form the typical V shape and tested with required testing techniques (Mittal & Chopra, 2011).In Figure 2, a V-model phase diagram is depicted. The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing.

In requirements analysis, the requirements of the proposed system are collected by analyzing the needs of the user. This phase is related to establishing what the ideal system has to do. However, it does not determine how the software will be designed or built. Usually, the users are interviewed and user requirements document is generated. The user requirements document will typically define the system's functional, physical, interface, performance, data, and security requirements as expected by the user.

In system design phase, system designers analyze and understand the business of the proposed system by studying the user requirements document. The figure out possibilities and techniques by which the user requirements can be implemented. If any of the requirements are not feasible, the user is informed of the problem. As soon as a solution is found, the user requirement document is edited accordingly. The architecture phase can also be called as high-level design. The baseline in selecting the architecture is that it should realize all which typically consists of the list of modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, and technology details.The integration testing design is carried out in this phase. The module design can be called as low-level design. In module design phase, the functions and procedures which will be used in system, are designed and coded. So, the unit test procedures are developed in this phase. We have used V-model for E-Learning Testing and Evaluation System design and tested every intermediate product according to V-model steps.

.

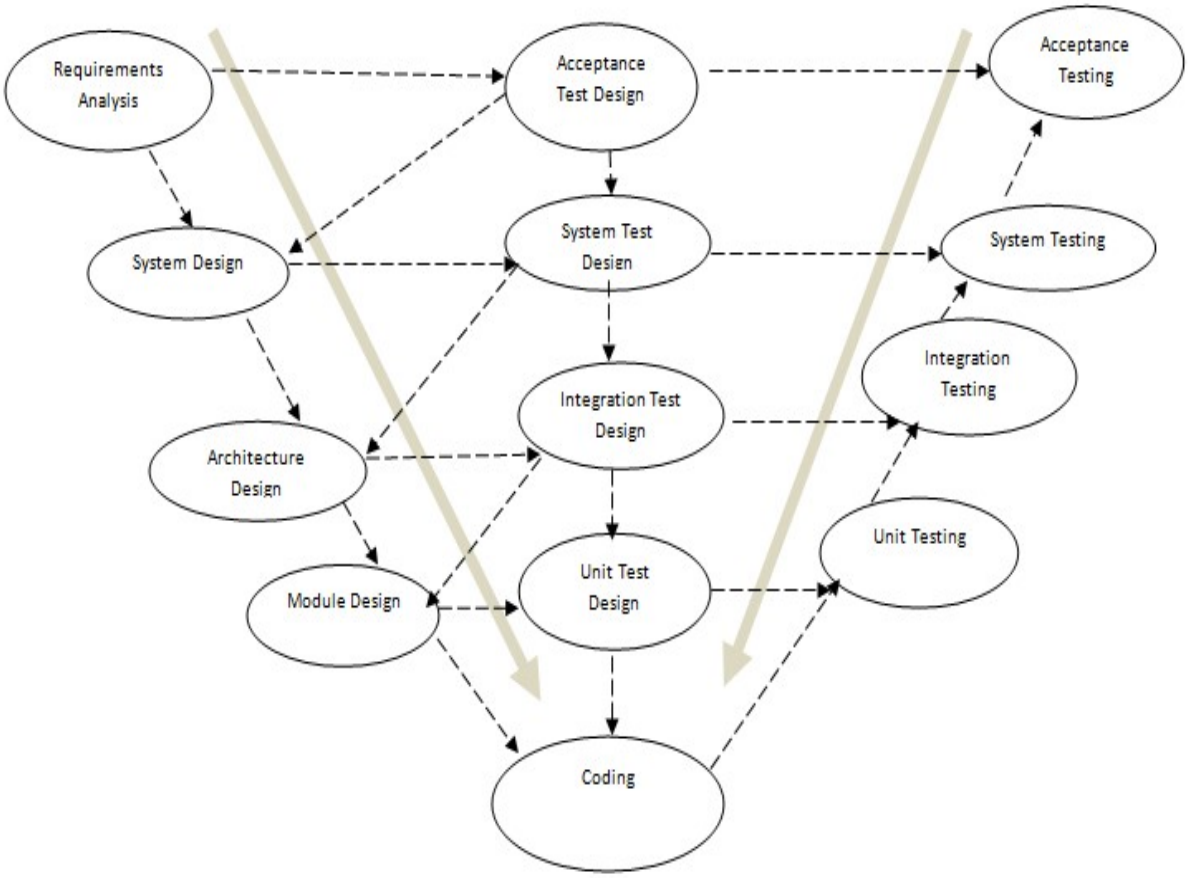


Figure 2: Software Development Life Cycle V-model

Design and Implementation of E-Learning Testing and Evaluation System

E-Learning Testing and Evaluation system has been designed by applying software test techniques on distance education software and defining required methods while this application. The following steps which we should use while designing an E-Learning Testing and Evaluation System;

- Designing E-Learning database with database normalization techniques.
- Designing E-Learning Testing and Evaluation System with tracing an eligible software development lifecycle.
- After the completion of E-Learning system design, testing web pages with software testing techniques.
- Testing E-Learning database with software testing techniques.

Measurement and evaluation system has been developed with React framework 8.9.3 System works based on web. Mongo-Db Server 2021 has been used on database part. On database design, four roles have been defined: administrator, teacher, student and managerial

personals. Each of roles has different authorization for different E-Learning pages. To use E-Learning Testing and Evaluation System efficiently, a walkthrough guide can be prepared

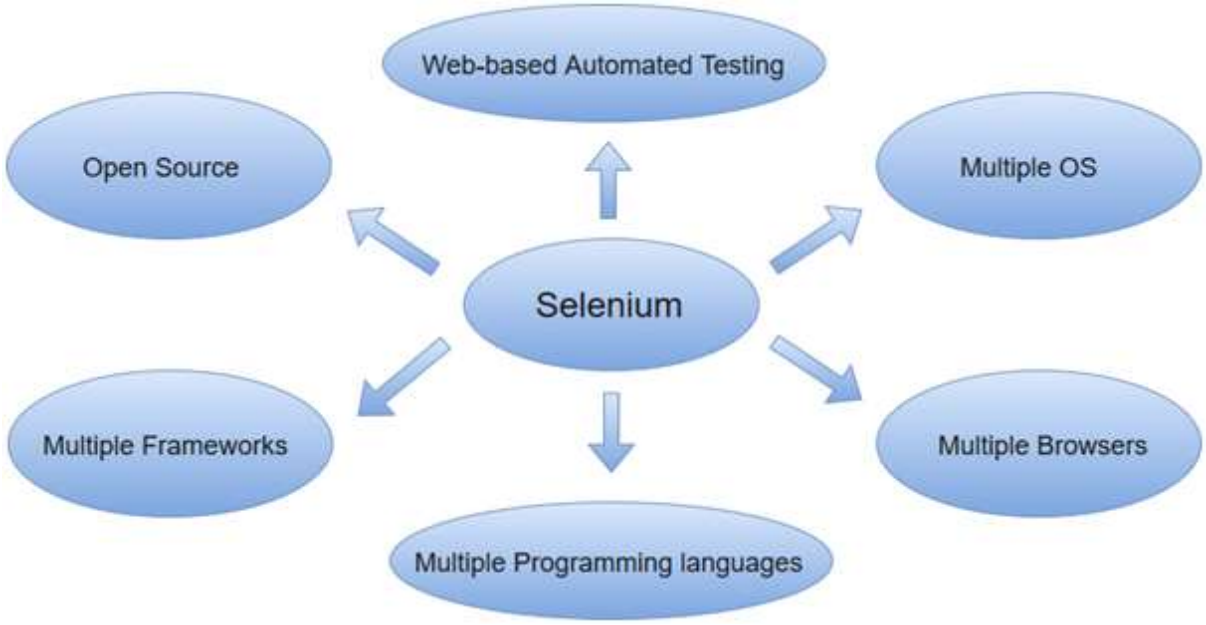
5.1 Software used for testing of application:

Selenium

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.

Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.



Keywords used in selenium during testing of application

Action - these are commands that change the state of the application

These category consist following commands

- Click
- Type
- Edit

Assertion: these are commands that are used for verification purpose. these category consist following commands

- verifytext {verification of text}
- asserttitle {show your page title}

Assessors: these are commands that store and retrieve information from DB

- store {store your value}
- storedVars {retrieve stores variable}

Synchronization commands: these commands that are used to provide an extra time in execution

these category consist of following commands

- waitforpagetoload {wait till object found}
- pause {wait for particular time}

Snapshot of running Selenium I D E

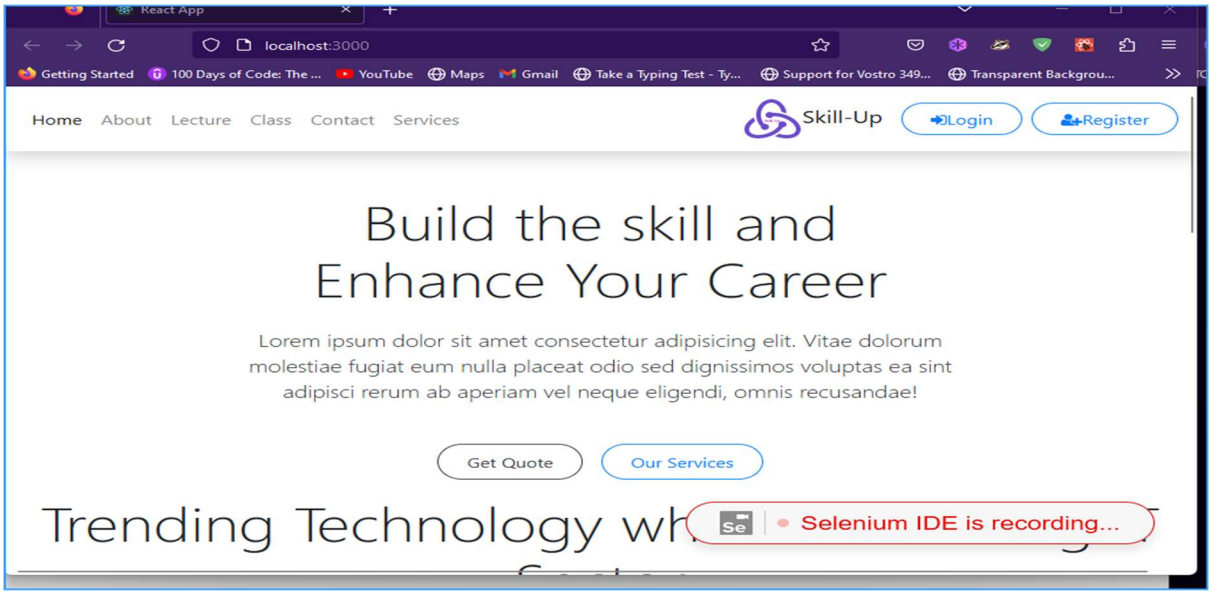


Fig 3 Selenium IDE started

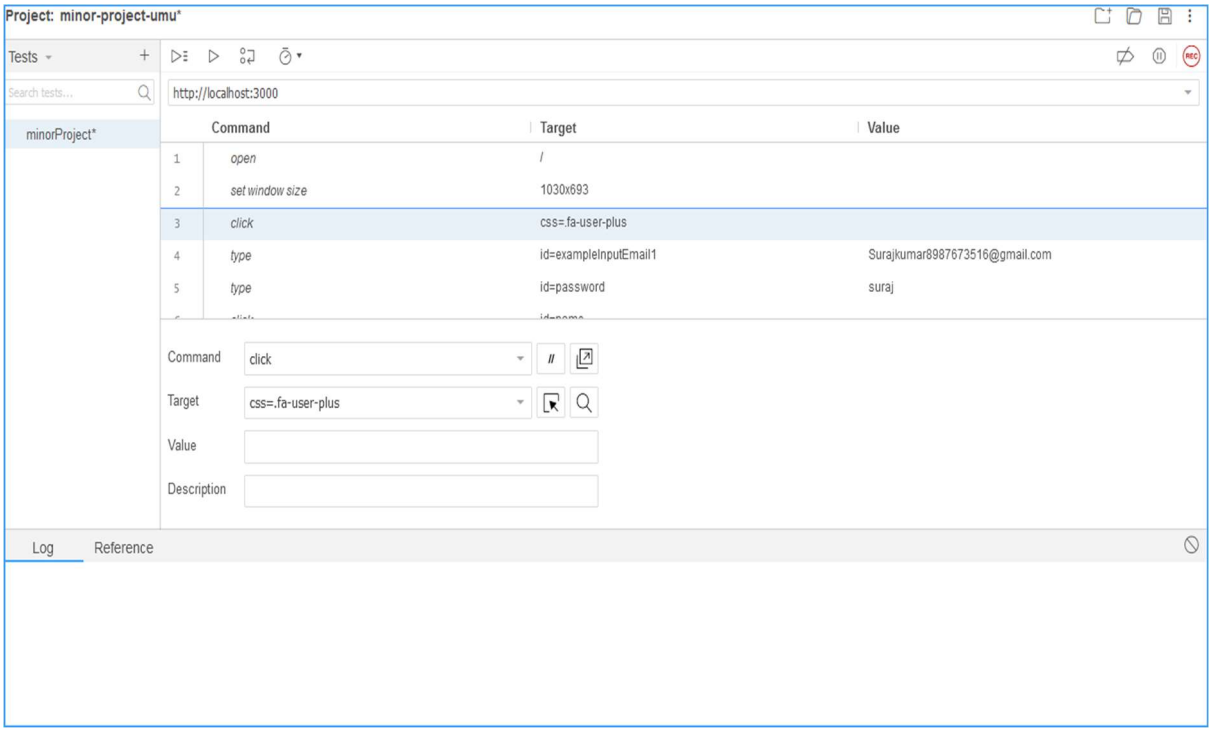


Fig 4 Operational Screen of Selenium IDE

With "Web Security Test" menu, we can determine E-Learning web pages security vulnerabilities via Vega Subgraph. Vega Subgraph is an Eclipse based test tool which helps to finding security deficiencies (Subgraph Vega, 2012). Both wanted URL addresses can be entered singly and operations which are done Vega Proxy can be tested simultaneously. Taken alerts classified as "Low", "Intermediate" and "High". Two "Low" level security deficiencies have been found while Vega Subgraph tests operations. The result which related to this security deficiency is seen Figure 6.

Unmasked page links have brought "Low" level security deficiency. Url rewrite methods have been applied to web pages to recover this deficiency.

SqlQueryStress tool has been used for E-Learning system database queries. By increasing number of query and number of threads, database answer time and average answer time have been determined (SQLQueryStress, 2012).

Code Used for testing and generated report code

```
// Generated by Selenium IDE
const { Builder, By, Key, until } = require('selenium-webdriver')
const assert = require('assert')

describe('minorProject', function() {
  this.timeout(30000)
  let driver
  let vars
  beforeEach(async function() {
    driver = await new
Builder().forBrowser('firefox').usingServer('http://localhost:4444/wd/hub').build()
    vars = {}
  })
  afterEach(async function() {
    await driver.quit();
  })
  it('minorProject', async function() {
    // Test name: minorProject
    // Step # | name | target | value
    // 1 | open | / |
    await driver.get("http://localhost:3000/")
    // 2 | setWindowSize | 1030x693 |
    await driver.manage().window().setRect({ width: 1030, height: 693 })
    // 3 | click | css=.fa-user-plus |
    await driver.findElement(By.css(".fa-user-plus")).click()
    // 4 | type | id=exampleInputEmail1 | Surajkumar8987673516@gmail.com
```

```
    await
driver.findElement(By.id("exampleInputEmail1")).sendKeys("Surajkumar898767351
6@gmail.com")
    // 5 | type | id=password | suraj
    await driver.findElement(By.id("password")).sendKeys("suraj")
    // 6 | click | id=name |
    await driver.findElement(By.id("name")).click()
    // 7 | type | id=name | Suraj kumarSingh
    await driver.findElement(By.id("name")).sendKeys("Suraj kumarSingh")
    // 8 | mouseDownAt | id=exampleInputEmail1 | 11,21
    {
      const element = await driver.findElement(By.id("exampleInputEmail1"))
      await driver.actions({ bridge: true
}).moveToElement(element).clickAndHold().perform()
    }
    // 9 | mouseMoveAt | id=exampleInputEmail1 | 11,21
    {
      const element = await driver.findElement(By.id("exampleInputEmail1"))
      await driver.actions({ bridge: true }).moveToElement(element).perform()
    }
    // 10 | mouseUpAt | id=exampleInputEmail1 | 11,21
    {
      const element = await driver.findElement(By.id("exampleInputEmail1"))
      await driver.actions({ bridge: true
}).moveToElement(element).release().perform()
    }
    // 11 | click | id=exampleInputEmail1 |
```

```
    await driver.findElement(By.id("exampleInputEmail1")).click()
    // 12 | type | id=exampleInputEmail1 | suraj31kumar1999
    await
driver.findElement(By.id("exampleInputEmail1")).sendKeys("suraj31kumar1999")
    // 13 | click | css=.col-md-6 |
    await driver.findElement(By.css(".col-md-6")).click()
    // 14 | click | id=profession |
    await driver.findElement(By.id("profession")).click()
    // 15 | type | id=profession | student
    await driver.findElement(By.id("profession")).sendKeys("student")
    // 16 | click | id=phone |
    await driver.findElement(By.id("phone")).click()
    // 17 | type | id=phone | 9876543210
    await driver.findElement(By.id("phone")).sendKeys("9876543210")
    // 18 | click | id=password |
    await driver.findElement(By.id("password")).click()
    // 19 | type | id=password | 123
    await driver.findElement(By.id("password")).sendKeys("123")
    // 20 | click | id=cpassword |
    await driver.findElement(By.id("cpassword")).click()
    // 21 | type | id=cpassword | 123
    await driver.findElement(By.id("cpassword")).sendKeys("123")
    // 22 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 23 | assertAlert | Register Successfully... Redirecting to Login.. |
```

```
    assert(await driver.switchTo().alert().getText() == "Register Successfully... Redirecting to Login..")

    // 24 | type | id=email | Surajkumar8987673516@gmail.com

    await driver.findElement(By.id("email")).sendKeys("Surajkumar8987673516@gmail.com")

    // 25 | type | id=password | suraj

    await driver.findElement(By.id("password")).sendKeys("suraj")

    // 26 | click | id=email |

    await driver.findElement(By.id("email")).click()

    // 27 | type | id=email | suraj31kumar1999@gmail.com

    await driver.findElement(By.id("email")).sendKeys("suraj31kumar1999@gmail.com")

    // 28 | click | id=password |

    await driver.findElement(By.id("password")).click()

    // 29 | type | id=password | 123

    await driver.findElement(By.id("password")).sendKeys("123")

    // 30 | click | css=.btn-primary |

    await driver.findElement(By.css(".btn-primary")).click()

    // 31 | assertAlert | Password is Wrong |

    assert(await driver.switchTo().alert().getText() == "Password is Wrong")

    // 32 | click | id=password |

    await driver.findElement(By.id("password")).click()

    // 33 | type | id=password | 1234

    await driver.findElement(By.id("password")).sendKeys("1234")

    // 34 | click | css=.btn-primary |

    await driver.findElement(By.css(".btn-primary")).click()

    // 35 | assertAlert | Password is Wrong |
```

```
    assert(await driver.switchTo().alert().getText() == "Password is Wrong")
    // 36 | click | id=email |
    await driver.findElement(By.id("email")).click()
    // 37 | type | id=password | suraj
    await driver.findElement(By.id("password")).sendKeys("suraj")
    // 38 | type | id=email | Surajkumar8987673516@gmail.com
    await
driver.findElement(By.id("email")).sendKeys("Surajkumar8987673516@gmail.com")
    // 39 | click | id=password |
    await driver.findElement(By.id("password")).click()
    // 40 | type | id=password | 8988
    await driver.findElement(By.id("password")).sendKeys("8988")
    // 41 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 42 | assertAlert | Password is Wrong |
    assert(await driver.switchTo().alert().getText() == "Password is Wrong")
    // 43 | click | css=form |
    await driver.findElement(By.css("form")).click()
    // 44 | click | id=password |
    await driver.findElement(By.id("password")).click()
    // 45 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 46 | assertAlert | Password is Wrong |
    assert(await driver.switchTo().alert().getText() == "Password is Wrong")
    // 47 | type | id=email | Surajkumar8987673516@gmail.com
```

```
    await
driver.findElement(By.id("email")).sendKeys("Surajkumar8987673516@gmail.com")
    // 48 | type | id=password | suraj
    await driver.findElement(By.id("password")).sendKeys("suraj")
    // 49 | click | id=email |
    await driver.findElement(By.id("email")).click()
    // 50 | click | id=password |
    await driver.findElement(By.id("password")).click()
    // 51 | type | id=password | 8988
    await driver.findElement(By.id("password")).sendKeys("8988")
    // 52 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 53 | assertAlert | Password is Wrong |
    assert(await driver.switchTo().alert().getText() == "Password is Wrong")
    // 54 | click | id=password |
    await driver.findElement(By.id("password")).click()
    // 55 | type | id=password | 123
    await driver.findElement(By.id("password")).sendKeys("123")
    // 56 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 57 | assertAlert | Password is Wrong |
    assert(await driver.switchTo().alert().getText() == "Password is Wrong")
    // 58 | click | id=email |
    await driver.findElement(By.id("email")).click()
    // 59 | type | id=email | suraj31kumar1999@gmail.com
```

```
    await
driver.findElement(By.id("email")).sendKeys("suraj31kumar1999@gmail.com")
    // 60 | click | id=password |
    await driver.findElement(By.id("password")).click()
    // 61 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 62 | assertAlert | Password is Wrong |
    assert(await driver.switchTo().alert().getText() == "Password is Wrong")
    // 63 | click | id=email |
    await driver.findElement(By.id("email")).click()
    // 64 | type | id=email | suraj@gmail.com
    await driver.findElement(By.id("email")).sendKeys("suraj@gmail.com")
    // 65 | click | id=password |
    await driver.findElement(By.id("password")).click()
    // 66 | type | id=password | 8988
    await driver.findElement(By.id("password")).sendKeys("8988")
    // 67 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 68 | assertAlert | User Login Successful... |
    assert(await driver.switchTo().alert().getText() == "User Login Successful...")
    // 69 | click | linkText=Lecture |
    await driver.findElement(By.linkText("Lecture")).click()
    // 70 | click | linkText=Class |
    await driver.findElement(By.linkText("Class")).click()
    // 71 | click | linkText=Wanna JOIN |
    await driver.findElement(By.linkText("Wanna JOIN")).click()
```

```
    // 72 | click | linkText=Class |
    await driver.findElement(By.linkText("Class")).click()
    // 73 | click | linkText=Wanna Create |
    await driver.findElement(By.linkText("Wanna Create")).click()
    // 74 | click | css=.form-control |
    await driver.findElement(By.css(".form-control")).click()
    // 75 | type | css=.form-control | 1
    await driver.findElement(By.css(".form-control")).sendKeys("1")
    // 76 | click | css=.btn-secondary |
    await driver.findElement(By.css(".btn-secondary")).click()
    // 77 | click | css=.VsTVUAD89KWleD0YRVsD |
    await driver.findElement(By.css(".VsTVUAD89KWleD0YRVsD")).click()
    // 78 | click | id=inputGroup-sizing-default |
    await driver.findElement(By.id("inputGroup-sizing-default")).click()
    // 79 | type | id=inputGroup-sizing-default |
C:\fakepath\052615_0721_Howtocreate1.png
    await driver.findElement(By.id("inputGroup-sizing-
default")).sendKeys("C:\\fakepath\\052615_0721_Howtocreate1.png")
    // 80 | click | id=inputGroupFileAddon04 |
    await driver.findElement(By.id("inputGroupFileAddon04")).click()
    // 81 | click | linkText=Services |
    await driver.findElement(By.linkText("Services")).click()
    // 82 | click | linkText=About |
    await driver.findElement(By.linkText("About")).click()
    // 83 | click | css=.ml-3 |
    await driver.findElement(By.css(".ml-3")).click()
```

```
    // 84 | click | css=.btn:nth-child(5) |
    await driver.findElement(By.css(".btn:nth-child(5)")).click()
    // 85 | click | linkText=Lecture |
    await driver.findElement(By.linkText("Lecture")).click()
    // 86 | click | linkText=Login |
    await driver.findElement(By.linkText("Login")).click()
    // 87 | type | id=email | Surajkumar8987673516@gmail.com
    await
driver.findElement(By.id("email")).sendKeys("Surajkumar8987673516@gmail.com")
    // 88 | type | id=password | suraj
    await driver.findElement(By.id("password")).sendKeys("suraj")
    // 89 | click | css=.btn-outline-dark |
    await driver.findElement(By.css(".btn-outline-dark")).click()
    // 90 | type | id=exampleInputEmail1 | Surajkumar8987673516@gmail.com
    await
driver.findElement(By.id("exampleInputEmail1")).sendKeys("Surajkumar898767351
6@gmail.com")
    // 91 | type | id=password | suraj
    await driver.findElement(By.id("password")).sendKeys("suraj")
    // 92 | click | css=.btn-primary |
    await driver.findElement(By.css(".btn-primary")).click()
    // 93 | assertAlert | Error Found |
    assert(await driver.switchTo().alert().getText() == "Error Found")
    // 94 | click | css=.ml-5 |
    await driver.findElement(By.css(".ml-5")).click()
  })
})
```

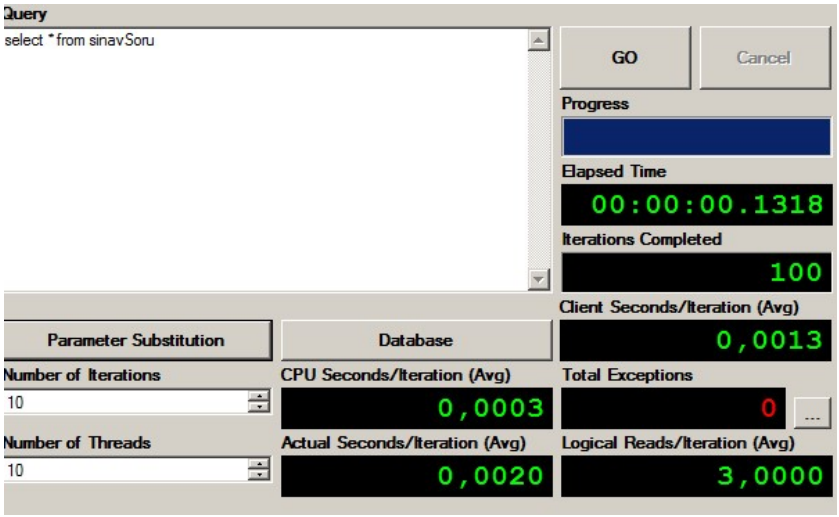
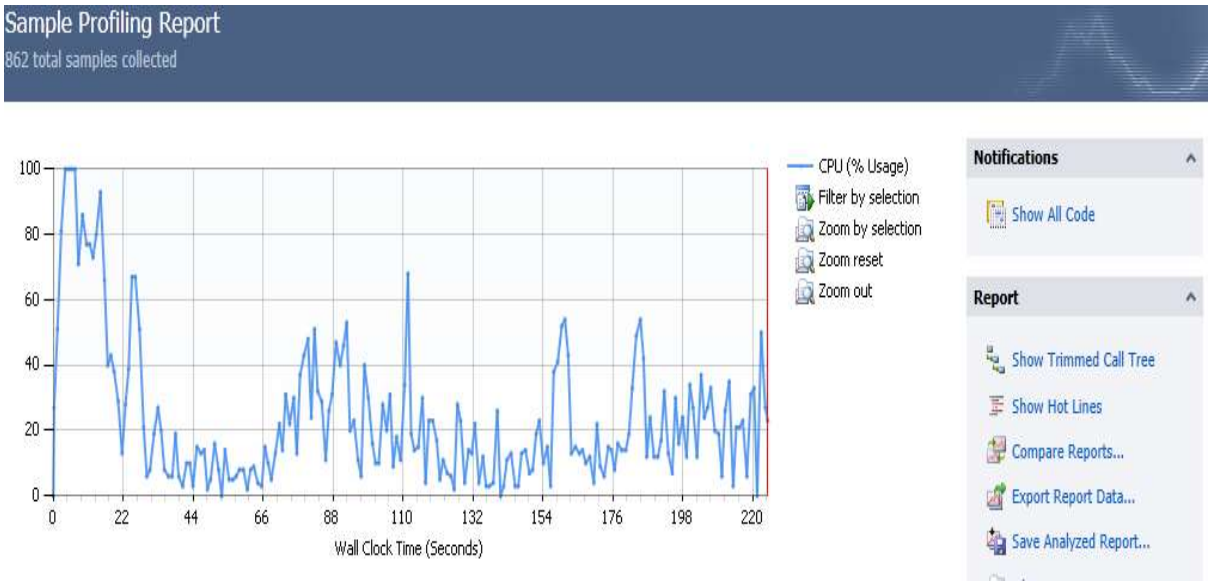


Figure 7: Vega Subgraph security test result screen

With CPU usage test, we can obtain CPU usage rate while E-Learning system operations. CPU usage test measured with Performance Analyzer which exits in Visual Studio Ultimate and Premium distributions. Performance test results which have been done for measurement and evaluation system have been explained as following. Blue color lines which seen at Figure 8 symbolized used class and CPU usage percent functions while Project is working. Working result is seen for 862 (class, function) examples. When the system works firstly, CPU usage percent reaches out to %100 then shows variability according to used function and classes



Chapter 6

SNAPSHOT OF THE RUNNING MODEL

Fig 13: Home Page

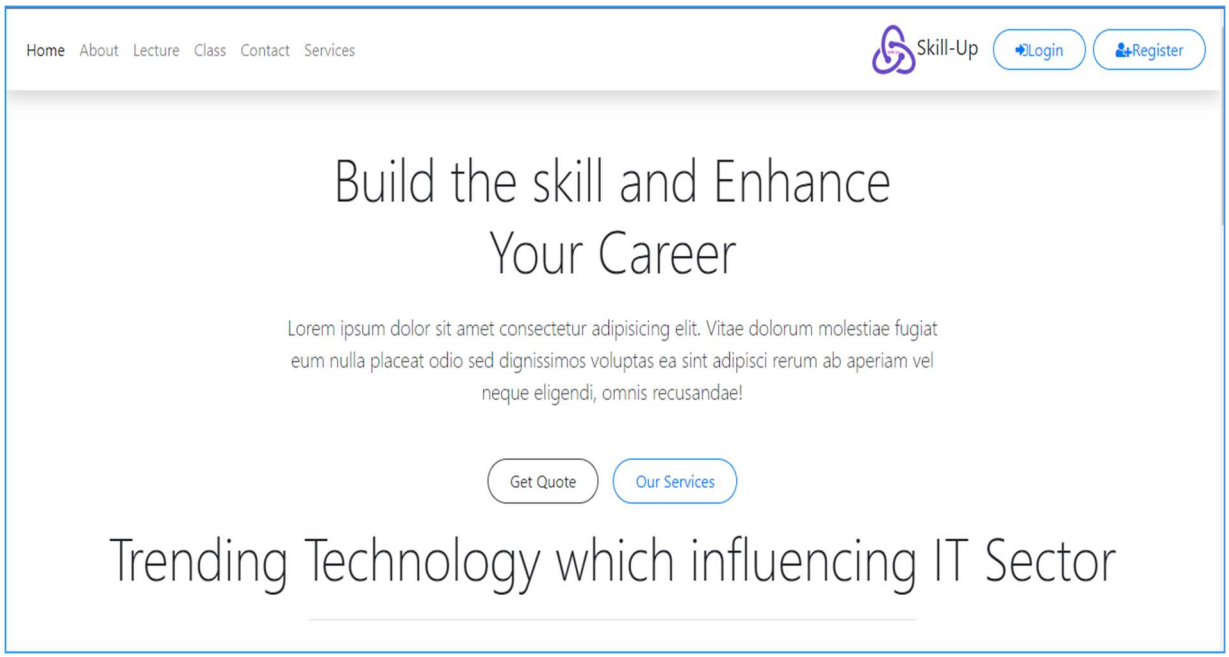


Fig 14: Technologies Page

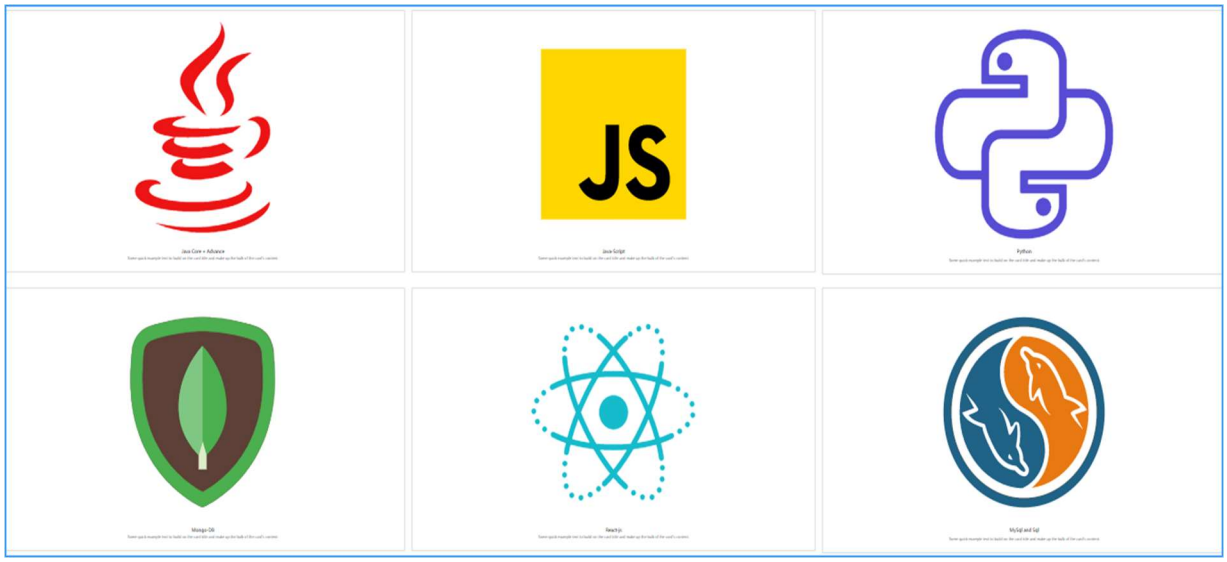


Fig 15: Footer page

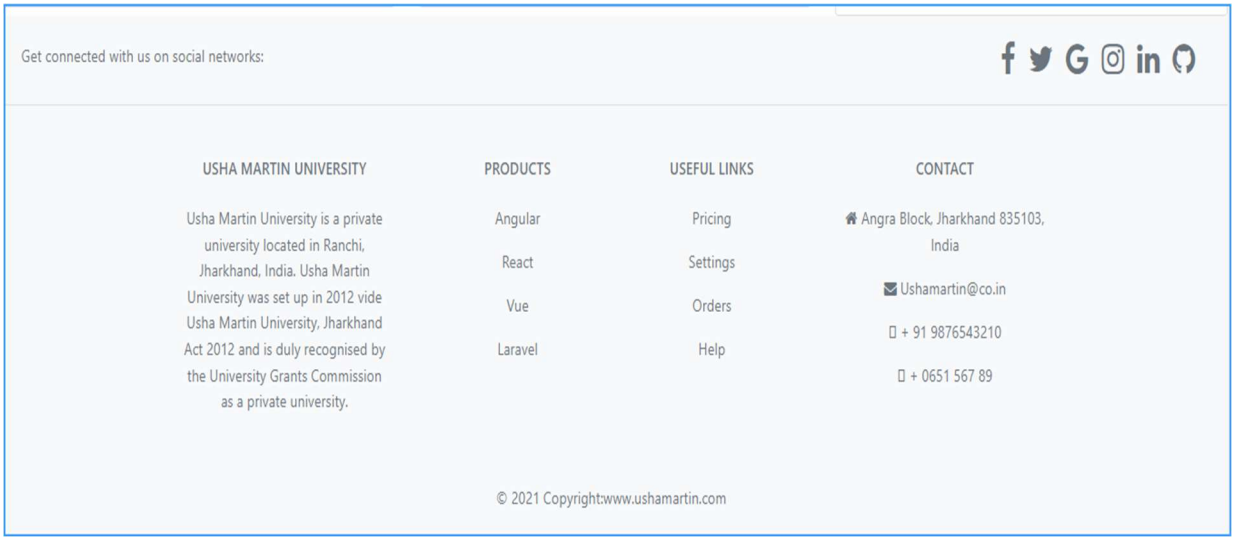


Fig 16: About Page

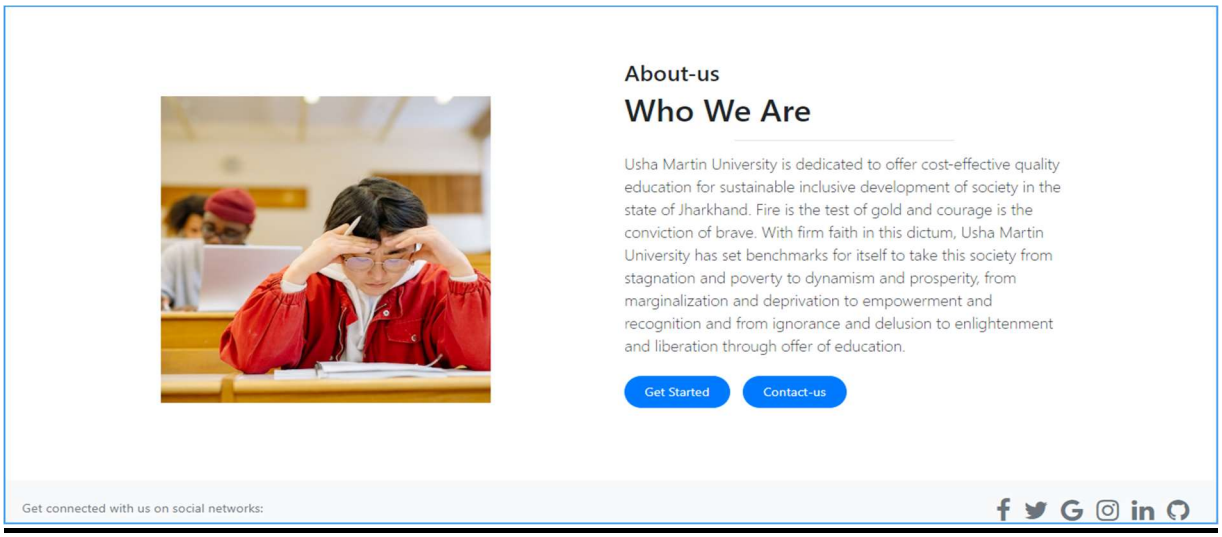


Fig 17: Class Joining Page

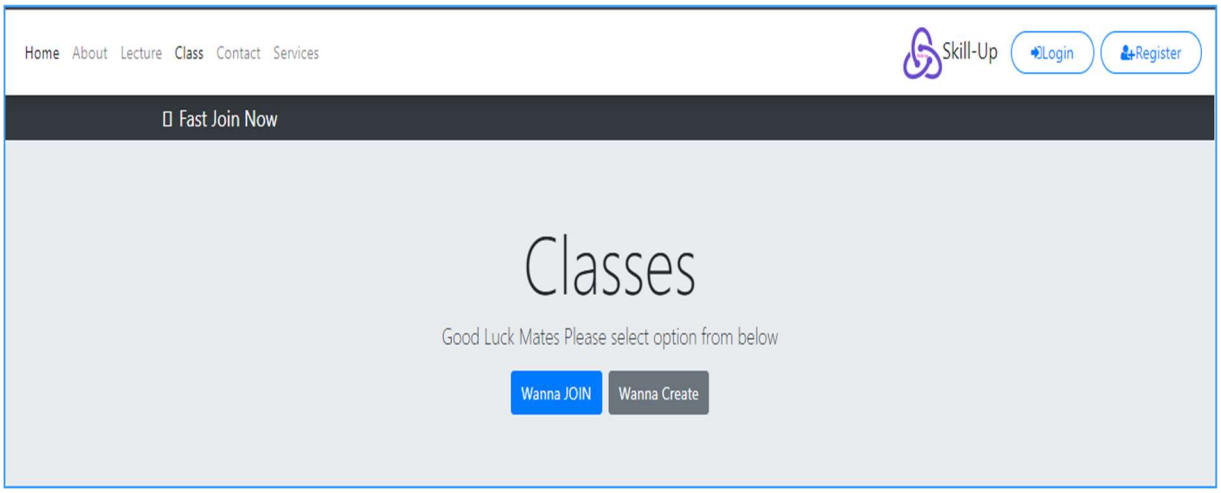


Fig 18: Contact Us Page

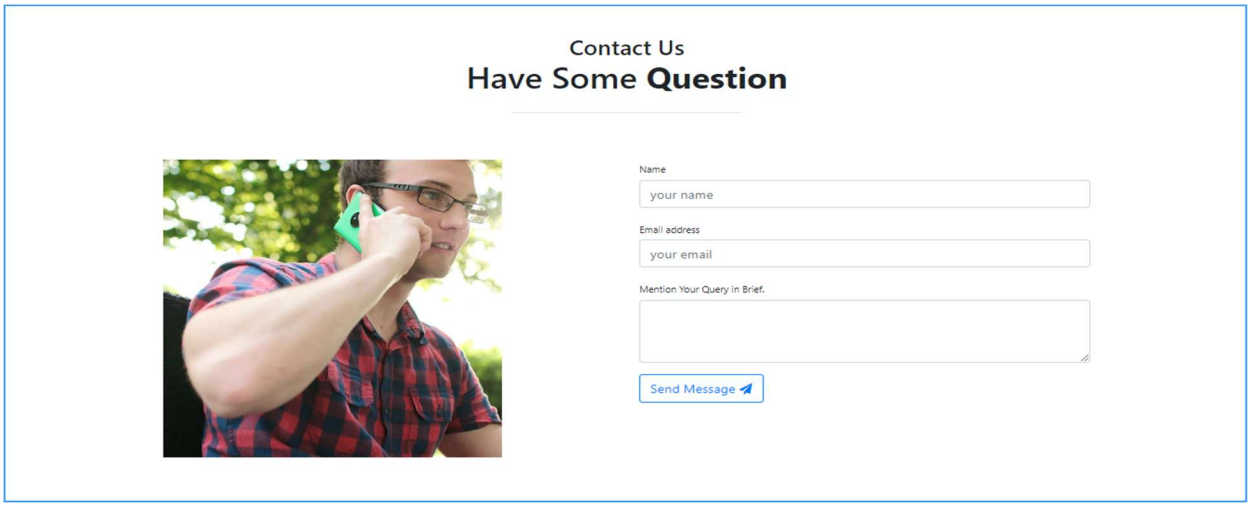


Fig 19: Services Page

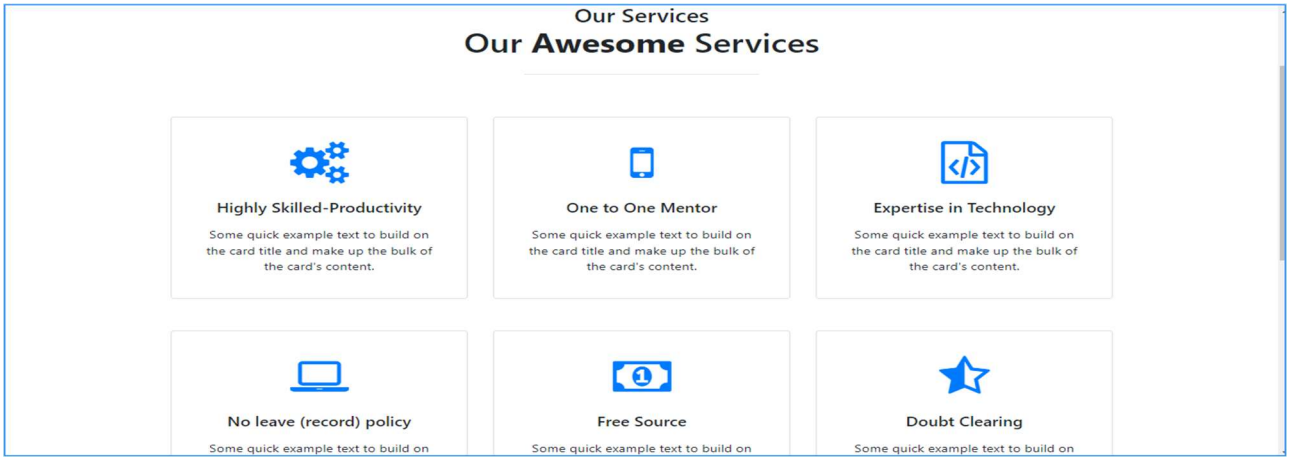


Fig 19: Register Page

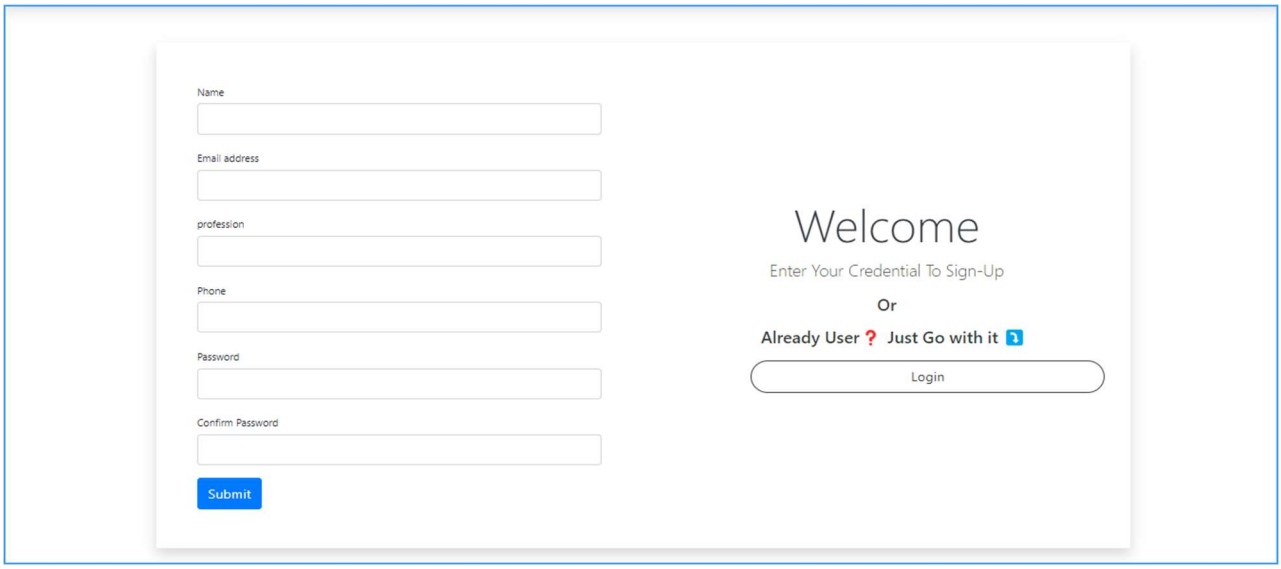


Fig 21: Login Page

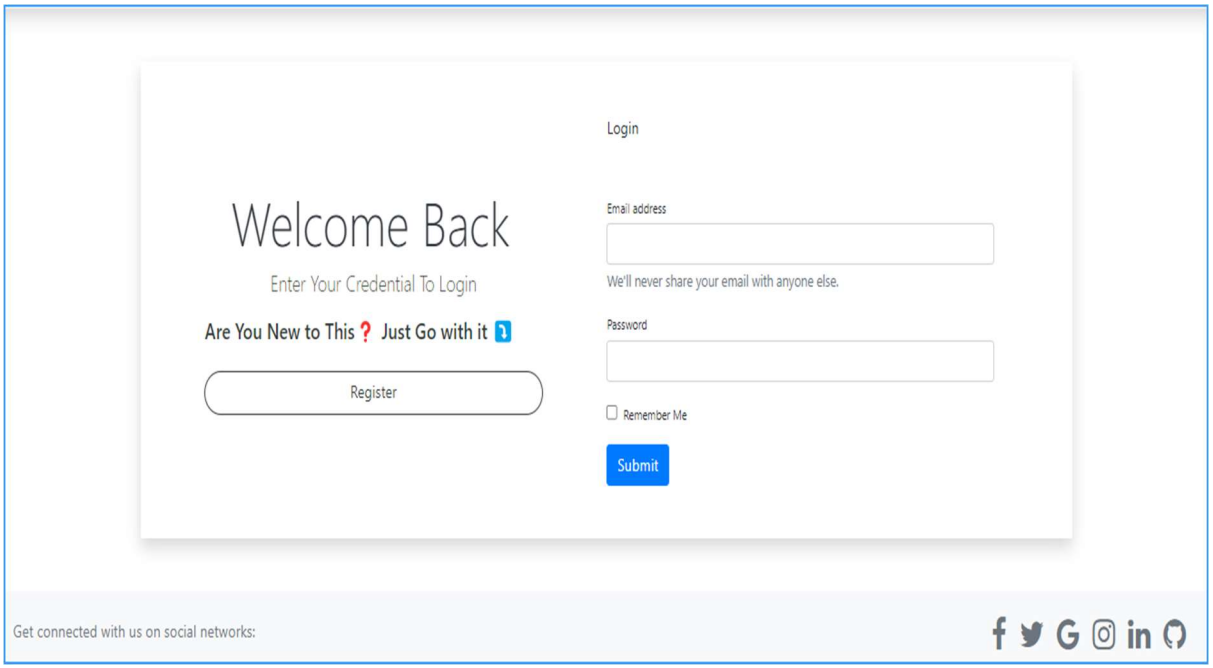


Fig 21: Create Room Page

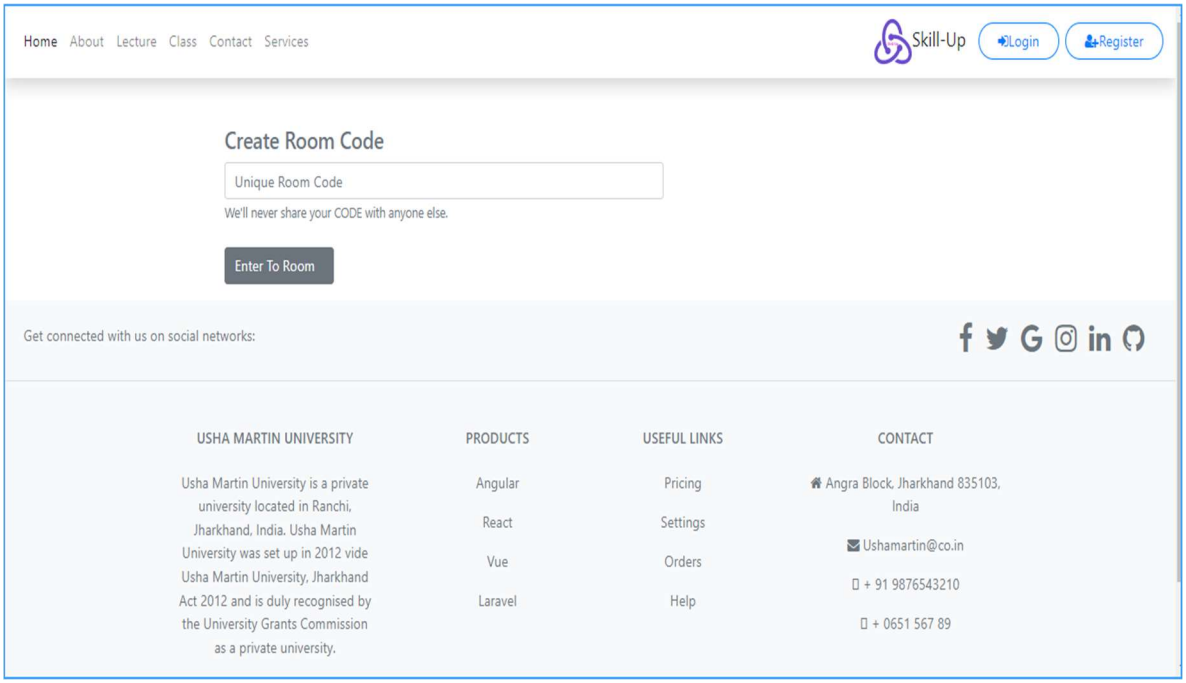


Fig 21:Room Joining Page

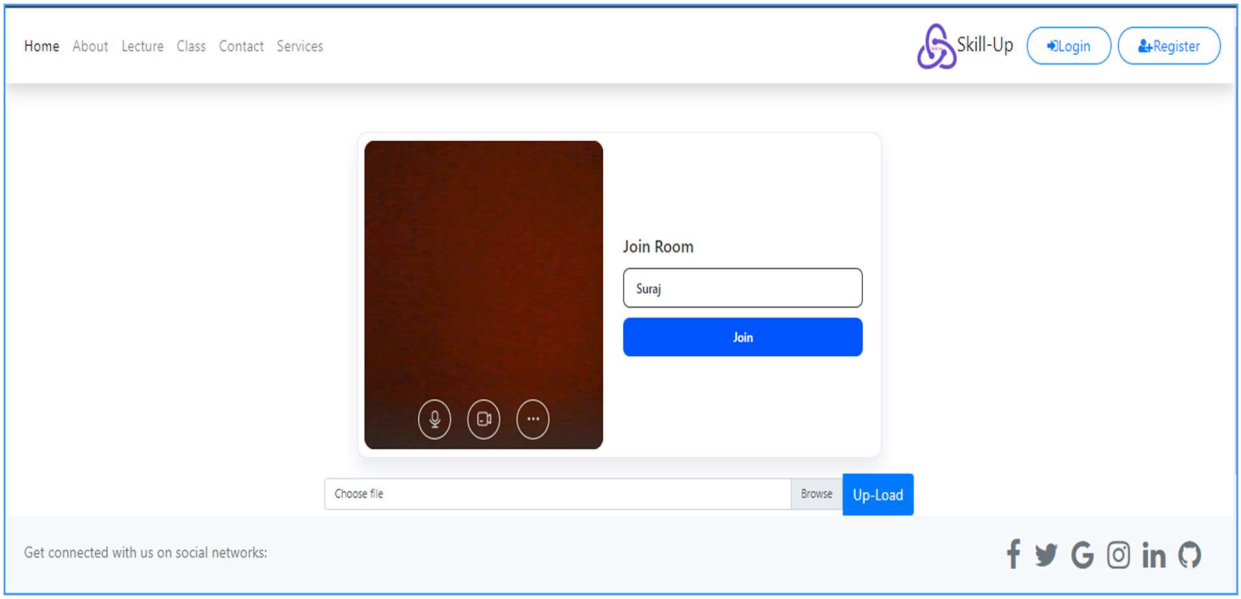
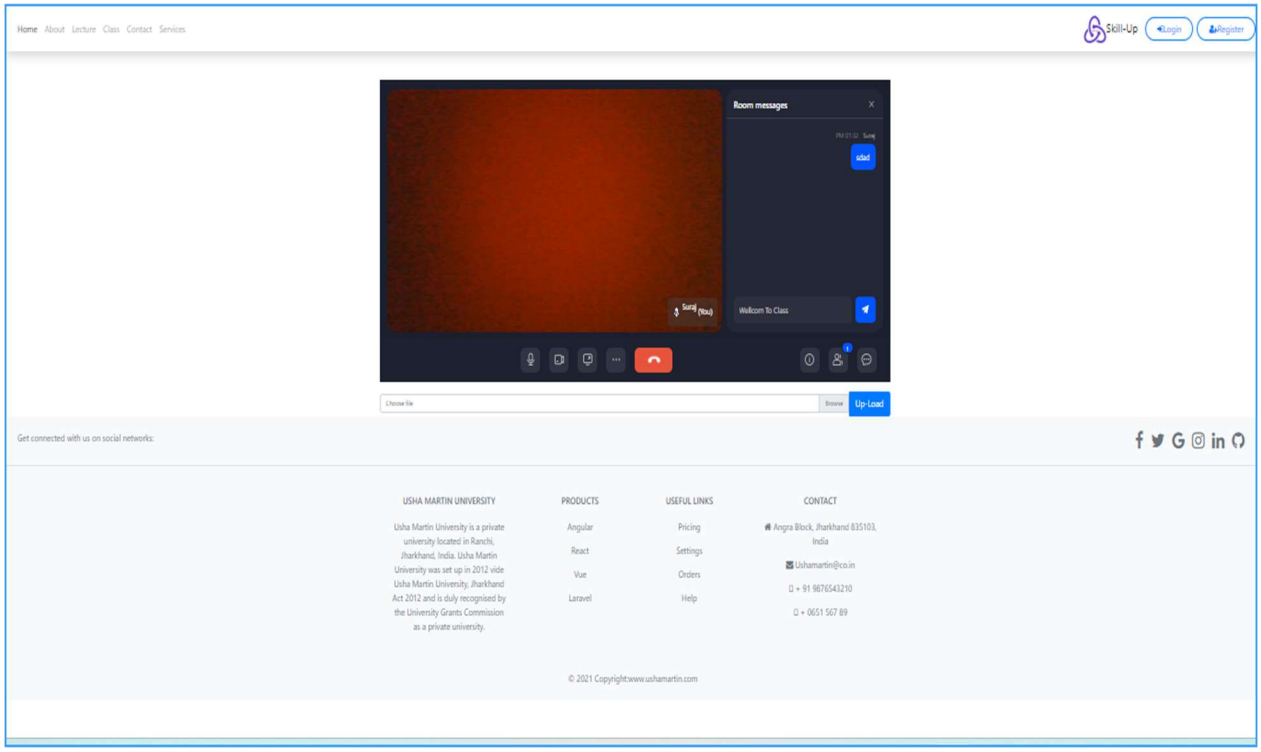


Fig 21: Class Lobby



Chapter 7

Limitation, Conclusion and further Remark

7.1 Limitation Of E-learning platform

7.2 Conclusion

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses. At the end it is concluded that we have made effort on following points...

A description of the background and context of the project and its relation to work already done in the area.

 Made statement of the aims and objectives of the project.

The description of Purpose, Scope, and applicability.

We &fine the problem on which we are working in the project.

We describe the requirement Specifications of the system and the actions that can be done on these things.

We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system. We included features and operations in detail, including screen layouts. We designed user interface and security issues related to system. Finally the system is implemented and tested according to test cases.

7.3 Further Scope

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding: We can add printer in future. We can give more advance software for E-learning Management System including more facilities We will host the platform on online servers to make it accessible worldwide Integrate multiple load balancers to distribute the loads of the system Create the master and slave database structure to reduce the overload of the database queries Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Assignment and Student. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the E-learning Management System. Enhancements can be done to maintain all the Assignment, Student, TEACHER, QUIZ, QUESTION. We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. ln the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process.

Chapter 8

SOURCE CODE

Server Part

``````````````````````

# Dependencies of Server


```
{
 "name": "MERN_project",
 "version": "1.0.0",
 "description": "",
 "main": "index.js",
 "scripts": {
   "test": "echo \"Error: no test specified\" && exit 1",
   "start": "nodemon app.js"
 },
 "keywords": [],
 "author": "",
 "license": "ISC",
 "dependencies": {
   "bcryptjs": "^2.4.3",
   "cookie-parser": "^1.4.5",
   "dotenv": "^16.0.3",
   "express": "^4.18.2",
   "jsonwebtoken": "^9.0.0",
   "mongoose": "^6.8.1"
 },
 "watch": [
```
```

```
    "src/"
  ],
  "ignore": [
    "dist/",
    "node_modules",
    ".git"
  ],
  "ext": "js,json",
  "env": {
    "NODE_ENV": "development"
  },
  "execMap": {
    "js": "node"
  },
  "devDependencies": {
    "nodemon": "^2.0.20"
  }
}
```

## ***Main Server***

```
const express = require('express')
const bodyParser = require('body-parser')
const mongoose = require('mongoose')
const cors = require('cors')
const multer =require('multer')
// const {Server} = require('socket.io')
```

```
// const articleRouter = require('./Controler/fileControler')
const userController = require('./Controler/userControler')
const fileControler = require('./Controler/fileControler')
const port = 8000

mongoose.set("strictQuery", false);
mongoose.connect('mongodb://127.0.0.1:27017/myapp', async (err)=>{
  if(!err){
    await console.log("DATABASE CONNECTED SUCCESSFULLY...");
  }else{
    console.log("OOPS SOMETHING WENT WRONG ==>  ",err)
  }
});

//io for video server
// const io = new Server({cors:true})
const app = express()
app.use(cors())


//parse application /x-www-font-urlencoded
app.use(bodyParser.urlencoded({extended:false}))

//parse application json
app.use(bodyParser.json())
```

```
// Routing part
app.post('/register', userController.signup)
app.post('/signin', userController.signin)
app.get('/vroom/allpost', fileControler.allPost)
app.post('/vroom/addpost', fileControler.addPost)
app.get('/vroom/find/:id', fileControler.findArticleId)
app.put('/vroom/update/:id',fileControler.findArticleIdAndUpdate)
app.delete('/vroom/delete/:id',fileControler.findArticleIdAndDelete)
// app.use('/vroom', articleRouter)
// ////////////////////////////////////////////////////////////////////////////
// const emailToSocketMapping = new Map()
// const socketTOEmailMapping = new Map()

// io.on('connection', (socket)=>{
//   console.log("New Connection Joined");
//   socket.on('join-room', (data) =>{
//     const {roomId, emailId} = data;
//     console.log("User", emailId, "Joined Room", roomId)
//     emailToSocketMapping.set(emailId, socket.id)
//     socketTOEmailMapping.set(socket.id,emailId)
//     socket.join(roomId)
//     socket.emit('Joined-Room', {roomId})
//     socket.broadcast.to(roomId).emit('user-joined', {emailId})
```

```
//   })
//   socket.on("call-user", (data)=>{
//     const{emailId, offer} = data;
//     const fromEmail = socketTOEmailMapping.get(socket.id)
//     const socketId = emailToSocketMapping.get(emailId)
//     socket.to(socketId).emit("Incomming call",{from: fromEmail, offer })
//   })
//   socket.on('call-accepted', (data)=>{
//     const {emailId, ans} = data
//     const socketId = emailToSocketMapping.get(emailId)
//     socket.to(socketId).emit('call-accepted',{ans})
//   })
// })




// ////////////////////////////////////////////////////////////////////////////
app.listen(port, () => {console.log(`Server App Running... on   P O R T = ${port}`)})
// io.listen(8001, ()=>{console.log("Video Server Strated in P O R T =  8001");})
```

Model of Project

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema
```

```
const articleSchema = new Schema({
    title: {
        type: String,
        required: true
    },
    description: {
        type: String,
        required: true
    },
    img: {
        type: String,
        required: true
    },
    author: {
        type: String,
        required: true
    },
    postDate: { type: Date, default: Date.now }
})
const Articles = mongoose.model('Article', articleSchema)


module.exports = Articles;
```

***user model***

```
    const mongoose = require('mongoose')
    module.exports = mongoose.model('User', {
```

```
    name: {
        type: String,
        required: true
    },
    email: {
        type: String,
        required: true
    },
    profession: {
        type: String,
        required: true
    },
    phone: {
        type: String,
        required: true
    },
    password: {
        type: String,
        required: true
    },
    cpassword: {
        type: String,
        required: true
    }
})
```

***Controller Part***

```
const Articles = require('../Model/article')


//REQUEST Get all Post
module.exports.allPost = (req, res)=>{
    Articles.find().then(article => res.json(article))
    .catch(err => res.status(400).json(`Error found => ${err} `))
    console.log(req.body)
}
// REQUEST ADD NEW ARTICLE
module.exports.addPost = (req,res)=>{
    const newArticle = new Articles({
        title:req.body.title,
        description:req.body.description,
        img:req.body.img,
        author:req.body.author
    })
    newArticle.save()
    .then(()=>res.json("The new Article Added SUCCESS"))
    .catch(err=>res.status(400).json(`ERROR:=> ${err}`))
}
// REQUEST FIND ARTICLE BY ID
module.exports.findArticleId = (res,req)=>{
```

```
    Articles.findById(req.params.id)
    .then(article=>res.json(article))
    .catch(err=>res.status(400).json(`ERROR:=> ${err}`))
}
// FIND ARTICLE BY ID AND UPDATE
module.exports.findArticleIdAndUpdate=(req,res)=>{
    Articles.findById(req.params.id)
    .then(article=>{
        article.title = req.body.title
        article.description = req.body.description
        article.img = req.body.img
        article.author=req.body.author
        article.save()
        .then(()=>res.json('The Article is Updated Successfully'))
        .catch(err=>res.status(400).json(`ERROR:=> ${err}`))
    })
    .catch(err=>res.status(400).json(`ERROR:=> ${err}`))
}

// REQUEST FIND ARTICLE BY ID AND DELETE
module.exports.findArticleIdAndDelete=(req,res)=>{
    Articles.findByIdAndDelete(req.params.id)
    .then(()=>res.json("The article is Deleted"))
    .catch(err=>res.status(400).json(`ERROR:=> ${err}`))
```

```
}
```

***User Controller***

```
const userModal = require('../Model/userModel')


module.exports.signup = (req, res) => {
    // creating instance of userModal
    const newUser = new userModal({
        name: req.body.name,
        email: req.body.email,
        profession: req.body.profession,
        phone: req.body.phone,
        password: req.body.password,
        cpassword: req.body.cpassword,
    })




    newUser.save().then(() => {
        res.send({ code: 200, message: "Sign-Up SUCCESS..." })
    }).catch((err) => {
        res.send({ code: 500, message: "Something went Wrong on server" })
        console.log(err)
    })
    console.log(req.body);
}
```

```
module.exports.signin = (req, res) => {


   // matching email.and password
   userModal.findOne({ email: req.body.email }).then(result => {
      console.log(result);
      // matching password
      if (result.password !== req.body.password) {
         res.send({ code: 404, message: "opps!! Pass Not matched" })
      } else {
         res.send({ code: 200, message: "user Found", token: "qwertyiop" })
      }
   }).catch(err => { res.send({ code: 500, message: "User NOT found....." }) })


}
```

# ***Client Side***

- ***Classroom-part***

```
import React, { useState, useEffect, useCallback } from 'react'
import { useNavigate } from 'react-router-dom'
import { useSocket } from '../Providers/Socket'
const Classhome = () => {
   const Socket = useSocket();
   const navigate = useNavigate();
```

```
const [email, setEmail] = useState("")
const [roomId, setRoomId] = useState("")



const handleRoomJoined = useCallback(({ roomId }) => {
    console.log("Room joined => ", roomId);
    navigate(`/classroom/${roomId}`);
}, [navigate]);
useEffect(() => {
    Socket.on("Joined-Room", handleRoomJoined);
    return () => {
        Socket.off("Joined-Room", handleRoomJoined)
    }
}, [handleRoomJoined, Socket]);

const handleJoinRoom = () => {
    Socket.emit('join-room', { emailId: email, roomId });
    navigate(`/classroom/${roomId}`);
};




return (
    <>
```

```
        <div className="homepage-container">

          <div className='input-container'>

            <input value={email} onChange={e => setEmail(e.target.value)}
type="email" placeholder='Enter your email here' />

            <input value={roomId} onChange={e => setRoomId(e.target.value)}
type="text" placeholder='Enter Room Code' />

            <button onClick={handleJoinRoom}>Enter Room</button>

          </div>

        </div>

        {/* ////////////////////////////////////////////////////////////////////////// */}

        {/* <form  className="form mt-5 w-75 mt-5">

          <div className="form-group w-50 mx-auto  ">

            <h4 className="text-muted">Create Room Code</h4>

            <input type="email" className="form-control mb-2" value={email}
onChange={e => setEmail(e.target.value)} required placeholder="Enter Email" />

            <input type="text" className="form-control" value={roomId}
onChange={e => setRoomId(e.target.value)} required placeholder="Unique Room Code"
/>

            <small id="emailHelp" className="form-text text-muted">We'll never
share your CODE with anyone else.</small><br/>

            <button type="button" onClick={handleJoinedRoom} className=" btn btn-
secondary text-center d-flex  w-25 btn-block">Enter To Room</button>

          </div>

          <div className="">

          </div>

        </form> */}

        {/* ////////////////////////////////////////////////////////////////////////////// */}

      </>

    )
```

```
}


export default Classhome


……..

import React, { useEffect, useCallback, useState } from 'react'

import ReactPlayer from 'react-player'

import { useSocket } from '../Providers/Socket'

import { usePeer } from '../Providers/Peer';

const Classroom = () => {

  const socket = useSocket();

  const { peer, createOffer, createAnswer, setRemoteAns, sendStream, remoteStream } =
usePeer();

  const [myStream, setMyStream] = useState(null);

  const [remoteEmailId, setRemoteEmailId] = useState(null)


  const handleNewUserJoined = useCallback(async (data) => {

    const { emailId } = data

    console.log("New user joined", emailId)

    const offer = await createOffer();

    socket.emit('call-user', { emailId, offer })

    setRemoteEmailId(emailId)

  },

    [createOffer, socket]

  );
```

```
  const handleIncomingCall = useCallback(async (data) => {
    const { from, offer } = data
    console.log("Incoming Call From ", from, offer);
    const ans = await createAnswer(offer)
    socket.emit('call-accepted', { emailId: from, ans })
    setRemoteEmailId(from)
  }, [createAnswer, socket])

  const handleCallAccepted = useCallback(async (data) => {
    const { ans } = data
    console.log('Call Got Accepted', ans)
    await setRemoteAns(ans)

  }, [setRemoteAns])

  const getUserMediaStream = useCallback(async () => {
    const stream = await navigator.mediaDevices.getUserMedia({ audio: true, video: true
});
    setMyStream(stream)
  }, [])

  const handleNegosiation = useCallback(() => {
    const localOffer = peer.localDescription;
    socket.emit('call-user', { emailId: remoteEmailId, offer: localOffer })
  }, [peer.localDescription, remoteEmailId, socket])
```

```
useEffect(() => {
  socket.on('user-joined', handleNewUserJoined)
  socket.on('incoming-call', handleIncomingCall)
  socket.on('call-accepted', handleCallAccepted)
  return () => {
    socket.off('user-joined', handleIncomingCall)
    socket.off('incoming-call', handleIncomingCall)
    socket.off('call-accepted', handleCallAccepted)
  }
}, [handleCallAccepted, handleIncomingCall, handleNewUserJoined, socket])

useEffect(() => {
  peer.addEventListener('negotiationneeded', handleNegosiation)
  return () => {
    peer.removeEventListener('negotiationneeded', handleNegosiation)
  }
}, [])

useEffect(() => {
  getUserMediaStream();
}, [getUserMediaStream])

return (
  <>
    <div className='room-page-container'>
```

```
        <h1>Room Page</h1>
        <h4>You are connected to {remoteEmailId}</h4>
        <button onClick={e => sendStream(myStream)} >Send My Video</button>
        <ReactPlayer url={myStream} playing muted />
        <ReactPlayer url={remoteStream} playing />
      </div>
    </>
  )
}
export default Classroom
```

## ***Lobby of Student***

```
import React, { useState } from "react";
import { useNavigate } from "react-router-dom";

const VHomePage = () => {
    const navigate = useNavigate()
    const [roomCode, setRoomCode] = useState('')
    const handleFormSubmit = (ev) => {
        ev.preventDefault()
        navigate(`/vroom/${roomCode}`)
    }
    return (
        <>
            <form className="form mt-5 w-75 mt-5">
```

```
            <div className="form-group w-50 mx-auto  ">

                <h4 className="text-muted">Create Room Code</h4>

                <input type="text" className="form-control" value={roomCode}
onChange={e => setRoomCode(e.target.value)} required placeholder="Unique Room
Code" />

                <small id="emailHelp" className="form-text text-muted">We'll never
share your CODE with anyone else.</small><br />

                <button type="button" onClick={handleFormSubmit} className=" btn btn-
secondary text-center d-flex  w-25 btn-block">Enter To Room</button>

            </div>

            <div className="">

            </div>

          </form>

       </>

    )

}


export default VHomePage
```

…….

```
import React from 'react'

import { useParams } from 'react-router-dom'

import { ZegoUIKitPrebuilt } from '@zegocloud/zego-uikit-prebuilt';




const VRoomPage = () => {

    const { roomId } = useParams()
```

```
const myMeeting = async (element) => {
    const appID = 1154075355;
    const sereverSecret = '7b7b6395d8d0f1991f0fc97b4944cf0f';
    const kitToken = ZegoUIKitPrebuilt.generateKitTokenForTest(
        appID,
        sereverSecret,
        roomId,
        Date.now().toString(),
        "Suraj"
    )
    const zp = ZegoUIKitPrebuilt.create(kitToken);

    zp.joinRoom({
        container: element,
        scenario: {
            mode: ZegoUIKitPrebuilt.VideoConference
        }
    })
}
return (
    <>
        <div className="room-page mx-auto w-75 mt-5">
            <div ref={myMeeting} className="container" />
        </div>
```

```
        <div className="container w-50">
            <div className="input-group mb-3">
            </div>
            <div className="input-group">
                <div className="custom-file mt-1">
                    <input type="file" className="custom-file-input" id="inputGroup-sizing-default" aria-describedby="inputGroupFileAddon04" />
                    <label className="custom-file-label" for="inputGroupFile04">Choose file</label>
                </div>
                <div className="input-group-append">
                    <button className="btn btn-primary " type="button" id="inputGroupFileAddon04">Up-Load</button>
                </div>
            </div>
        </div>
      </>
   )
}

export default VRoomPage
import React from 'react'
import { useEffect } from 'react'
import pic1 from '../assets/about1.jpg'
function About() {

   const callAboutPage = async () => {
```

```
    try {
        await fetch('/about')
    } catch (error) {
        console.log(error)
    }
}
useEffect(() => {
    callAboutPage();
}, []);



return (
    <>

        <section id="about">
            <div className="container my-5 py-5">
                <div className="row">
                    <div className="col-md-6">
                        <img src={pic1} alt="img" className='w-75 me-5 mt-5' />
                    </div>
                    <div className="col-md-6">
                        <h3 className="fs-5">About-us</h3>
                        <h1 className="display-6 mb-2">Who <b /> We <b />Are </h1>
                        <hr className='w-50' />
                        <p className="lead mb-4">Lorem ipsum dolor sit amet, consectetur
```

```
                    adipisicing elit. Labore numquam maxime magnam consequatur
nesciunt
                    accusantium, necessitatibus similique amet blanditiis dicta atque
nemo debitis,
                    fugit quas eum fuga praesentium at non. Assumenda reprehenderit
iusto nulla!
                    Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum
dolor sit,
                    amet consectetur adipisicing elit. Provident placeat delectus illo quis
quas pariatur
                    voluptatibus, nostrum, accusamus laboriosam unde, quod distinctio
ducimus corrupti eum
                    veniam fuga hic doloribus neque.
                </p>
                <button className='btn btn-primary rounded-pill px-4 py-2'>Get
Started</button>
                <button className='btn btn-primary rounded-pill px-4 py-2 ms-2 ml-
3'>Contact-us</button>
              </div>
            </div>
          </div>
        </section>

    </>
  )
}

export default About
```

……..

```
import React from 'react'
import { Link } from 'react-router-dom'




function Class() {
  return (
    <>
      {/* ////////////////////////////////////////////////////// */}
      <nav className="bg-dark navbar-dark">
        <div className="container">
          <a href="/" className="navbar-brand"><i className="fas fa-tree mr-2"></i>Fast
Join Now</a>
        </div>
      </nav>
      <section id="header" className="jumbotron text-center">
        <h1 className="display-3">Classes</h1>
        <p className="lead">Good Luck Mates Please select option from below</p>
        <Link to="/vroom" className="btn btn-primary">Wanna JOIN</Link>
        <Link to="/vhome" className="btn btn-secondary ml-2">Wanna Create</Link>
      </section>
      <section id="gallery">
        <div className="container">
          <div className="row">
            <div className="col-lg-4 mb-4">
              <div className="card">
```

```
            <img src="https://images.unsplash.com/photo-1477862096227-3a1bb3b08330?ixlib=rb-1.2.1&auto=format&fit=crop&w=700&q=60" alt="" className="card-img-top" />

            <div className="card-body">

              <h5 className="card-title">Window Programing</h5>

              <p className="card-text">Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum similique repellat a laborum, rerum voluptates ipsam eos quo tempore iusto dolore modi dolorum in pariatur. Incidunt repellendus praesentium quae!</p>

              <a href="/" className="btn btn-outline-success btn-sm">Read More</a>

              <a href="/" className="btn btn-outline-danger btn-sm"><i className="far fa-heart"></i></a>

            </div>

          </div>

        </div>

        <div className="col-lg-4 mb-4">

          <div className="card">

            <img src="https://images.unsplash.com/photo-1516214104703-d870798883c5?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=700&q=60" alt="" className="card-img-top" />

            <div className="card-body">

              <h5 className="card-title">Complier Design</h5>

              <p className="card-text">Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum similique repellat a laborum, rerum voluptates ipsam eos quo tempore iusto dolore modi dolorum in pariatur. Incidunt repellendus praesentium quae!</p>

              <a href="/" className="btn btn-outline-success btn-sm">Read More</a>

              <a href="/" className="btn btn-outline-danger btn-sm"><i className="far fa-heart"></i></a>

            </div>

          </div>
```

```
          </div>

          <div className="col-lg-4 mb-4">

            <div className="card">

              <img src="https://images.unsplash.com/photo-1477862096227-3a1bb3b08330?ixlib=rb-1.2.1&auto=format&fit=crop&w=700&q=60" alt="" className="card-img-top" />

              <div className="card-body">

                <h5 className="card-title">CryptoGraphy</h5>

                <p className="card-text">Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ut eum similique repellat a laborum, rerum voluptates ipsam eos quo tempore iusto dolore modi dolorum in pariatur. Incidunt repellendus praesentium quae!</p>

                <a href="/" className="btn btn-outline-success btn-sm">Read More</a>

                <a href="/" className="btn btn-outline-danger btn-sm"><i className="far fa-heart"></i></a>

              </div>

            </div>

          </div>

        </div>

      </section>

    </>

  )

}


export default Class

……..

import React from 'react'
```

```
import pic1 from '../assets/contact1.jpg'

function Contact() {

   return (

      <>


         <section id="contact">

            <div className="container my-5 py-2">

               <div className="row mb-5">

                  <div className="col-12">

                     <h3 className='fs-5 text-center mb-0'>Contact Us</h3>

                     <h1 className='display-6 text-center mb-4'>Have Some
<b>Question</b></h1>

                     <hr className='w-25 mx-auto' />

                  </div>

               </div>

               <div className="row">

                  <div className="col-md-6">

                     <img src={pic1} alt="imgs" className='w-75' />

                  </div>

                  <div className="col-md-6">

                     <form action="">

                        <div className="form-group">

                           <label for="exampleFormControlInput1">Name </label>

                           <input type="email" className="form-control" id="name"
placeholder="your name" />

                        </div>
```

```
                        <div className="form-group">
                            <label for="exampleFormControlInput1">Email address</label>
                            <input type="email" className="form-control" id="email"
placeholder="your email" />
                        </div>

                        <div className="form-group">
                            <label for="exampleFormControlTextarea1">Mention Your Query
in Brief.</label>
                            <textarea className="form-control" id="message"
rows="3"></textarea>
                        </div>
                        <button type="submit" className='btn btn-outline-primary'>Send
Message <i className="fa fa-paper-plane ms-2"></i></button>
                    </form>
                </div>
            </div>
        </div>
    </section>


    </>
  )
}

export default Contact
```

……

```
import React, { useEffect } from 'react'
import { useNavigate } from 'react-router-dom'
import Card from 'react-bootstrap/Card';
import pic1 from '../assets/cirkit.jpg'



function Dashbord() {
  const history = useNavigate()
  useEffect(() => {
    const token = localStorage.getItem('TOKEN')
    if (!token) {
      history('/login')
    }
  }, [])
  return (
    <>
      <hr>
        <nav className="nav nav-pills nav-justified">
          <link className="nav-item nav-link active rounded-pill" to="/">className or
Course</link>
          <link className="nav-item nav-link rounded-pill active" to="/">Offline
className</link>
          <link className="nav-item nav-link rounded-pill active" to="/">Profile</link>
          <button className="nav-item nav-link danger active rounded-pill"
to="/">Logout</button>
        </nav>
```

```
      {/*
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////// */}

      <Card classNameName="bg-dark text-bold mt-3">

        <Card.Img src={pic1} alt="Card image" classNameName='fluid' style={{ height:
'20rem', Padding: "1rem" }} />

        <Card.ImgOverlay>

          <Card.Title>Card title</Card.Title>

          <Card.Text >

            This is a wider card with supporting text below as a natural lead-in

            to additional content. This content is a little bit longer.

          </Card.Text>

          <Card.Text>Last updated 3 mins ago</Card.Text>

        </Card.ImgOverlay>

      </Card>

      {/*
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////// */}

      <hr className="container">

        <hr className="main-body">

          {/* <!-- Breadcrumb --> */}

          <nav aria-label="breadcrumb" className="main-breadcrumb">

            <ol className="breadcrumb">

              <li className="breadcrumb-item"><link to="index.html">Home</link></li>

              <li className="breadcrumb-item"><link
to="javascript:void(0)">User</link></li>

              <li className="breadcrumb-item active" aria-current="page">User
Profile</li>

            </ol>
```

```
        </nav>

        {/* <!-- /Breadcrumb --> */}


        <hr className="row gutters-sm">

          <div className="col-md-4 mb-3">

            <div className="card">

              <div className="card-body">

                <div className="d-flex flex-column align-items-center text-center">

                  <img src="https://bootdey.com/img/Content/avatar/avatar7.png"
alt="Admin" className="rounded-circle" width="150" />

                  <div className="mt-3">

                    <h4>John Doe</h4>

                    <p className="text-secondary mb-1">Full Stack Developer</p>

                    <p className="text-muted font-size-sm">Bay Area, San Francisco,
CA</p>

                    <button className="btn btn-primary">Follow</button>

                    <button className="btn btn-outline-primary">Message</button>

                  </div>

                </div>

              </div>

            </div>

            <div className="card mt-3">

              <ul className="list-group list-group-flush">

                <li className="list-group-item d-flex justify-content-between align-items-
center flex-wrap">

                  <h6 className="mb-0"><svg xmlns="http://www.w3.org/2000/svg"
width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-
width="2" stroke-linecap="round" stroke-linejoin="round" className="feather feather-
```

globe mr-2 icon-inline"><circle cx="12" cy="12" r="10"></circle><line x1="2" y1="12" x2="22" y2="12"></line><path d="M12 2a15.3 15.3 0 0 1 4 10 15.3 15.3 0 0 1-4 10 15.3 15.3 0 0 1-4-10 15.3 15.3 0 0 1 4-10z"></path></svg>Website</h6>

https://bootdey.com

</li>

<li className="list-group-item d-flex justify-content-between align-items-center flex-wrap">

<h6 className="mb-0"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" className="feather feather-github mr-2 icon-inline"><path d="M9 19c-5 1.5-5-2.5-7-3m14 6v-3.87a3.37 3.37 0 0 0-.94-2.61c3.14-.35 6.44-1.54 6.44-7A5.44 5.44 0 0 0 20 4.77 5.07 5.07 0 0 0 19.91 1S18.73.65 16 2.48a13.38 13.38 0 0 0-7 0C6.27.65 5.09 1 5.09 1A5.07 5.07 0 0 0 5 4.77a5.44 5.44 0 0 0-1.5 3.78c0 5.42 3.3 6.61 6.44 7A3.37 3.37 0 0 0 9 18.13V22"></path></svg>Github</h6>

bootdey

</li>

<li className="list-group-item d-flex justify-content-between align-items-center flex-wrap">

<h6 className="mb-0"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" className="feather feather-twitter mr-2 icon-inline text-info"><path d="M23 3a10.9 10.9 0 0 1-3.14 1.53 4.48 4.48 0 0 0-7.86 3v1A10.66 10.66 0 0 1 3 4s-4 9 5 13a11.64 11.64 0 0 1-7 2c9 5 20 0 20-11.5a4.5 4.5 0 0 0-.08-.83A7.72 7.72 0 0 0 23 3z"></path></svg>Twitter</h6>

@bootdey

</li>

<li className="list-group-item d-flex justify-content-between align-items-center flex-wrap">

<h6 className="mb-0"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" className="feather feather-instagram mr-2 icon-inline text-danger"><rect x="2" y="2" width="20" height="20" rx="5" ry="5"></rect><path d="M16 11.37A4 4 0 1 1 12.63 8 4 4 0 0 1 16 11.37z"></path><line x1="17.5" y1="6.5" x2="17.51" y2="6.5"></line></svg>Instagram</h6>

bootdey

```
            </li>

            <li className="list-group-item d-flex justify-content-between align-items-center flex-wrap">

              <h6 className="mb-0"><svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" className="feather feather-facebook mr-2 icon-inline text-primary"><path d="M18 2h-3a5 5 0 0 0-5 5v3H7v4h3v8h4v-8h3l1-4h-4V7a1 1 0 0 1 1-1h3z"></path></svg>Facebook</h6>

              <span className="text-secondary">bootdey</span>

            </li>

          </ul>

        </div>

      </div>

      <hr className="col-md-8">

        <hr className="card mb-3">

          <hr className="card-body">

            <div className="row">

              <div className="col-sm-3">

                <h6 className="mb-0">Full Name</h6>

              </div>

              <div className="col-sm-9 text-secondary">

                Kenneth Valdez

              </div>

            </div>

            <hr>

              <div className="row">

                <div className="col-sm-3">

                  <h6 className="mb-0">Email</h6>
```

```
  </div>
  <div className="col-sm-9 text-secondary">
    fip@jukmuh.al
  </div>
</div>
<hr>
  <div className="row">
    <div className="col-sm-3">
      <h6 className="mb-0">Phone</h6>
    </div>
    <div className="col-sm-9 text-secondary">
      (239) 816-9029
    </div>
  </div>
  <hr>
    <div className="row">
      <div className="col-sm-3">
        <h6 className="mb-0">Mobile</h6>
      </div>
      <div className="col-sm-9 text-secondary">
        (320) 380-4539
      </div>
    </div>
    <hr>
      <div className="row">
```

```
                        <div className="col-sm-3">
                          <h6 className="mb-0">Address</h6>
                        </div>
                        <div className="col-sm-9 text-secondary">
                          Bay Area, San Francisco, CA
                        </div>
                      </div>
                      <hr>
                        <div className="row">
                          <div className="col-sm-12">
                            <link className="btn btn-info " target="__blank"
to={"https://www.bootdey.com/snippets/view/profile-edit-data-and-skills"}>Edit</link>
                          </div>
                        </div>
                      </hr>
                    </hr>
                    <div className="row gutters-sm">
                      <div className="col-sm-6 mb-3">
                        <div className="card h-100">
                          <div className="card-body">
                            <h6 className="d-flex align-items-center mb-3"><i
className="material-icons text-info mr-2">assignment</i>Project Status</h6>
                            <small>Web Design</small>
                            <div className="progress mb-3" style={{ height: "5px" }}>
                              <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "80%" }} aria-valuenow="80" aria-valuemin="0" aria-
valuemax="100"></div>
```

```
                        </div>

                        <small>Website Markup</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "72%" }} aria-valuenow="72" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                        <small>One Page</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "89%" }} aria-valuenow="89" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                        <small>Mobile Template</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "55%" }} aria-valuenow="55" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                        <small>Backend API</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "66%" }} aria-valuenow="66" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                      </div>

                    </div>

                    <div className="col-sm-6 mb-3">
```

```
                    <div className="card h-100">

                      <div className="card-body">

                        <h6 className="d-flex align-items-center mb-3"><i
classNam="material-icons text-info mr-2">assignment</i>Project Status</h6>

                        <small>Web Design</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "80%" }} aria-valuenow="80" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                        <small>Website Markup</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "72%" }} aria-valuenow="72" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                        <small>One Page</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "89%" }} aria-valuenow="89" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                        <small>Mobile Template</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>

                          <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "55%" }} aria-valuenow="55" aria-valuemin="0" aria-
valuemax="100"></div>

                        </div>

                        <small>Backend API</small>

                        <div className="progress mb-3" style={{ height: "5px" }}>
```

```
                                <div className="progress-bar bg-primary" role="progressbar"
style={{ width: "66%" }} aria-valuenow="66" aria-valuemin="0" aria-
valuemax="100"></div>

                            </div>

                          </div>

                        </div>

                      </div>

                    </hr>

                  </hr>

                </hr>

              </hr>

            </hr>

          </hr>

        </hr>

      </hr>

    </>

  )

}

export default Dashbord
```

……

```
import React from 'react'


function Footer() {
```

```
  return (
    <>

      <footer className="text-center text-lg-start bg-light text-muted">
        {/* <!-- Section: Social media --> */}
        <section className="d-flex justify-content-center justify-content-lg-between p-4 border-bottom">
          {/* <!-- Left --> */}
          <div className="me-5 d-none d-lg-block">
            <span>Get connected with us on social networks:</span>
          </div>
          {/* <!-- Left --> */}

          {/* <!-- Right --> */}
          <div>
            <a href="/" className="me-4 text-reset ">
              <i className="fa fa-facebook fa-2x mr-3"></i>
            </a>
            <a href="/" className="me-4 text-reset">
              <i className="fa fa-twitter fa-2x mr-3"></i>
            </a>
            <a href="/" className="me-4 text-reset">
              <i className="fa fa-google fa-2x mr-3"></i>
            </a>
            <a href="/" className="me-4 text-reset">
              <i className="fa fa-instagram fa-2x mr-3"></i>
```

```
    </a>
    <a href="/" className="me-4 text-reset">
      <i className="fa fa-linkedin fa-2x mr-3"></i>
    </a>
    <a href="/" className="me-4 text-reset">
      <i className="fa fa-github fa-2x mr-3"></i>
    </a>
  </div>
  {/* <!-- Right --> */}
</section>
{/* <!-- Section: Social media --> */}

{/* <!-- Section: Links  --> */}
<section className="">
  <div className="container text-center text-md-start mt-5">
    {/* <!-- Grid row --> */}
    <div className="row mt-3">
      {/* <!-- Grid column --> */}
      <div className="col-md-3 col-lg-4 col-xl-3 mx-auto mb-4">
        {/* <!-- Content --> */}
        <h6 className="text-uppercase fw-bold mb-4">
          <i className="fas fa-gem me-3"></i>Usha Martin University
        </h6>
        <p>
          Usha Martin University is a private university located in Ranchi,
```

Jharkhand, India. Usha Martin University was set up in 2012 vide Usha Martin University,

Jharkhand Act 2012 and is duly recognised by the University Grants Commission as a private

university.

{/* <!-- Grid column --> */}

{/* <!-- Grid column --> */}

{/* <!-- Links --> */}

<h6 className="text-uppercase fw-bold mb-4">

Products

</h6>

<a href="#!" className="text-reset">Angular</a>

<a href="#!" className="text-reset">React</a>

<a href="#!" className="text-reset">Vue</a>

<a href="#!" className="text-reset">Laravel</a>

```
  </p>
</div>
{/* <!-- Grid column --> */}

{/* <!-- Grid column --> */}
<div className="col-md-3 col-lg-2 col-xl-2 mx-auto mb-4">
  {/* <!-- Links --> */}
  <h6 className="text-uppercase fw-bold mb-4">
    Useful links
  </h6>
  <p>
    <a href="#!" className="text-reset">Pricing</a>
  </p>
  <p>
    <a href="#!" className="text-reset">Settings</a>
  </p>
  <p>
    <a href="#!" className="text-reset">Orders</a>
  </p>
  <p>
    <a href="#!" className="text-reset">Help</a>
  </p>
</div>
{/* <!-- Grid column --> */}
```

```
          {/* <!-- Grid column --> */}
          <div className="col-md-4 col-lg-3 col-xl-3 mx-auto mb-md-0 mb-4">
            {/* <!-- Links --> */}
            <h6 className="text-uppercase fw-bold mb-4">Contact</h6>
            <p><i className="fa fa-home me-3"></i> Angra Block, Jharkhand 835103,
India</p>
            <p>
              <i className="fa fa-envelope me-3 mr-1"></i>
              Ushamartin@co.in
            </p>
            <p><i className="fas fa-phone me-3"></i> + 91 9876543210</p>
            <p><i className="fas fa-print me-3"></i> + 0651 567 89</p>
          </div>
          {/* <!-- Grid column --> */}
        </div>
        {/* <!-- Grid row --> */}
      </div>
    </section>
    {/* <!-- Section: Links  --> */}

    {/* <!-- Copyright --> */}
    <div className="text-center p-4">
      © 2021 Copyright:
      <a className="text-reset fw-bold mr-2" href="/">www.ushamartin.com</a>
    </div>
    {/* <!-- Copyright --> */}
```

```
      </footer>
      {/* <!-- Footer --> */}
    </>
  )
}

export default Footer
```

……

```
import React from 'react'
import { NavLink } from 'react-router-dom';
import './App.css';
import pic1 from '../assets/icons8-java.svg'
import pic2 from '../assets/icons8-javascript.svg'
import pic3 from '../assets/icons8-python.svg'
import pic4 from '../assets/icons8-mongodb.svg'
import pic5 from '../assets/icons8-react.svg'
import pic6 from '../assets/icons8-my-sql.svg'
function Home() {
  return (
    <>
      <section id='home'>
        <div className="container">
          <div className="row justify-content-center">
            <div className="col-md-8 mt-5">
              <h1 className="display-4 fw-border mb-4 text-center">Build the skill and Enhance Your Career</h1>
```

```
                    <p className="lead text-center fs-4 mb-5">

                        Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae
dolorum molestiae

                        fugiat eum nulla placeat odio sed dignissimos voluptas ea sint adipisci
rerum ab aperiam

                        vel neque eligendi, omnis recusandae!

                    </p>

                    <div className="buttons d-flex justify-content-center">

                        <NavLink to='/contact' className='btn btn-outline-dark me-4
rounded-pill px-4 py-2'>Get Quote</NavLink>

                        <NavLink to='/services' className='btn btn-outline-primary ml-3
rounded-pill px-4 py-2'>Our Services</NavLink>

                    </div>

                </div>

            </div>

        </div>

        <h4 className='display-4 fw-border mb-4 text-center mt-3'> Trending
Technology which influencing IT Sector</h4>

        <hr className='w-50 mx-auto' />

        {/* <div className="container my-5 py-5"> */}

        <div className="row">

            <div className="col-12">

            </div>

        </div>

        <div className="row mt-5">

            <div className="col-md-4">

                <div className="card p-3">

                    <div className="card-body text-center">
```

```
                        <img src={pic1} alt="img" className='w-50' />

                        <h5 className="card-title mb-1 fs-4 fw-bold">Java Core + Advance
</h5>

                        <p className="card-text">Some quick example text to build on the
card title and make up the bulk of the card's content.</p>

                    </div>

                </div>

            </div>

            <div className="col-md-4">

                <div className="card p-3">

                    <div className="card-body text-center">

                        <img src={pic2} alt="img" className='w-50' />

                        <h5 className="card-title mb-1 fs-4 fw-bold"> Java-Script </h5>

                        <p className="card-text">Some quick example text to build on the
card title and make up the bulk of the card's content.</p>

                    </div>

                </div>

            </div>

            <div className="col-md-4">

                <div className="card p-3">

                    <div className="card-body text-center">

                        <img src={pic3} alt="img" className='w-50' />

                        <h5 className="card-title fs-4 fw-bold">Python</h5>

                        <p className="card-text">Some quick example text to build on the
card title and make up the bulk of the card's content.</p>

                    </div>

                </div>
```

```
            </div>
          </div>
          <div className="row mt-5">
            <div className="col-md-4">
              <div className="card p-3">
                <div className="card-body text-center">
                  <img src={pic4} alt="img" className='w-50' />
                  <h5 className="card-title mb-1 fs-4 fw-bold">Mongo-DB </h5>
                  <p className="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
                </div>
              </div>
            </div>
            <div className="col-md-4">
              <div className="card p-3">
                <div className="card-body text-center">
                  <img src={pic5} alt="img" className='w-50' />
                  <h5 className="card-title mb-1 fs-4 fw-bold">React-js</h5>
                  <p className="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
                </div>
              </div>
            </div>
            <div className="col-md-4">
              <div className="card p-3">
                <div className="card-body text-center">
```

```
                        <img src={pic6} alt="img" className='w-50' />

                        <h5 className="card-title  fs-4 fw-bold">MySql and Sql</h5>

                        <p className="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>

                    </div>

                </div>

            </div>

        </div>

        {/* </div> */}

    </section>

  </>

 )

}


export default Home
```

……

```
import React, { useState } from 'react'

import { NavLink, useNavigate } from 'react-router-dom'

import axios from 'axios'

function Login() {

 // ////////////////////////////////////////////////////////////////////////////

 const history = useNavigate()

 const [email, setEmail] = useState("")

 const [password, setPassword] = useState("")

 const loginUser = (e) => {

  e.preventDefault();
```

```
    console.log(email, password,)
    axios.post("http://localhost:8000/signin",
        {
            email: email,
            password: password
        }
    ).then(res => {
        console.log(res.data)
        if (res.data.code === 500) {
            alert("User Not Found redirecting to Sign-up")
            history('/register')
        }
        if (res.data.code === 404) {
            alert("Password is Wrong")
        }
        if (res.data.code === 200) {
            alert("User Login Successful...")
            history('/dashbord')
            localStorage.setItem('TOKEN', res.data.token)
        }
    }).catch(err => { console.log(err) })
}
// ////////////////////////////////////////////////////////////////////////////
return (
    <>
```

```
<div className="container shadow my-5">

<div className="row">

<div className="ml-5 col-md-5 d-flex flex-column text-dark align-item-center justify-content-center form">

<h1 className="display-4 ml-5 fw-bolder">Welcome Back</h1>

<p className="lead text-center">Enter Your Credential To Login</p>

<h5 className='mb-4 ml-3'>Are You New to This ? Just Go with it ↴</h5>

<div className="container justify-content-center">

<NavLink to='/signup' className='btn btn-outline-dark rounded-pill pb-2 w-100'>Register</NavLink>

</div>

</div>

<div className="col-md-6 p-5">

<div className="display-6 fw-bolder mb-5 text-align-center">Login</div>

<form method='POST'>

<div className="form-group">

<label >Email address</label>

<input type="email" name='email' className="form-control" value={email} onChange={(e) => setEmail(e.target.value)} id="email" aria-describedby="emailHelp" />

<small id="emailHelp" className="form-text text-muted">We'll never share your email with anyone else.</small>

</div>

<div className="form-group">

<label >Password</label>

<input type="password" name='password' className="form-control" value={password} onChange={(e) => setPassword(e.target.value)} id="password" />
```

```
                    </div>

                    <div className="form-group form-check">

                        <input type="checkbox" className="form-check-input"
id="exampleCheck1" />

                        <label className="form-check-label" >Remember Me</label>

                    </div>

                    <button type="submit" className="btn btn-primary"
onClick={loginUser}>Submit</button>

                </form>

            </div>

        </div>

    </div>

  </>

 )

}


export default Login
```

……

```
import React from 'react'

import { NavLink } from 'react-router-dom'

import Logo from '../assets/icons8-redux.svg'

function Navbar() {

 return (

  <>


    {/* ////////////////////////////////////////// */}
```

```
<nav className="navbar navbar-expand-lg navbar-light shadow">

    <button className="navbar-toggler" type="button" data-toggle="collapse" data-target="/navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span className="navbar-toggler-icon"></span>
    </button>

    <div className="collapse navbar-collapse" id="navbarSupportedContent">
      <ul className="navbar-nav mr-auto">
        <li className="nav-item active">
          <NavLink className="nav-link" to="/">Home <span className="sr-only">(current)</span></NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link" to="/about">About</NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link" to="/lecture">Lecture</NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link" to="/class">Class</NavLink>
        </li>
        <li className="nav-item">
          <NavLink className="nav-link" to="/contact">Contact</NavLink>
        </li>
        <li className="nav-item">
```

```
                    <NavLink className="nav-link" to="/services">Services</NavLink>

                </li>

            </ul>

            <NavLink className="navbar-brand fw-border fs-4 max-auto" to="/"><img
src={Logo} alt="" />Skill-Up</NavLink>

            <NavLink to='login' className="btn btn-outline-primary mr-2 px-4
rounded-pill">

                <i className='fa fa-sign-in me-2'></i>Login</NavLink>

            <NavLink to='/signup' className="btn btn-outline-primary ms-2 px-4
rounded-pill">

                <i className='fa fa-user-plus me-2'></i>Register</NavLink>

        </div>

    </nav>



    {/* ////////////////////////////////////////////// */}



    </>

  )

}


export default Navbar
```

……

```
import React from 'react'


function Services() {

  return (

    <>
```

```
<section id='service'>
  <div className="container my-5 py-5">
    <div className="row">
      <div className="col-12">
        <h3 className="fs-5 text-center mb-0">Our Services</h3>
        <h1 className="display-6 text-center mb-4">Our <b> Awesome</b> Services</h1>
        <hr className='w-25 mx-auto' />
      </div>
    </div>
    <div className="row mt-5">
      <div className="col-md-4">
        <div className="card p-3">
          <div className="card-body text-center">
            <i className='fa fa-cogs fa-4x mb-4 text-primary'></i>
            <h5 className="card-title mb-3 fs-4 fw-bold">Highly Skilled-Productivity </h5>
            <p className="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
          </div>
        </div>
      </div>
      <div className="col-md-4">
        <div className="card p-3">
          <div className="card-body text-center">
            <i className='fa fa-mobile fa-4x mb-4 text-primary'></i>
```

```
                        <h5 className="card-title mb-3 fs-4 fw-bold">One to One Mentor
</h5>

                        <p className="card-text">Some quick example text to build on
the card title and make up the bulk of the card's content.</p>

                    </div>

                </div>

            </div>

            <div className="col-md-4">

                <div className="card p-3">

                    <div className="card-body text-center">

                        <i className='fa fa-file-code-o fa-4x mb-4 text-primary'></i>

                        <h5 className="card-title mb-3 fs-4 fw-bold">Expertise in
Technology</h5>

                        <p className="card-text">Some quick example text to build on
the card title and make up the bulk of the card's content.</p>

                    </div>

                </div>

            </div>

        </div>

        <div className="row mt-5">

            <div className="col-md-4">

                <div className="card p-3">

                    <div className="card-body text-center">

                        <i className='fa fa-laptop fa-4x mb-4 text-primary'></i>

                        <h5 className="card-title mb-3 fs-4 fw-bold">No leave (record)
policy </h5>

                        <p className="card-text">Some quick example text to build on
the card title and make up the bulk of the card's content.</p>
```

```
                    </div>
                </div>
            </div>
            <div className="col-md-4">
                <div className="card p-3">
                    <div className="card-body text-center">
                        <i className='fa fa-money fa-4x mb-4 text-primary'></i>
                        <h5 className="card-title mb-3 fs-4 fw-bold">Free Source</h5>
                        <p className="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
                    </div>
                </div>
            </div>
            <div className="col-md-4">
                <div className="card p-3">
                    <div className="card-body text-center">
                        <i className='fa fa-star-half-o fa-4x mb-4 text-primary'></i>
                        <h5 className="card-title mb-3 fs-4 fw-bold">Doubt Clearing</h5>
                        <p className="card-text">Some quick example text to build on the card title and make up the bulk of the card's content.</p>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```
            </section>
        </>
    )
}

export default Services
```

……

```
import React, { useState } from 'react'
import { NavLink, useNavigate } from 'react-router-dom'
import axios from 'axios'
function Signup() {
    // ////////////////////////////////////////////////////////////////////////////////////////////////////
    const history = useNavigate();
    const [name, setName] = useState("")
    const [email, setEmail] = useState("")
    const [profession, setProfession] = useState("")
    const [phone, setPhone] = useState("")
    const [password, setPassword] = useState("")
    const [cpassword, setCpassword] = useState("")

    const handelSubmit = (e) => {
        e.preventDefault();
        console.log(name, email, profession, phone, password, cpassword)
        axios.post("http://localhost:8000/register",
            {
```

```
            name: name,
            email: email,
            profession: profession,
            phone: phone,
            password: password,
            cpassword: cpassword
        }
    ).then(res => {
        console.log(res.data)
        if (res.data.code !== 200) {
            alert("Error Found")
            return
        } else {
            alert('Register Successfully... Redirecting to Login..')
            history('/login')
        }
    }).catch(err => { console.log(err) })
}

// ////////////////////////////////////////////////////////////////////////////////////////////////

return (
    <>
        <div className="container shadow my-5">
            <div className="row justify-content-center-">
                <div className="ml-5 col-md-5 d-flex flex-column text-dark align-item-center justify-content-center form order-2">
```

```
                <h1 className="display-4  text-center fw-bolder">Welcome </h1>

                <p className="lead text-center ">Enter Your Credential To Sign-Up</p>

                <h5 className='mb-3 text-center'>Or</h5>

                <h5 className='mb-3 ml-3 text-center'>Already User ?  Just Go with it ↷</h5>

                <div className="container justify-content-center">

                  <NavLink to='/login' className='btn text-center justify-content-center btn-outline-dark rounded-pill  mx-5  pb-2 w-100'>Login</NavLink>

                </div>

              </div>

              <div className="col-md-6 p-5">

                <form method='POST'>

                  <div className="form-group" id='register-form'>

                    <label >Name</label>

                    <input type="text" name='name' value={name} onChange={(e) => { setName(e.target.value) }} className="form-control" id="name" />

                  </div>

                  <div className="form-group">

                    <label>Email address</label>

                    <input type="email" name='email' value={email} onChange={(e) => { setEmail(e.target.value) }} className="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" />

                  </div>

                  <div className="form-group">

                    <label>profession</label>

                    <input type="text" name='profession' value={profession} onChange={(e) => { setProfession(e.target.value) }} className="form-control" id="profession" />

                  </div>
```

```
                    <div className="form-group">

                        <label >Phone</label>

                        <input type="text" name='phone' value={phone} onChange={(e) =>
{ setPhone(e.target.value) }} className="form-control" id="phone" />

                    </div>

                    <div className="form-group">

                        <label>Password</label>

                        <input type="password" name='password' value={password}
onChange={(e) => { setPassword(e.target.value) }} className="form-control"
id="password" />

                    </div>

                    <div className="form-group">

                        <label>Confirm Password</label>

                        <input type="password" name='cpassword' value={cpassword}
onChange={(e) => { setCpassword(e.target.value) }} className="form-control"
id="cpassword" />

                    </div>

                    <button type="submit" onClick={handelSubmit} className="btn btn-
primary">Submit</button>

                </form>

            </div>

        </div>

    </div>

  </>

 )

}



export default Signup
```

……

## ***CSS parts of application***

```
@import "bootstrap/functions";

@import "bootstrap/variables";

@import "bootstrap/mixins";


$font-family-sans-serif-2: 'Poppins', sans-serif;

$font-family-serif: 'Source Serif Pro', serif;




body {

    font-family: $font-family-sans-serif;

    background-color: #272343;

}

p {

    color: darken(#ccc, 10%);

    font-weight: 300;

}

h1, h2, h3, h4, h5, h6,

.h1, .h2, .h3, .h4, .h5, .h6 {

    font-family: $font-family-sans-serif;

}
```

```
a {
    transition: .3s all ease;
    &, &:hover {
        text-decoration: none!important;
    }
}
.content {
    padding: 7rem 0;
     background-color: #272343;
}
h2 {
    font-size: 20px;
}
```

```
/* 10 */

.site-blocks-cover {
    background-size: cover;
    background-repeat: no-repeat;
    background-position: top;
    background-attachment: fixed ;
```

```
position: relative;
&.overlay {
	position: relative;
	&:before {
		position: absolute;
		content: "";
		left: 0;
		bottom: 0;
		right: 0;
		top: 0;
		background: rgba($black, .2);
	}
}
&, & > .container > .row {
	min-height: 600px;
	height: calc(100vh);
}
h1 {
	font-size: 8rem;
	font-weight: 900;
	line-height: 1;
	@include media-breakpoint-down(md) {
		font-size: 4rem;
	}
}
```

```
.sub-text {
	font-size: 1.4rem;
	color: lighten($black, 50%);
	font-weight: 300;
	@include media-breakpoint-down(md) {
		color: $black;
	}
}
.img-wrap {
	position: absolute;
	z-index: -1;
	width: calc(100% - 50%);
	top: 0;
	height: 100%;
	z-index: 2;
	right: 50%;
	min-height: 600px;
	overflow: hidden;
	border-bottom-right-radius: 200px;
	@include media-breakpoint-down(md) {

		width: 100%;
		right: 0%;
		top: 0;
		.hero-slider {
```

```
            &:before {
                position: absolute;
                content: "";
                background: $white;
                opacity: .5;
                z-index: 2;
                top: 0;
                left: 0;
                right: 0;
                bottom: 0;
            }
        }
    }
    .slide {
        height: 100vh;
        position: relative;
        img {
            position: absolute;
            top: 0;
            height: 100%;
            width: 100%;
            object-fit: cover;
        }
    }
```

```
    }
    .intro {
        z-index: 3;
        position: relative;
        .heading {

            margin-left: -150px;
            @include media-breakpoint-down(md) {
                margin-left: 0;
            }
        }
        .text {
            padding-left: 50px;
            @include media-breakpoint-down(md) {
                padding-left: 0;
            }
        }
    }
}
.site-menu-toggle {
    .menu-text {
        position: relative;
        top: -6px;
        text-transform: uppercase;
    }
```

```
}

.hero-slider {

    .owl-nav {
        position: absolute;
        bottom: 120px;
        right: 50px;
        z-index: 100;
        .owl-prev, .owl-next {
            width: 50px;
            height: 50px;
            text-align: center;
            line-height: 50px;
            border-radius: 50%;
            background: $primary;
            color: $white;
            margin-bottom: 10px;
            font-size: 1.3rem;
            span {
                color: $white;
                font-size: 30px;
            }
            &:active, &:focus {
                outline: none;
```

```
                    }
                }
            }
        }
```

```
.btn-pill {
    border-radius: 30px;
    padding: 15px 30px;
    border-color: $white;
    color: $white;
    &:hover {
        background: $white;
        border-color: $white;
        color: $black;
    }
}
```

# REFERENCES


- https://reactjs.dev/docs/getting-started.
- https://reactjs.dev/docs/components-and-apis
- https://reactjs.dev/docs/accessibilityinfo
- https://reactjs.dev/architecture/overview
- https://nodejs.org/en/docs/
- https://www.w3schools.com/js/default.asp
- https://www.zegocloud.com/docs/api-reference