PROJECT REPORT

**On**

**“SmartBrain”**

**Submitted by**

**Shreyash Srivastava**

**(191500787)**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



**GLA University**

**Mathura- 281406, INDIA**

**2021-2022**

**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**



**Declaration**

We hereby declare that the work which is being presented in the Project “**SmartBrain”,** in partial fulfillment of the requirements for Project viva voice, is an authentic record of mine carried out while learning **The Complete Web Development – Zero to Mastery** course from **Udemy**.

Group Members: Shreyash Srivastava (191500787)

Course: B.Tech (Computer Science and Engineering)

Year: 3rd

**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**



**About the Project**

We will be building a face detection application. The application will have two buttons, one for picking an image and one for detecting the faces. The button for picking an image will ask the user to select a source. Once a photo is selected, the button for detecting faces becomes visible. Tapping this button will send a request to face detection API, which returns the data for the detected faces. Using the response of the API, little boxes are drawn around the detected faces.

**Motivation**

A **facial recognition system** is a technology capable of matching a [human face](https://en.wikipedia.org/wiki/Human_face) from a [digital image](https://en.wikipedia.org/wiki/Digital_image) or a [video frame](https://en.wikipedia.org/wiki/Film_frame) against a [database](https://en.wikipedia.org/wiki/Database) of faces, typically employed to [authenticate](https://en.wikipedia.org/wiki/Authenticate) users through [ID verification services](https://en.wikipedia.org/wiki/ID_verification_service), works by pinpointing and measuring facial features from a given image. Facial recognition systems are employed throughout the world today by governments and private companies.[[3]](https://en.wikipedia.org/wiki/Facial_recognition_system#cite_note-3) Their effectiveness varies, and some systems have previously been scrapped because of their ineffectiveness. The use of facial recognition systems has also raised controversy, with claims that the systems violate citizens' privacy, commonly make incorrect identifications, encourage [gender norms](https://en.wikipedia.org/wiki/Gender_role) and [racial profiling](https://en.wikipedia.org/wiki/Racial_profiling), and do not protect important biometric data. These claims have led to the ban of facial recognition systems in several cities in the [United States](https://en.wikipedia.org/wiki/United_States).[[4]](https://en.wikipedia.org/wiki/Facial_recognition_system#cite_note-:10-4) As a result of growing societal concerns [Meta](https://en.wikipedia.org/wiki/Meta_Platforms) announced[[5]](https://en.wikipedia.org/wiki/Facial_recognition_system#cite_note-5) that it plans to shut down [Facebook facial recognition system](https://en.wikipedia.org/wiki/DeepFace), deleting the face scan data of more than one billion users[[6]](https://en.wikipedia.org/wiki/Facial_recognition_system#cite_note-6). This change will represent one of the largest shifts in facial recognition usage in the technology’s history.

Human-robot interaction receives an increasing attention among the researches in

different areas of interest during last years. In the current project we address one

of the computer vision tasks, involved in developing of such an interactive system.

Namely, we are interested in detecting human user presence and tracking of his at-

tention. Possible applications for this would include receptionist robot, which needs

to perform particular actions when human approaches or leaves the system, interac-

tive games, commercials etc.

Object detection has been a great challenge from the past few years. In computer

vision human face detection is an important research topic. It is needed for many

computer applications like HCI, surveillance, human-robot interaction, etc. In this

ﬁeld facial tracking is ﬁnding more growth of use in security and safety applications

to detect various situations1 . This tracking domain can be used to control or com-

municate with robots.

Detecting human faces in a video is a great challenging problem. These conﬁgu-

rations may be like angle of view, background intensity, and various illuminations.

This is due to high variety of conﬁgurations that may occur. The complexness of the

face results in a particular degree of issue for fast detection and tracking [1].

For face detection and tracking in a given video sequence different algorithms

have been introduced over the past few years. Each algorithm has got its own ad-

vantages and disadvantages. But any facetracking algorithm will have some errors

which will cause deviation from the required object. The tracker can be accurate if

and only if it is able to minimize this deviation. The technique used in this report is

one of the effective approaches. It is quicker and simpler and we make use of the

eigen vectors for detecting the faces along with the superﬁcial points of our faces.

Object detection and tracking are important in many computer vision appli-

cations including activity recognition, automotive safety, and surveillance. In this

example, you will develop a simple face tracking system by dividing the tracking

**Requirements**

**a). Software Requirements:**

* Technology Implemented: React.js, Node.js
* Languages/Technologies Used: Front-End Technologies
* IDE Used: Sublime Text
* Web Browser: Google Chrome
* GitHub: GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub Repository: A GitHub repository can be used to store a development project. It can contain folders and any type of files (HTML, CSS, JavaScript, Documents, Data, Images). A GitHub repository should also include a license file and a README file about the project. A GitHub repository can also be used to store ideas, or any resources that you want to share.
* Sublime Text: **Sublime Text** is a commercial [source code editor](https://en.wikipedia.org/wiki/Source_code_editor). It natively supports many [programming languages](https://en.wikipedia.org/wiki/Programming_languages) and [markup languages](https://en.wikipedia.org/wiki/Markup_languages). Users can expand its functionality with [plugins](https://en.wikipedia.org/wiki/Plug-in_(computing)), typically community-built and maintained under [free-software licenses](https://en.wikipedia.org/wiki/Free_software_licenses). To facilitate plugins, Sublime Text features a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [API](https://en.wikipedia.org/wiki/API).

**b). Hardware Requirements:**

* Operating System: Windows 10, Linux, MacOS
* RAM: 2GB (Browser supported)
* Hardware Devices: Computer System
* Hard Disk: 256GB

**SmartBrain**

**Abstract**

The face is one of the easiest ways to distinguish the individual identity of each other. Face recognition is a personal identification system that uses personal characteristics of a person to identify the person's identity. Human face recognition procedure basically consists of two phases, namely face detection, where this process takes place very rapidly in humans, except under conditions where the object is located at a short distance away, the next is the introduction, which recognize a face as individuals. Stage is then replicated and developed as a model for facial image recognition (face recognition) is one of the much-studied biometrics technology and developed by experts. Facial recognition could be a technology that involves classifying and recognizing human faces, largely by mapping individual face expression and recording the distinctive quantitative relation mathematically and storing the info as a face print. The face detection in your mobile camera makes use of this technology.

Facial recognition could be a biometric technology that uses the distinguishable facial expression to spot someone.

**Contents**

**Abstract…………………………………………………**06

1. **Introduction:**

Introduction........…......08

1. **Technologies Used:**

HTML………………………………………….09

CSS…………………………………………….09-10

Javascript………………………………..............10

React.js…………………………………………10

Pre-requisite…………………………………….11

**3. List of Figures...……………………………………**12-14

**4. Software Testing..............................................................**15-18

**5.Conclusion……………………………………………**19

**6.Bibliography………………………………………….**20

**Chapter 1**

**Introduction**

## A Brief Introduction To Clarifai

We will be using [Clarifai](https://www.clarifai.com/), a platform for visual recognition that offers a free tier for developers. They offer a comprehensive set of tools that enable you to manage your input data, annotate inputs for training, create new models, predict and search over your data. However, there are other face recognition API that you can use. Their documentation will help you to integrate them into your app, as they all almost use the same model and process for detecting a face.

## Getting Started With Clarifai API [#](https://www.smashingmagazine.com/2020/06/facial-recognition-web-application-react/#getting-started-with-clarifai-api)

One of the Clarifai model is called [Face Detection](https://www.clarifai.com/models/face-detection-image-recognition-model-a403429f2ddf4b49b307e318f00e528b-detection). This particular model returns probability scores on the likelihood that the image contains human faces and coordinates locations of where those faces appear with a bounding box. This model is great for anyone building an app that monitors or detects human activity. The [Predict API](https://docs.clarifai.com/api-guide/predict) analyzes your images or videos and tells you what’s inside of them. The API will return a list of concepts with corresponding probabilities of how likely it is that these concepts are contained within the image.

Human-robot interaction receives an increasing attention among the researches in

different areas of interest during last years. In the current project we address one

of the computer vision tasks, involved in developing of such an interactive system.

Namely, we are interested in detecting human user presence and tracking of his at-

tention. Possible applications for this would include receptionist robot, which needs

to perform particular actions when human approaches or leaves the system, interac-

tive games, commercials etc.

Object detection has been a great challenge from the past few years. In computer

vision human face detection is an important research topic. It is needed for many

computer applications like HCI, surveillance, human-robot interaction, etc. In this

ﬁeld facial tracking is ﬁnding more growth of use in security and safety applications

to detect various situations1 . This tracking domain can be used to control or com-

municate with robots.

Detecting human faces in a video is a great challenging problem. These conﬁgu-

rations may be like angle of view, background intensity, and various illuminations.

This is due to high variety of conﬁgurations that may occur. The complexness of the

face results in a particular degree of issue for fast detection and tracking [1].

For face detection and tracking in a given video sequence different algorithms

have been introduced over the past few years. Each algorithm has got its own ad-

vantages and disadvantages. But any facetracking algorithm will have some errors

which will cause deviation from the required object. The tracker can be accurate if

and only if it is able to minimize this deviation. The technique used in this report is

one of the effective approaches. It is quicker and simpler and we make use of the

eigen vectors for detecting the faces along with the superﬁcial points of our faces.

Object detection and tracking are important in many computer vision appli-

cations including activity recognition, automotive safety, and surveillance. In this

example, you will develop a simple face tracking system by dividing the tracking

Human-robot interaction receives an increasing attention among the researches in

different areas of interest during last years. In the current project we address one

of the computer vision tasks, involved in developing of such an interactive system.

Namely, we are interested in detecting human user presence and tracking of his at-

tention. Possible applications for this would include receptionist robot, which needs

to perform particular actions when human approaches or leaves the system, interac-

tive games, commercials etc.

Object detection has been a great challenge from the past few years. In computer

vision human face detection is an important research topic. It is needed for many

computer applications like HCI, surveillance, human-robot interaction, etc. In this

ﬁeld facial tracking is ﬁnding more growth of use in security and safety applications

to detect various situations1 . This tracking domain can be used to control or com-

municate with robots.

Detecting human faces in a video is a great challenging problem. These conﬁgu-

rations may be like angle of view, background intensity, and various illuminations.

This is due to high variety of conﬁgurations that may occur. The complexness of the

face results in a particular degree of issue for fast detection and tracking [1].

For face detection and tracking in a given video sequence different algorithms

have been introduced over the past few years. Each algorithm has got its own ad-

vantages and disadvantages. But any facetracking algorithm will have some errors

which will cause deviation from the required object. The tracker can be accurate if

and only if it is able to minimize this deviation. The technique used in this report is

one of the effective approaches. It is quicker and simpler and we make use of the

eigen vectors for detecting the faces along with the superﬁcial points of our faces.

Object detection and tracking are important in many computer vision appli-

cations including activity recognition, automotive safety, and surveillance. In this

example, you will develop a simple face tracking system by dividing the tracking

Human-robot interaction receives an increasing attention among the researches in

different areas of interest during last years. In the current project we address one

of the computer vision tasks, involved in developing of such an interactive system.

Namely, we are interested in detecting human user presence and tracking of his at-

tention. Possible applications for this would include receptionist robot, which needs

to perform particular actions when human approaches or leaves the system, interac-

tive games, commercials etc.

Object detection has been a great challenge from the past few years. In computer

vision human face detection is an important research topic. It is needed for many

computer applications like HCI, surveillance, human-robot interaction, etc. In this

ﬁeld facial tracking is ﬁnding more growth of use in security and safety applications

to detect various situations1 . This tracking domain can be used to control or com-

municate with robots.

Detecting human faces in a video is a great challenging problem. These conﬁgu-

rations may be like angle of view, background intensity, and various illuminations.

This is due to high variety of conﬁgurations that may occur. The complexness of the

face results in a particular degree of issue for fast detection and tracking [1].

For face detection and tracking in a given video sequence different algorithms

have been introduced over the past few years. Each algorithm has got its own ad-

vantages and disadvantages. But any facetracking algorithm will have some errors

which will cause deviation from the required object. The tracker can be accurate if

and only if it is able to minimize this deviation. The technique used in this report is

one of the effective approaches. It is quicker and simpler and we make use of the

eigen vectors for detecting the faces along with the superﬁcial points of our faces.

Object detection and tracking are important in many computer vision appli-

cations including activity recognition, automotive safety, and surveillance. In this

example, you will develop a simple face tracking system by dividing the tracking

**Chapter 2**

**Technologies Used**

**HTML**

HTML is the standard markup language for creating Web pages.Every website you open in your web browser, from social networks to music services, uses HTML. A look under the hood of any website would reveal a basic HTML code page, written with an HTML structure editor, providing structure for all the page’s components, including its header element, footer element, main content, and other inline elements.When using HTML for your website, you get to control nearly everything on your website. Because you build your website from the core using a markup language, accessing, and modifying the code is easier than with WordPress. This makes it much more flexible when integrating certain additional features.

**CSS**

CSS is the language we use to style a Web page.With CSS, we are able to create rules, and apply those rules to many elements within the website. This approach offers many advantages when site-wide changes are required by a client. Since the content is completely separated from the design, we can make those changes in our Style Sheet and have it effect every applicable instance.

Since rules are only downloaded once by the browser, then are cached and used for each page load, the use of CSS can lead to lighter page loads, and improved performance.

This contributes to lighter server load and lower requirements, which overall saves money for our clients. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language.

**Javascript**

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive.

JavaScript enables dynamic interactivity on websites when it is applied to an HTML document.JavaScript helps the users to build modern web applications to interact directly without reloading the page every time. JavaScript is commonly used to dynamically modify HTML and CSS to update a user interface by the DOM API. It is mainly used in web applications.

**React.js Front End**

The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML.

React’s strong suit is handling stateful, data-driven interfaces with minimal code and minimal pain, and it has all the bells and whistles you’d expect from a modern web framework: great support for forms, error handling, events, lists, and more.

**Pre-requisite**

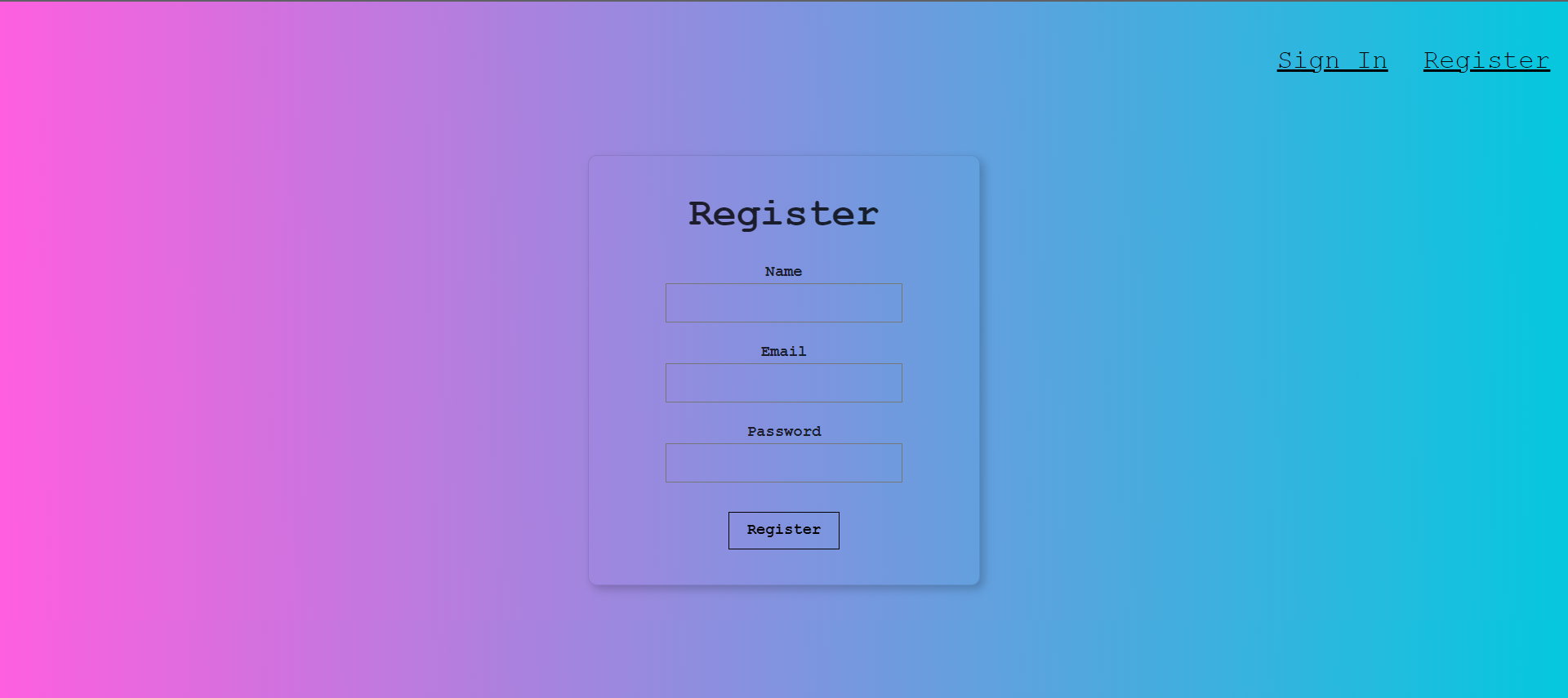
Hands-on knowledge of JavaScript, HTML and CSS is essential before working on the concepts for making of webpages. Make sure that you have the browser or chrome installed and running before opening website.

And knowledge about how to use Github is a must, and Git commands. Also, we have to make our website live on Github server, and also have to maintain our whole project on Github, so knowledge of it is very necessary.

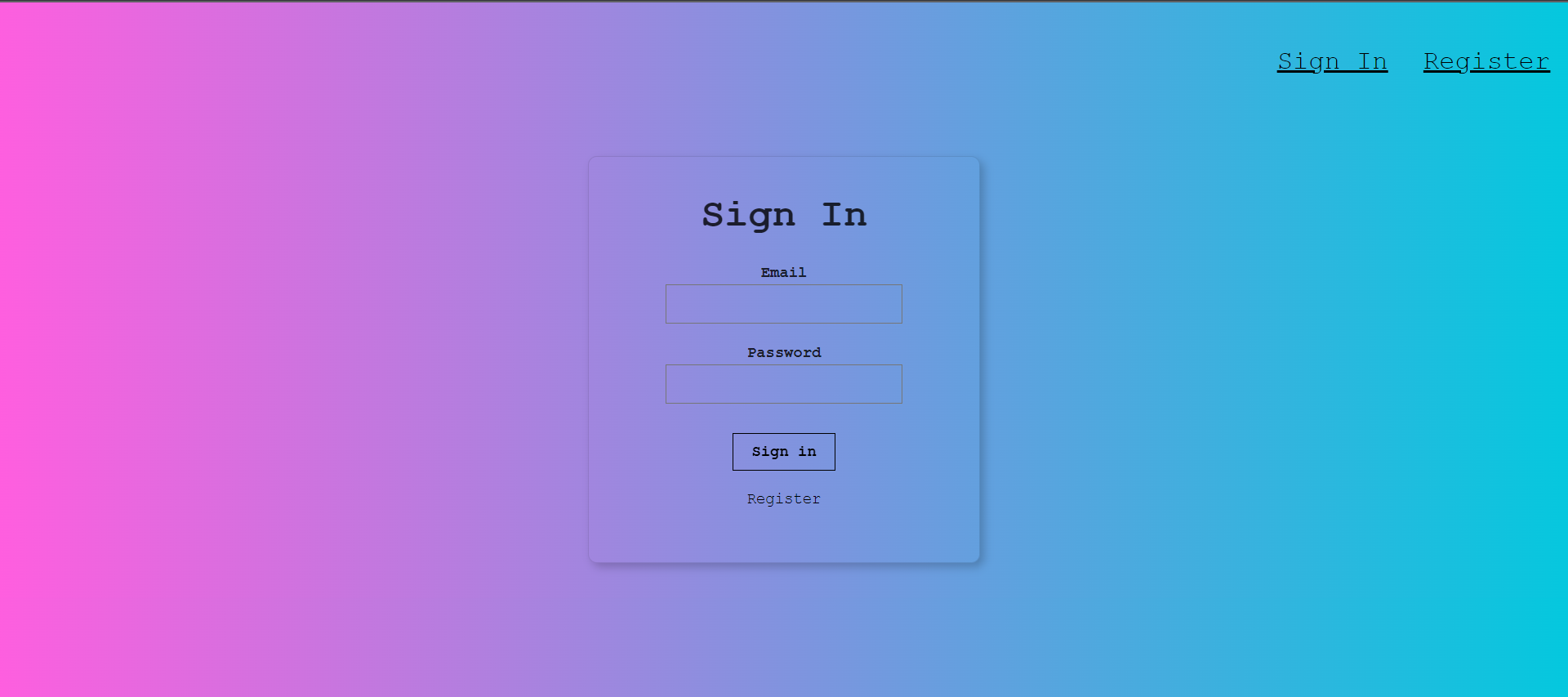
**Chapter 3**

**List of Figures**

**1. REGISTER Page**

****

**2. SIGN IN Page**

****

**3. MAIN page**



**Chapter 4**

**Software Testing**

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing include designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques. System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data. In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking. Source code clarity is enhanced by structured coding techniques, by good coding style, by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages. The implementation team should be provided with a well-defined set of software requirement, an architectural design specification, and a detailed design description. Each team member must understand the objectives of implementation.

4.1 TERMINOLOGY

Error The term error is used in two ways. It refers to the difference between the actual output of software and the correct output, in this interpretation, error is essential a measure of the difference between actual and ideal. Error is also toused to refer to human action that result in software containing a defect or fault.

Fault is a condition that causes to fail in performing its required function. A fault is a basic reason for software malfunction and is synonymous with the commonly used term Bug.

Failure is the inability of a system or component to perform a required function according to its specifications. A software failure occurs if the behavior of the software is the different from the specified behavior. Failure may be caused due to functional or performance reasons.

4.2 TYPES OF TESTING

**a. Unit Testing**The term unit testing comprises the sets of tests performed by an individual programmer prior to integration of the unit into a larger system. A program unit is usually small enough that the programmer who developed it can test it in great detail, and certainly in greater detail than will be possible when the unit is integrated into an evolving software product. In the unit testing the programs are tested separately, independent of each other. Since the check is done at the program level, it is also called program teasing.

**b. Module Testing**A module and encapsulates related component. So can be tested without other system module.

**c. Subsystem Testing**Subsystem testing may be independently design and implemented common problems are sub-system interface mistake in this checking we concenton it. There are four categories of tests that a programmer will typically perform on a program unit.

i Functional test

ii Performance test

iii Stress test

iv Structure test

**Functional Test**Functional test cases involve exercising the code with Nominal input values for which expected results are known; as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values.

**Performance Test**Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device utilization by the program unit. A certain amount of avoid expending too much effort on fine-tuning of a program unit that contributes little to the overall performance of the entire system. Performance testing is most productive at the subsystem and system levels.

**Stress Test** Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks.

**Structure Test** Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. Some authors refer collectively to functional performance and stress testing as “black box” testing. While structure testing is referred to as “white box” or “glass box” testing. The major activities in structural testing are deciding which path to exercise, deriving test date to exercise those paths, determining the test coverage criterion to be used, executing the test, and measuring the test coverage achieved when the test cases are exercised.

**Chapter 5**

**Conclusion**

Our project repository is available at

<https://aloksrivastava7.github.io/Voila/>

**Chapter 6**

**Bibliography**

[**www.google.com**](http://www.google.com)

[**www.w3schools.com**](http://www.w3schools.com)

[**www.udemy.com**](http://www.udemy.com)

[**www.tachyons.io/**](http://www.tachyons.io/)