**PROJECT REPORT**

**on**

**FACE RECOGNITION LOGIN FREATURE & VOICE SEARCH**

**IMPLLEMENTED ON A WEBSITE**

**(CSE VI Semester Mini project)**

**2021 – 2022**

****

**Submitted to:**

Mrs. Himani Sivaraman

(CC-CSE-E-VI-Sem)

**Submitted by:**

Mr. Priyansu Bisht

**Roll. No.:** 1918571

CSE-E-VI-Sem

**Session:** 2021-2022

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**GRAPHIC ERA HILL UNVERSITY, DEHRADUN**

**CERTIFICATE**

### This is to certify that Mr. Priyansu Bisht (Roll no.: 1918571) has successfully completed and submitted the project, its synopsis, and its report on the topic ‘Face Recognition Login Feature & Voice Search Implemented on a Website’, for the fulfilment of B Tech – CSE Mini Project VI Semester.

### The aforementioned candidate has completed the project to the best of my/our knowledge. I/We wish him the best for his future endeavours..

Date: 28-June-2022

Mrs. Himani Sivaraman)

**Class Co-Ordinator**

**CSE-E-VI-Sem**

(CSE Department)

Graphic Era Hill University, Dehradun

**ACKNOWLEDGMENT**

I would like to express my gratitude and my humblest of the humble respect towards the Almighty, the most Beneficent, and the most Merciful for my successful completion of the previously mentioned project and its components.

I wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

I would like to forward my thanks and the warmest of regards to our class coordinator Mrs. Himani Shivaraman, for his guidance and camaraderie during and after the course towards not just me, but all of my classmates as well.

I am hugely indebted by all the people who helped me during, before, and after the course completion. I would like to forward my regards and thanks to all of the mentioned above.

**Mr. Priyansu Bisht**

**Roll No.- 1918571**

**CSE-E-VI-Sem**

**Session: 2021-2022**

**Graphic Era Hill University, Dehradun**

**TABLE OF CONTENTS**

**TOPIC Page No.**

1. **ABSTRACT 1**

1. **INTRODUCTION 2**
2. **PROJECT DETAILS 3 – 4**
3. **IMPLENEMTATION 5 – 6**
4. **WORKING 7 – 9**
5. **REFERENCES 10**

**ABSTRACT**

* 1. **About the Project**

This a mini project, on embedding face recognition login feature and voice search into a website to make the website more accessible, user friendly, and secure. The website created in this project is like an online reading platform, there are various novels of different genre that are uploaded and stored in a database. These novels then can be retrieved at any moment by user, who holds the account on the website, to read.

The project covers following features:

* User Friendly GUI
* Creating an account and storing the data
* Log into the account using email and password
* Further security ensured by face recognition
* Provide user with books to read
* Voice search for easy search of books
* User friendly reading environment
  1. **Working Of Project**

This Project is created on Windows operating system and is suitable to run on any operating system on a browser. this project is suitable to run on laptop or a desktop and is not configured to run on smaller devices such as tablets or smartphones.

This steps for running the project are:

* Start the npm server
* Open the browser and use URL localhost:3000/
* The home page opens, we can go to different pages like sign in, sign up, books, etc
* Create account in sign up page
* Login using the username and password used in creating the account
* Verify yourself using face recognition
* Use the search bar either by writing or by using voice commands
* Click on the image or name of the book that you want to read
* Use logout button to log out of the website
  1. **Need of The Project**

This project can be very helpful like:

* Individually
  + Helpful in learning new technologies like note JS Mongoose database etc
  + Increases the understanding of a technology by leaps and bounds
  + A good way to get started in topics like machine learning and image processing
* Socially
  + Providing and online space to read book
  + Help in strengthening web security
  + Improving user interface and experience using voice search

**INTRODUCTION**

* 1. **Aim**

To create a Website with Face Recognition login feature and capable of using voice commands to search data.

* 1. **Introduction**
* Face recognition

Automated facial recognition was pioneered in the 1960s. Woody Bledsoe, Helen Chan Wolf, and Charles Bisson worked on using the computer to recognize human faces. Their early facial recognition project was dubbed "man-machine" because the coordinates of the facial features in a photograph had to be established by a human before they could be used by the computer for recognition.

It analyses mathematically the incoming image with some margin of error, and it verifies that the biometric data matches the person who must use the service or is requesting access to an application, system or even building.

Face recognition make use of artificial intelligence (AI) and machine learning technologies.

Major Algorithms Used Nowadays:

Local Binary Patterns Histograms (LBPH) (1996)

Fisherfaces (1997)

Scale Invariant Feature Transform (SIFT) (1999)

Speed Up Robust Features (SURF) (2006)

* Speech to text conversion

Speech to text is a speech recognition software that enables the recognition and translation of spoken language into text through computational linguistics. It is also known as speech recognition or computer speech recognition. Specific applications, tools, and devices can transcribe audio streams in real-time to display text and act on it.

Major Algorithms Used Nowadays:

Mel frequency cepstral coefficients (MFCC)

Linear prediction cepstral coefficients (LPCC).

Most popular recognition models are

Vector quantization (VQ)

Dynamic time warping (DTW)

And artificial neural network (ANN)

* 1. **Project**

This project map's the feature of face recognition and voice search into a website. The project is a website named reading eye which provides the users a platform to read books online. The log in feature in this website is secured with face recognition only after face recognition can I use the get access to the books stored in the database.

**PROJECT DETAILS**

* 1. **Technologies Used**
  + Front End: HTML5, CSS3, Bootstrap
  + Back End: Node.js, Express.js
  + Database: Mongoose Database
  + Scripting: JavaScript
  + Platform: Windows OS, Mozilla Firefox, Google Chrome, VS Code, MongoDBCompass
  + Others: JSON
  1. **External Node Module / Dependencies Used**
  + express: Used as a framework to setup custom Local Server
  + express-fileupload: Module used to Collect files send by the user and work on them
  + express-handlebars: View engine used to display dynamic web content
  + express-session: Module used for session handling
  + face-api.js: face recognition JavaScript API build on tensorflow.JS
  + hbs: View engine used to display dynamic web content
  + mongoose: Module used to facilitate connection to the Mongoose database
  + nodemon: module that allows dynamic reloading of server at any change made on any file
  1. **System Requirements**
  + Devices: Desktop, Laptops
  + Hardware: Intel Core I5 or Above
  + Operating System: Windows, or any other pc OS
  + Software: Browser supporting Scripts and recent versions on technologies used
  1. **Project Modules Completed**
* A nice-looking GUI
* Home Page to the website
* Signing up and entering data into database
* Sign in functionality
  + Email and password verification
  + Face validation and recognition
* Custom remote local server
  + Capable of maintaining connection with database
  + Entering data into the database
  + Running queries in database
  + Collecting get request and sending responses
  + Collecting post request and sending responses
  + Sending data to the client machine for dynamic web content
  + Displaying dynamic web content
* Maintain logged in and logged out status
* Provide pages like Books, Authors, Series to the signed in user
* Provide user with the search bar to facilitate searching
  + Enter the text and search
  + Use voice command to search
    - Convert audio to text
    - Use the converted text to search in the database
* Display books in a grid manner
* Display detailed information on book that is clicked
* Read the book
* Logout Button
  1. **Future Works**
* Improve the GUI
* add 2 factor authorization for better security
* Apply node session and cookies modules to provide better security for the website
  + Sessions to maintain information about the user and this session
  + Cookies used to improve the browsing speed
* Remove glitches
* Create a user account page
  + Record users reading history
  + Suggestions based on reading history
  + Option to switch between security measures like 2 factor authorisation, and face recognition
* Add more books to the database
* Provide more options other than reading like bookmarks, rating, etc.
* Evolve the website into online library
* Purchase e-books and permission to display the books to the public
* Deploy the website

**IMPLEMENTATION**

**MEATHODALOGY**

1. **Install Necessary Software’s**

Install Node.JS, MongoDBCompass, VS Code, Firefox

1. **Create a Node application**

Open Command Prompt at Selected Folder

Run command: *npm init*

1. **Install all Necessary Node Modules**

Run Command: *npm install <module-name>*

1. **Write the Script to create a Server**

app.listen( port, () => {

    console.log(`Server Running at PORT : ${port}`);

})

1. **Write Codes for displaying Front End Pages**

Create Home.hbs File

1. **Set Up Database Connection**

const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/faceVoiceNodeProject', {

    useNewUrlParser: true, useUnifiedTopology: true

}).then( () => { console.log(`Node.js - MongoDB : Connection Sucessful`);

}).catch( (err) => { console.log(`Node.js - MongoDB : Connection Un-Sucessful`);

})

1. **Set up GET and POST requests and Responses**

app.post('/authinticate', async (req, res) => {

})

app.get('/isAuthenticated', (req, res) => {

})

1. **Set up Sign Up & Sign in Pages**
2. **Set up Face Recognition Feature**

Used face-api.js, which is a tensorflow.js core api

const myInterval = setInterval(async () => {

            const detections = await faceapi.detectAllFaces(video).withFaceLandmarks().withFaceDescriptors();

            const resizedDetections = faceapi.resizeResults(detections, displaySize);

            canvas.getContext('2d').clearRect(0, 0, canvas.width, canvas.height);

            const results = resizedDetections.map((d) => {

                return faceMatcher.findBestMatch(d.descriptor);

            })

            results.forEach( (result, i) => {

                authName = result.toString();

                if( authName.substring(0, username.length ) === username ) {

                    window.location.href = "/isAuthenticated";

                }

            })

            count++;

            if (count === 100) {

                clearInterval(myInterval);

                window.location.href = "/junction";

            }

        }, 100)

1. **Set Up Voice Search Form**

Voice Search Done Using: *webkitSpeechRecognition*

var speehToText = new webkitSpeechRecognition();

    speehToText.continuous = false;

    speehToText.interimResults = false;

    speehToText.lang = 'en-US';

    speehToText.start();

    speehToText.onresult = function ( e ) {

        document.getElementById('searchBar').value = e.results[0][0].transcript;

        document.getElementById('searchBar').innerHTML = e.results[0][0].transcript;

        speehToText.stop();

    };

    speehToText.onerror = function (e) {

        speehToText.stop();

    };

1. **Display** Books

**Books are displayed is form of tiles and can be further opened** by clicking on the Image and Name

1. **Create Log—Out Functionality**

**WORKING**

1. **Home Page**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Sign Up Page**

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Sign In Page**

**Graphical user interface, website

Description automatically generated**

1. **Face Recognition Page**

**A screenshot of a person

Description automatically generated with medium confidence**

1. **Home Page After Log In**

**Graphical user interface

Description automatically generated**

1. **Books Page**

**Graphical user interface, website

Description automatically generated**

1. **Search Result Page**

**A screenshot of a computer

Description automatically generated**

1. **Book View Page**

**A screenshot of a computer

Description automatically generated**

**REFERENCES**

* Basarat Ali Syed (auth.)-Beginning Node.js-Apress Highlighted Book
* Eloquent Javascript A Modern Introduction to Programming Book
* https://www.youtube.com/
* https://www.npmjs.com/package/express
* <https://nodejs.dev/learn/>
* <https://stackoverflow.com/>
* <https://www.w3schools.com/>
* <https://www.github.com>
* https://www.mongodb.com/docs/manual/reference/