**A**

**MAJOR PROJECT REPORT ON**

# WATER QUALITY ANALYSIS USING EXTRA TREE

# CLASSIFIER

*Submitted in partial fulfillment of the requirements*

***For the award of Degree of***

## BACHELOR OF ENGINEERING

**IN**

## CSE

**Submitted By**

**K.SAI VAMSHI REDDY 245521733027**

**V.AKSHAY 245521733063**

**K.TRISHANK 245521733301**

**Under the guidance Of**

**ASSISTANT PROFESSOR**

**Mrs.V.SWAPNA**

A logo with text on it

Description automatically generated

**Department of CSE**

## KESHAV MEMORIAL ENGINEERING COLLEGE

Kachavanisingaram Village, Hyderabad, Telangana 500058.

## A logo with text on it Description automatically generatedKESHAV MEMORIAL ENGINEERING COLLEGE

A unit of Keshav Memorial Technical Education Society (KMITES)

(Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad)

D.NO. 10-TC-111, Kachavanisingaram(V), Ghatkesar (M), Medchal Malkajgiri, Telangana – 50088 Website: [www.kmec.in](http://www.kmec.in/) Email-id: [principal@kmec.in](mailto:principal@kmec.in) M: +91 40 29560274

**CERTIFICATE**

*This is to certify that the Mini project work entitled* “**WATER QUALITY ANALYSIS USING EXTRA TREE CLASSSIFIER”** *is a bonafide work carried out by* **K.SAIVAMSHIREDDY(245521733027**),**V.AKSHAY(245521733063**),**K.TRISHANK(245521733301)** of IV Year VII semester **Bachelor of Engineering** *in CSE by Osmania University, Hyderabad during the academic year* **2021-2022** *is a record of bona fide work carried out by them*. *The results embodied in this report have not been submitted to any other University or Institution for the award of any degree.*

### Internal Guide Head of Department

Mrs.V.SWAPNA Dr.CH.Rathan kumar

Assistant Professor Associate Professor

**External**

## A logo with text on it Description automatically generated KESHAV MEMORIAL ENGINEERING COLLEGE

A unit of Keshav Memorial Technical Education Society (KMITES)

(Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad)

D.NO. 10-TC-111, Kachavanisingaram(V), Ghatkesar (M), Medchal Malkajgiri, Telangana – 50088 Website: [www.kmec.in](http://www.kmec.in/) Email-id: [principal@kmec.in](mailto:principal@kmec.in) M: +91 40 29560274

# DECLARATION

We hereby declare that the Mini Project Report entitled, “**WATER QUALITY ANALYSIS USING EXTRA TREE CLASSIFIER**” submitted for the B.E degree is entirely our work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree.

**Date:**

**KOLIPALLY SAI VAMSHI REDDY 245521733027**

**VUPPALA AKSHAY 245521733063**

**K.TRISHANK 245521733301**

## A logo with text on it Description automatically generatedKESHAV MEMORIAL ENGINEERING COLLEGE

A unit of Keshav Memorial Technical Education Society (KMITES)

(Approved by AICTE, New Delhi & Affiliated to Osmania University, Hyderabad)

D.NO. 10-TC-111, Kachavanisingaram(V), Ghatkesar (M), Medchal Malkajgiri, Telangana – 50088 Website: [www.kmec.in](http://www.kmec.in/) Email-id: [principal@kmec.in](mailto:principal@kmec.in) M: +91 40 29560274

# ACKNOWLEDGEMENT

We are happy to express our deep sense of gratitude to the principal of the college  **Dr.P.V.N.PRASAD, Professor**, Keshav memorial engineering college, for having provided us with adequate facilities to pursue our project.

We would like to thank**, Dr. CH. Rathan Kumar**, **Head of the Department CSE,** Keshav memorial engineering college, for having provided the freedom to use all the facilities available in the department, especially the laboratories and the library.

We would also like to thank my internal guide**”Mrs.V.SWAPNA”** for her technical guidance & constant encouragement.

We sincerely thank our seniors and all the teaching and non-teaching staff of the Department of Computer Science & Engineering for their timely suggestions, healthy criticism and motivation during this work.

Finally, we express our immense gratitude with pleasure to the other individuals who have either directly or indirectly contributed to our need at the right time for the development and success of this work.

**I**

# ABSTRACT

Based on many studies, it is now clear that the quality of the water one consumes defines the quality of their health. To prevent waterborne illnesses such as Typhoid, Cholera, the quality of drinking water should be kept in check. This project focuses on classifying whether a given water sample is potable or not.Our task is to develop an Extra Tree Classifier model for water quality analysis to predict the contamination level of water samples. The goal is to accurately categorize water samples into different quality classes (e.g., safe, mildly polluted, highly polluted) based on various physicochemical and biological attributes. This model aims to enhance water management by providing timely and reliable assessments of water quality, aiding in prompt decision-making for public health and environmental protection The proposed methodology employs nine input parameters, pH, hardness, solid content, chloramine, sulphates, conductivity, presence of organic carbon, trihalomethanes and turbidity of all the employed algorithms, the extra trees classifier comes out to be the most precise at 74.14% accuracy with a standard deviation of 2.76% whilst a densely connected (9,32,2) neural network trained over 300 epochs yields an accuracy of 67.92% with a standard deviation of 3.65%. Using XGBoost classifier or AdaBoost classifier yields 69.64% and 56.03% accuracy respectively with similar hyperparameters. Some of the existing and currently employed water quality prediction methods include the usage of Improved Grey Relational Analysis (IGRA) algorithm, Long-Short Term Memory (LSTM) neural network and Hybrid decision tree.The Extra trees classifier outperforms the present implementations on the basics of accuracy by at least4%. Extra Trees is like Random Forest, in that it builds multiple trees and splits nodes using random subsets of features, but with two key differences: it does not bootstrap observations(meaning it samples without replacement), and nodes are split on random splits but,not best splits which goes beyond.

**Keywords:** Extra Tree Classifier,Artificial neural network, Random Forest,Improved Grey Relational Analysis,Long-Short Term Memory Neural Network

# Key Features

* Algorithmic Features
* Water Quality Analysis Features
* Data Handling Features
* Software Features
* Hardware Features

# Skills Demonstrated

* **Machine Learning**: Implementing and training an Extra Tree Classifier model for water quality analysis.
* **Data Analysis**: Preprocessing, visualizing, and analyzing water quality data.
* **Programming**: Writing clean, efficient, and well-documented Python code.
* **Problem-solving**: Applying machine learning techniques to solve a real-world problem.

# System Requirements

* **Operating System:** Windows, macOS, or Linux
* **CPU:**Intel Core i3 or equivalent
* **RAM:**4 GB (minimum), 8 GB
* **IDE:**Jupyter Notebook, PyCharm, or any other Python-compatible ID

**II**

# TABLE OF CONTENTS

**CHAPTER TITLE PAGE NO.**

**ACKNOWLEDGEMENT I**

[ABSTRACT II](#_TOC_250008)

LIST OF FIGURES V

LIST OF TABLES V

1. INTRODUCTION
   1. [PROBLEM STATEMENT **1**](#_TOC_250007)
   2. [MOTIVATION **1**](#_TOC_250006)
   3. [SCOPE **1**](#_TOC_250005)
   4. [OUTLINE **2**](#_TOC_250004)
2. LITERATURE SURVEY
   1. EXISTING SYSTEM **3**
   2. PROPOSED SYSTEM **3**
3. SOFTWARE REQUIREMENT SPECIFICATION
   1. [OVERALL DESCRIPTION **4**](#_TOC_250003)
   2. [OPERATING ENVIRONMENT **4**](#_TOC_250002)
   3. [FUNCTIONAL REQUIREMENTS **4-5**](#_TOC_250001)
   4. [NON – FUNCTIONAL REQUIREMENTS **5-7**](#_TOC_250000)

III

1. SYSTEM DESIGN
   1. USE-CASE DIAGRAM **8**
   2. CLASS DIAGRAM **9**
   3. SEQUENCE DIAGRAM **10**
2. **IMPLEMENTATION**
   1. SAMPLE CODE **11 – 47**

## TESTING

* 1. TESTING **48-50**

1. **SCREENSHOTS 51-53**
2. **CONCLUSION AND FUTURE SCOPE 54**

**BIBLIOGRAPHY 55**

**APPENDIX A:** TOOLS AND TECHNOLOGY **56**

**IV**

|  |  |  |
| --- | --- | --- |
| List of Figures |  | |
| **Figure No.** | **Name of Figure** | **Page No.** |
| 1. | Use case Diagram | **8 - 9** |
| 2. | Class Diagram | **10** |
| 3. | Sequence Diagram | **11** |

|  |  |  |
| --- | --- | --- |
| **List of Tables** |  | |
| **Table No.** | **Name of Table** | **Page No.** |
| 1. | Testcases | **48 - 50** |

# V

**CHAPTER – 1 INTRODUCTION**

## PROBLEM STATEMENT

Our task is to develop an Extra Tree Classifier model for water quality analysis to

predict the contamination level of water samples. The goal is to accurately categorize water samples into different quality classes (e.g., safe, mildly polluted, highly polluted) based on various physicochemical and biological attributes. This model aims to enhance water management by providing timely and reliable assessments of water quality, aiding in prompt decision-making for public health and environmental protection.

## MOTIVATION

Water quality assessment is crucial for safeguarding human health and the environment. Conventional methods can be time-consuming and expensive. By employing an Extra Tree Classifier, we can harness its ability to handle complex relationships in water quality data, leading to efficient and accurate classification. This approach not only streamlines analysis but also contributes to proactive measures for maintaining clean water resources..

## SCOPE

The scope for water quality classification using the Extra Trees Classifier lies in its ability to handle multiple input variables, handle noisy data, and capture complex relationships. By applying this algorithm to a dataset containing various water quality parameters, such as pH, hardness, dissolved solids, turbidity, and trihalomethanes, it can accurately classify water samples into different quality classes. This approach has the potential to assist environmental management, identify pollution sources, prioritize remediation efforts, and support decision-making processes for maintaining and improving the health of aquatic ecosystems, making it a valuable tool for monitoring agencies, researchers, and policymakers.

## OUTLINE

Here is an outline of major components that are included in a water quality analysis using extra tree classifier: data collection, feature preprocessing, Extra Tree Classifier model training,and accuracy assessment for effective water quality analysis.

# CHAPTER – 2 LITERATURE SURVEY

## EXISTING SYSTEM:

Water Quality Classification Using Machine Learning Techniques" by Ayesha Khalid et al. (2019): This study explores the application of machine learning algorithms, including decision trees, random forests, support vector machines, and artificial neural networks, for water quality classification. The research compares the performance of these algorithms on various water quality parameters and evaluates their effectiveness in differentiating water quality classes.

"Water Quality Classification Using Machine Learning Techniques: A Comparative Study" by Charu Gupta et al. (2020): This research presents a comparative analysis of different machine learning algorithms, such as decision trees, random forests, k-nearest neighbors, and naïve Bayes, for water quality classification. The study evaluates the performance of these algorithms on a dataset containing multiple water quality parameters and provides insights into their accuracy and efficiency.

"Water Quality Classification Using Machine Learning Algorithms" by Subha Gomathy et al. (2021): This study investigates the use of machine learning algorithms, including decision trees, random forests, and support vector machines, for water quality classification. The research compares the classification performance of these algorithms on a dataset comprising parameters like pH, dissolved oxygen, and biochemical oxygen demand, and analyzes their effectiveness in water quality assessment.

"Water Quality Classification using Machine Learning Algorithms: A Case Study in the Amazon River Basin" by Felipe A. S. Pereira et al. (2021): This research focuses on water quality classification in the Amazon River Basin using machine learning algorithms. The study applies decision trees, random forests, and gradient boosting algorithms to classify water samples based on multiple parameters, including turbidity, chlorophyll-a, and conductivity.

## PROPOSED SYSTEM:

The proposed system for water quality classification using the Extra Trees Classifier aims to integrate real-time data collection from multiple sources, such as sensors and satellite imagery, with the robust classification capabilities of the Extra Trees algorithm. The system will leverage the ensemble learning approach of Extra Trees to accurately classify water samples into various quality categories, providing timely insights for water resource management and decision-making. Additionally, the system will incorporate features like data visualization, anomaly detection, and automated alert mechanisms to enhance the usability and practicality of the classification results.

Advantages:

 Robust Handling of Noisy Data: The Extra Trees Classifier is known for its robustness in handling noisy and incomplete data. In water quality assessment, where data can be subject to measurement errors or missing values, the Extra Trees Classifier can effectively handle such challenges and still provide accurate classification results. It employs an ensemble learning approach that combines multiple decision trees, which helps reduce the impact of noise and outliers in the data.

 Efficient Feature Selection: The Extra Trees Classifier performs automatic feature selection, allowing it to identify the most relevant features for water quality classification. This eliminates the need for manual feature engineering and reduces the risk of including irrelevant or redundant variables. By automatically selecting informative features, the Extra Trees Classifier can improve the efficiency and effectiveness of the classification process, leading to more accurate and reliable water quality classification results.

# CHAPTER - 3

**SOFTWARE REQUIREMENTS SPECIFICATION**

### Overall Description:

This SRS is an overview of the whole project scenario. This document is to present a detailed description of the course management system. It will explain the purpose and features of the system, the interfaces of the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both stakeholders and developers of the system.

### Operating Environment:

***Software Requirements:***

Operating System : Windows 7 (Min)

Front End : React JS

Back End : Flask

Coding Language : Python

***Hardware Requirements:***

Processor - Pentium –IV

RAM - 4 GB (min)

Hard Disk - 20 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

### Functional Requirements:

**User Functionality:**

* Users should be able to input water quality attributes via an intuitive interface for automated classification using the Extra Tree Classifier..
* The system should present clear and interpretable results, indicating the predicted water quality classes and associated confidence levels, assisting users in informed decision-making.

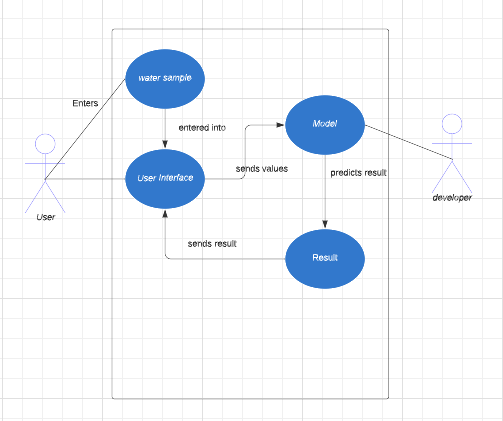
### Non-Functional Requirements:

* + 1. **Performance Requirements:**
* The Extra Tree Classifier should process water quality data and generate predictions within seconds for efficient real-time analysis.
* The system should achieve a classification accuracy of at least 90% on validation datasets to ensure reliable contamination level predictions.
* The application's response time to user inputs should be under 1 second to provide a seamless and responsive user experience.
* The system should be capable of handling a minimum of 1000 water quality samples per hour to accommodate varying analysis loads.
  + 1. **Safety Requirements:**
* The system must ensure the secure storage and transmission of sensitive water quality data to prevent unauthorized access or breaches.
* Clear error messages should be provided to users, guiding them in addressing any anomalies during data input or analysis.
* The system should adhere to data privacy regulations, protecting individuals' personal information used in the analysis.
* Regular monitoring and maintenance should be conducted to promptly address any potential security vulnerabilities and ensure the system's overall reliability.
  + 1. **Security Requirements:**
* User authentication and access control mechanisms must be implemented to restrict system usage to authorized personnel only.
* Data encryption during storage and transmission should be enforced to safeguard sensitive water quality information.
* Regular security audits and vulnerability assessments should be conducted to identify and mitigate potential threats.
* The system must have provisions for logging and tracking user activities to detect any unauthorized or suspicious behavior.
  + 1. **Software Quality Attributes:**
* Accuracy: The classifier should provide precise water quality predictions, reflecting the true contamination levels.
* Reliability: The system must consistently perform well over time, delivering consistent results across different datasets.
* Usability: The user interface should be intuitive, allowing users to easily input data and interpret classification outcomes.
* Scalability: The software should efficiently handle varying data volumes, adapting to increased analysis demands without compromising performance.

**CHAPTER-4**

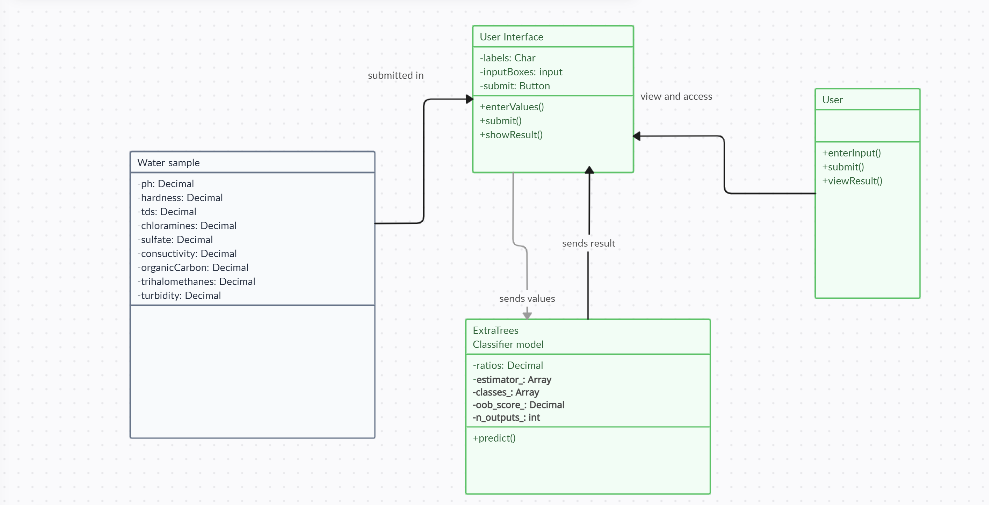
**SYSTEM DESIGN**

### Use case Diagram:

****

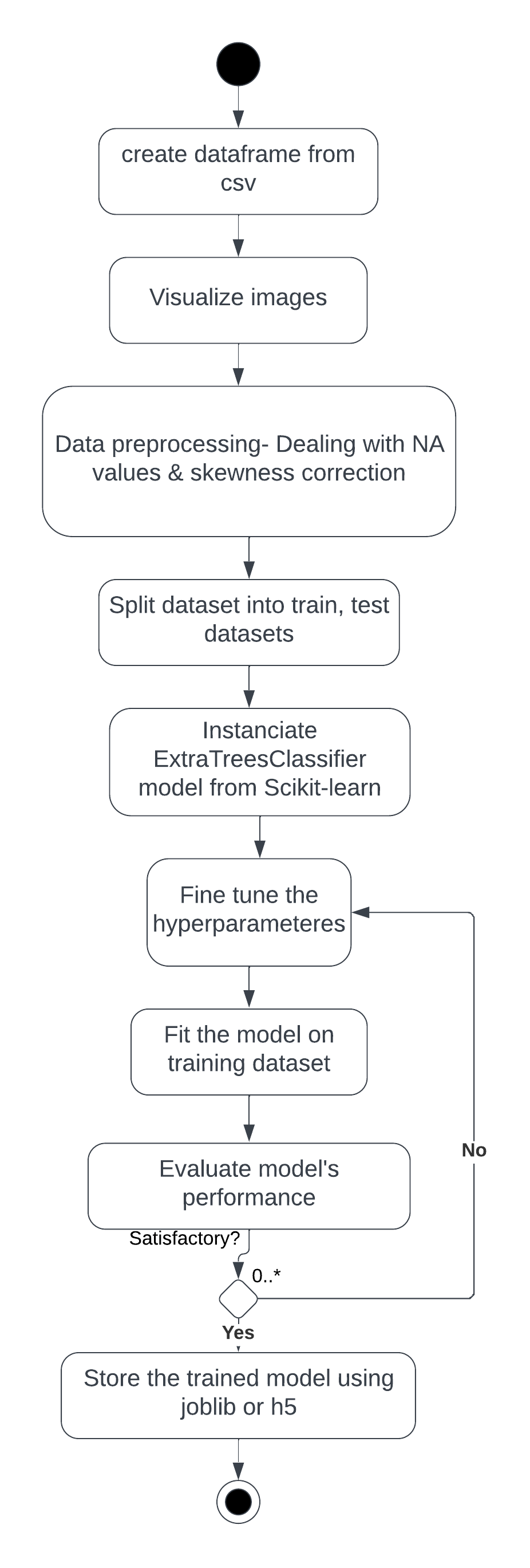
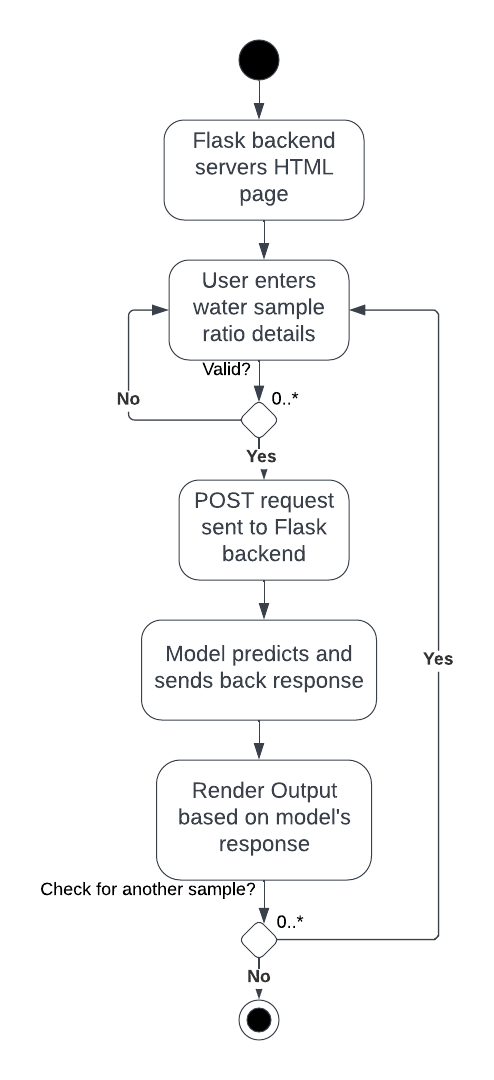
**Fig 4.1: Use case diagram**

### Class Diagram:

****

**Fig.4.3: Class diagram**

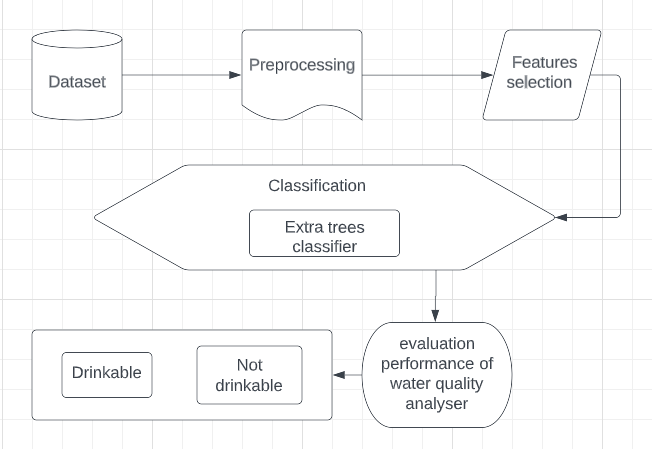
### Activity Diagram:

**** 

**Fig4.4: Activity**

**5.SYSTEM DESIGN**

**System Architecture**

****

**5. Implementation and Results**

**5.1 Languages and Technology used**

1. REACT JS
2. CSS
3. PYTHON – SCIKIT LEARN
4. RESTFUL API

**5.2Algorithms Used**

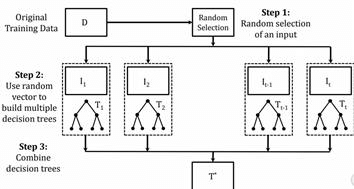
**EXTRA TREES CLASSIFIER**

The Extra Trees Classifier is an ensemble learning algorithm used for classification tasks. It belongs to the family of decision tree algorithms and is an extension of the Random Forest algorithm. The key idea behind the Extra Trees Classifier is to create a large number of decision trees, each trained on a random subset of features and using random splits at each node.

Here are some key characteristics and advantages of the Extra Trees Classifier:

* Random Feature Subsets: Unlike traditional decision trees, which choose the best feature to split a node, the Extra Trees Classifier selects random subsets of features to create diverse decision trees. This randomization helps to reduce the correlation among the trees and improves the robustness against overfitting.
* Random Splits: In addition to random feature subsets, the Extra Trees Classifier also employs random splits at each node. It randomly selects the splitting point instead of optimizing the split based on information gain or Gini impurity. This randomness further enhances the diversity of the individual trees.
* Aggregation of Predictions: During the classification phase, predictions from all individual trees are aggregated to make the final prediction. This can be done through majority voting (for classification tasks) or averaging (for regression tasks). The ensemble nature of the Extra Trees Classifier helps to improve the overall accuracy and generalization of the model.
* Robustness to Noisy Data: The Extra Trees Classifier is known for its robustness in handling noisy or incomplete data. By using multiple trees with random feature subsets, it can effectively mitigate the impact of noisy or irrelevant features, leading to more reliable predictions.
* Parallelizable and Scalable: The training process of the Extra Trees Classifier can be parallelized, allowing for faster training on large datasets. Moreover, as each tree is independently trained, the algorithm is highly scalable and can handle datasets with a large number of samples and features.
* Feature Importance: The Extra Trees Classifier can provide an estimate of feature importance, indicating which features are most relevant for classification. This information can be valuable for understanding the underlying relationships in the data and selecting the most informative features.

The Extra Trees Classifier is a versatile and powerful algorithm suitable for a wide range of classification tasks, including water quality classification. It offers robustness, scalability, and improved accuracy, making it a valuable tool in various domains where accurate classification is required.

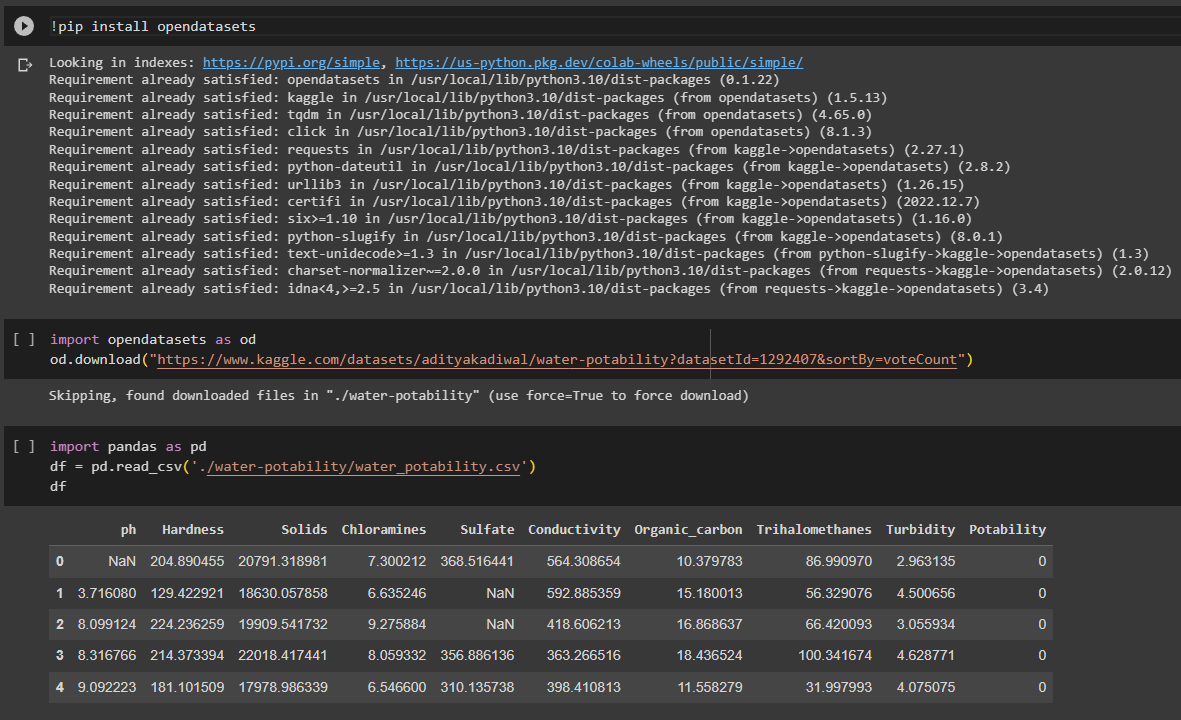


**Fig-7**

**5.3 Sample Code**

**Extra trees classifier model:**

* Loading the dataset

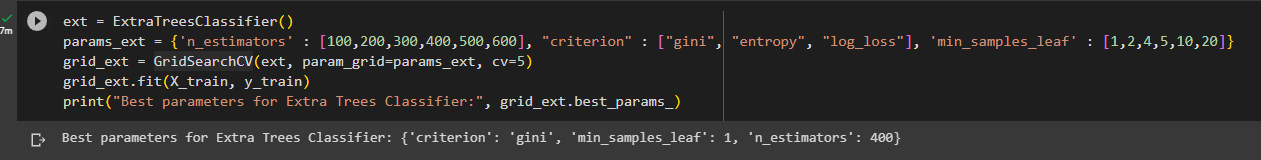


* Preprocessing



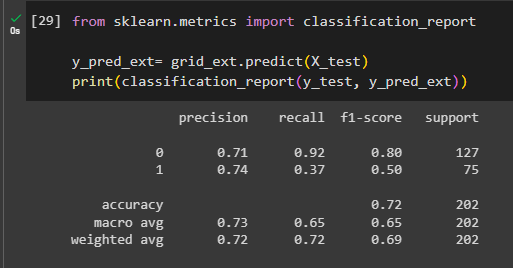
**Fig-9**

* Find the best hyperparameters and training our model

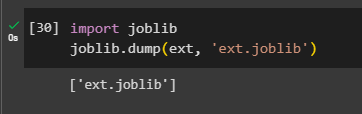


**Fig-10**

* Evalutating our model



* Saving our model



**Fig-12**

**Index.js:**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import { QueryClient, QueryClientProvider } from "react-query";

import App from './App';

import {

createBrowserRouter,

RouterProvider,

} from "react-router-dom";

import Result from './Pages/Result';

import SliderPredict from './Pages/SliderPredict';

const router = createBrowserRouter([

{

path:"/",

element: <App />

},

{

path: "/result",

element: <Result />

},

{

path: "/new",

element: <New />

}

])

const queryClient = new QueryClient();

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<QueryClientProvider client={queryClient}>

<RouterProvider router={router}/>

</QueryClientProvider>

);

}

});

**App.js:**

import "./App.css";

import Button from "@mui/material/Button";

import ArrowForwardIcon from "@mui/icons-material/ArrowForward";

import { Navigate, useNavigate } from "react-router-dom";

function App() {

const navigate = useNavigate();

return (

<div>

<div class="header">

<div class="inner-header flex">

<h1>Water Quality Analyser</h1>

</div>

<Button

style={{ color: "white" }}

onClick={() => {

navigate("/predict");

}}

endIcon={<ArrowForwardIcon />}

>

Predict

</Button>

<div>

<svg

class="waves"

xmlns="http://www.w3.org/2000/svg"

xmlnsXlink="http://www.w3.org/1999/xlink"

viewBox="0 24 150 28"

preserveAspectRatio="none"

shape-rendering="auto"

>

<defs>

<path

id="gentle-wave"

d="M-160 44c30 0 58-18 88-18s 58 18 88 18 58-18 88-18 58 18 88 18 v44h-352z"

/>

</defs>

<g class="parallax">

<use

xlinkHref="#gentle-wave"

x="48"

y="0"

fill="rgba(255,255,255,0.7"

/>

<use

xlinkHref="#gentle-wave"

x="48"

y="3"

fill="rgba(255,255,255,0.5)"

/>

<use

xlinkHref="#gentle-wave"

x="48"

y="5"

fill="rgba(255,255,255,0.3)"

/>

<use xlinkHref="#gentle-wave" x="48" y="7" fill="#fff" />

</g>

</svg>

</div>

</div>

<div class="content flex">

<p></p>

</div>

</div>

);

}

export default App;

</div>

</body>

</html>

**Result.jsx:**

import { Card, CardContent, Typography } from "@mui/material";

import React, { useEffect, useState } from "react";

import { useLocation } from "react-router-dom";

import TaskAltIcon from "@mui/icons-material/TaskAlt";

export default function Result() {

    const { state: values } = useLocation();

    const [answer, setAnswer] = useState();

    const [word, setWord] = useState("");

    useEffect(() => {

        fetch("http://127.0.0.1:5000/predict", {

            method: "POST",

            headers: {

                "Content-Type": "application/json",

            },

            body: JSON.stringify({ ratios: [7.080794504, 196.369496, 22014.09253, 7.122276793, 333.7757766, 426.2051107, 14.28497025, 66.39629295, 3.96678617] }),

        })

            .then((res) => res.json())

            .then((data) => {

                console.log(data, typeof data);

                setAnswer(data);

                if (data == 0) {

                    setWord("The water sample is not drinkable!");

                } else {

                    setWord("The water sample is drinkable!");

                }

            });

    });

    return (

        <div

            style={{

                position: "absolute",

                left: "50%",

                top: "50%",

                transform: "translate(-50%, -50%)",

            }}

        >

            <Typography variant="h4" style={{color : word==="The water sample is not drinkable!" ? "red" : "green"}}>{word}</Typography>

        </div>

    );

}

                        ),

                        Hardness: Yup.number("Please enter a number").required(

                            "This is a requried field"

                        ),

                        "Total dissolved Solids": Yup.number(

                            "Please enter a number"

                        ).required("This is a requried field"),

                        Chloramines: Yup.number(

                            "Please enter a number"

                        ).required("This is a requried field"),

                        Sulphates: Yup.number("Please enter a number").required(

                            "This is a requried field"

                        ),

                        Conductivity: Yup.number(

                            "Please enter a number"

                        ).required("This is a requried field"),

                        "Organic Carbon": Yup.number(

                            "Please enter a number"

                        ).required("This is a requried field"),

                        Trihalomethanes: Yup.number(

                            "Please enter a number"

                        ).required("This is a requried field"),

                        Turbidity: Yup.number("Please enter a number").required(

                            "This is a requried field"

                        ),

                    })}

                >

                    {(props) => {

                        return (

                            <form onSubmit={props.handleSubmit}>

                                <div

                                    style={{

                                        display: "flex",

                                        flexWrap: "wrap",

                                        justifyContent: "center",

                                    }}

                                >

                                    {data.map((item) => (

                                        <ParamCard

                                            {...props}

                                            {...item}

                                            key={item.name}

                                        />

                                    ))}

                                  ))}

                                </div>

                                <Button

                                    type="submit"

                                    variant="outlined"

                                    style={{

                                        marginTop: "10px",

                                        float: "right",

                                        marginRight: "45px",

                                    }}

                                >

                                    Submit

                                </Button>

                            </form>

                        );

                    }}

                </Formik>

            </Container>

        </React.Fragment>

    );

}

**Result.jsx:**

import { Card, CardContent, Typography } from "@mui/material";

import React, { useEffect, useState } from "react";

import { useLocation } from "react-router-dom";

import TaskAltIcon from "@mui/icons-material/TaskAlt";

export default function Result() {

    const { state: values } = useLocation();

    const [answer, setAnswer] = useState();

    const [word, setWord] = useState("");

    useEffect(() => {

        fetch("http://127.0.0.1:5000/predict", {

            method: "POST",

            headers: {

                "Content-Type": "application/json",

            },

            body: JSON.stringify({ ratios: [7.080794504, 196.369496, 22014.09253, 7.122276793, 333.7757766, 426.2051107, 14.28497025, 66.39629295, 3.96678617] }),

        })

            .then((res) => res.json())

            .then((data) => {

                console.log(data, typeof data);

                setAnswer(data);

                if (data == 0) {

                    setWord("The water sample is not drinkable!");

                } else {

                    setWord("The water sample is drinkable!");

                }

            });

    });

    return (

        <div

            style={{

                position: "absolute",

                left: "50%",

                top: "50%",

                transform: "translate(-50%, -50%)",

            }}

        >

            <Typography variant="h4" style={{color : word==="The water sample is not drinkable!" ? "red" : "green"}}>{word}</Typography>

        </div>

    );

}

### Comment\_frame.php

<html>

<head>

<title></title>

<link rel="stylesheet" type="text/css" href="assets/css/style.css">

</head>

<body>

<style type="text/css">

\* {

font-size: 12px;

font-family: Arial, Helvetica, Sans-serif;

}

</style>

<?php

require 'config/config.php'; include("includes/classes/User.php"); include("includes/classes/Post.php"); include("includes/classes/Notification.php");

if (isset($\_SESSION['username'])) {

$userLoggedIn = $\_SESSION['username'];

$user\_details\_query = mysqli\_query($con, "SELECT \* FROM users WHERE username='$userLoggedIn'");

$user = mysqli\_fetch\_array($user\_details\_query);

}

else {

header("Location: register.php");

}

?>

<script>

function toggle() {

var element = document.getElementById("comment\_section");

if(element.style.display == "block") element.style.display = "none";

else

element.style.display = "block";

}

</script>

<?php

//Get id of post if(isset($\_GET['post\_id'])) {

$post\_id = $\_GET['post\_id'];

}

$user\_query = mysqli\_query($con, "SELECT added\_by, user\_to FROM posts WHERE id='$post\_id'");

$row = mysqli\_fetch\_array($user\_query);

$posted\_to = $row['added\_by'];

$user\_to = $row['user\_to'];

if(isset($\_POST['postComment' . $post\_id])) {

$post\_body = $\_POST['post\_body'];

$post\_body = mysqli\_escape\_string($con, $post\_body);

$date\_time\_now = date("Y-m-d H:i:s");

$insert\_post = mysqli\_query($con, "INSERT INTO comments VALUES ('', '$post\_body', '$userLoggedIn', '$posted\_to', '$date\_time\_now', 'no', '$post\_id')");

if($posted\_to != $userLoggedIn) {

$notification = new Notification($con, $userLoggedIn);

$notification->insertNotification($post\_id, $posted\_to,

"comment");

}

if($user\_to != 'none' && $user\_to != $userLoggedIn) {

$notification = new Notification($con, $userLoggedIn);

$notification->insertNotification($post\_id, $user\_to, "profile\_comment");

}

$get\_commenters = mysqli\_query($con, "SELECT \* FROM comments WHERE post\_id='$post\_id'");

$notified\_users = array();

while($row = mysqli\_fetch\_array($get\_commenters)) {

$user\_to

if($row['posted\_by'] != $posted\_to && $row['posted\_by'] !=

&& $row['posted\_by'] != $userLoggedIn &&

!in\_array($row['posted\_by'], $notified\_users)) {

$notification = new Notification($con, $userLoggedIn);

$notification->insertNotification($post\_id, $row['posted\_by'], "comment\_non\_owner");

array\_push($notified\_users, $row['posted\_by']);

}

}

echo "<p>Comment Posted! </p>";

}

?>

<form action="comment\_frame.php?post\_id=<?php echo $post\_id; ?>" id="comment\_form" name="postComment<?php echo $post\_id; ?>" method="POST">

<textarea name="post\_body"></textarea>

<input type="submit" name="postComment<?php echo $post\_id; ?>" value="Post">

</form>

<!-- Load comments -->

<?php

$get\_comments = mysqli\_query($con, "SELECT \* FROM comments WHERE post\_id='$post\_id' ORDER BY id ASC");

$count = mysqli\_num\_rows($get\_comments); if($count != 0) {

while($comment = mysqli\_fetch\_array($get\_comments)) {

$comment\_body = $comment['post\_body'];

$posted\_to = $comment['posted\_to'];

$posted\_by = $comment['posted\_by'];

$date\_added = $comment['date\_added'];

$removed = $comment['removed'];

//Timeframe

$date\_time\_now = date("Y-m-d H:i:s");

dates

$start\_date = new DateTime($date\_added); //Time of post

$end\_date = new DateTime($date\_time\_now); //Current time

$interval = $start\_date->diff($end\_date); //Difference between

if($interval->y >= 1) { if($interval == 1)

$time\_message = $interval->y . " year ago"; //1 year ago

else

}

$time\_message = $interval->y . " years ago"; //1+ year ago

else if ($interval->m >= 1) { if($interval->d == 0) {

$days = " ago";

}

else if($interval->d == 1) {

$days = $interval->d . " day ago";

}

else {

$days = $interval->d . " days ago";

}

if($interval->m == 1) {

$time\_message = $interval->m . " month". $days;

}

else {

$time\_message = $interval->m . " months". $days;

}

}

else if($interval->d >= 1) { if($interval->d == 1) {

$time\_message = "Yesterday";

}

else {

$time\_message = $interval->d . " days ago";

}

}

else if($interval->h >= 1) { if($interval->h == 1) {

$time\_message = $interval->h . " hour ago";

}

else {

$time\_message = $interval->h . " hours ago";

}

}

else if($interval->i >= 1) {

if($interval->i == 1) {

$time\_message = $interval->i . " minute ago";

}

else {

$time\_message = $interval->i . " minutes ago";

}

}

else {

if($interval->s < 30) {

$time\_message = "Just now";

}

else {

$time\_message = $interval->s . " seconds ago";

}

}

$user\_obj = new User($con, $posted\_by);

?>

<div class="comment\_section">

<a href="<?php echo $posted\_by?>" target="\_parent"><img src="<?php echo $user\_obj->getProfilePic();?>" title="<?php echo $posted\_by;

?>" style="float:left;" height="30"></a>

<a href="<?php echo $posted\_by?>" target="\_parent"> <b> <?php echo $user\_obj->getFirstAndLastName(); ?> </b></a>

&nbsp;&nbsp;&nbsp;&nbsp; <?php echo $time\_message . "<br>" .

$comment\_body; ?>

<hr>

</div>

<?php

}

}

else {

echo "<center><br><br>No Comments to Show!</center>";

}

?>

</body>

</html>

### like.php

<html>

<head>

<title></title>

<link rel="stylesheet" type="text/css" href="assets/css/style.css">

</head>

<body>

<style type="text/css">

\* {

font-family: Arial, Helvetica, Sans-serif;

}

body {

background-color: #fff;

}

form {

position: absolute; top: 0;

}

</style>

<?php

require 'config/config.php'; include("includes/classes/User.php"); include("includes/classes/Post.php"); include("includes/classes/Notification.php");

if (isset($\_SESSION['username'])) {

$userLoggedIn = $\_SESSION['username'];

$user\_details\_query = mysqli\_query($con, "SELECT \* FROM users WHERE username='$userLoggedIn'");

$user = mysqli\_fetch\_array($user\_details\_query);

}

else {

header("Location: register.php");

}

//Get id of post if(isset($\_GET['post\_id'])) {

$post\_id = $\_GET['post\_id'];

}

$get\_likes = mysqli\_query($con, "SELECT likes, added\_by FROM posts WHERE id='$post\_id'");

$row = mysqli\_fetch\_array($get\_likes);

$total\_likes = $row['likes'];

$user\_liked = $row['added\_by'];

$user\_details\_query = mysqli\_query($con, "SELECT \* FROM users WHERE username='$user\_liked'");

$row = mysqli\_fetch\_array($user\_details\_query);

$total\_user\_likes = $row['num\_likes'];

//Like button if(isset($\_POST['like\_button'])) {

$total\_likes++;

$query = mysqli\_query($con, "UPDATE posts SET likes='$total\_likes' WHERE id='$post\_id'");

$total\_user\_likes++;

$user\_likes = mysqli\_query($con, "UPDATE users SET num\_likes='$total\_user\_likes' WHERE username='$user\_liked'");

$insert\_user = mysqli\_query($con, "INSERT INTO likes VALUES('', '$userLoggedIn', '$post\_id')");

//Insert Notification if($user\_liked != $userLoggedIn) {

$notification = new Notification($con, $userLoggedIn);

$notification->insertNotification($post\_id, $user\_liked, "like");

}

}

//Unlike button if(isset($\_POST['unlike\_button'])) {

$total\_likes--;

$query = mysqli\_query($con, "UPDATE posts SET likes='$total\_likes' WHERE id='$post\_id'");

$total\_user\_likes--;

$user\_likes = mysqli\_query($con, "UPDATE users SET num\_likes='$total\_user\_likes' WHERE username='$user\_liked'");

$insert\_user = mysqli\_query($con, "DELETE FROM likes WHERE username='$userLoggedIn' AND post\_id='$post\_id'");

}

//Check for previous likes

$check\_query = mysqli\_query($con, "SELECT \* FROM likes WHERE username='$userLoggedIn' AND post\_id='$post\_id'");

$num\_rows = mysqli\_num\_rows($check\_query);

if($num\_rows > 0) {

echo '<form action="like.php?post\_id=' . $post\_id . '" method="POST">

value="Unlike">

<input type="submit" class="comment\_like" name="unlike\_button"

<div class="like\_value">

'. $total\_likes .' Likes

</div>

</form>

';

}

else {

echo '<form action="like.php?post\_id=' . $post\_id . '" method="POST">

<input type="submit" class="comment\_like" name="like\_button"

value="Like">

<div class="like\_value">

'. $total\_likes .' Likes

</div>

</form>

';

}

?>

</body>

</html>

### message.php

<?php include("includes/header.php");

$message\_obj = new Message($con, $userLoggedIn); if(isset($\_GET['u']))

$user\_to = $\_GET['u'];

else {

$user\_to = $message\_obj->getMostRecentUser(); if($user\_to == false)

$user\_to = 'new';

}

if($user\_to != "new")

$user\_to\_obj = new User($con, $user\_to); if(isset($\_POST['post\_message'])) {

if(isset($\_POST['message\_body'])) {

$body = mysqli\_real\_escape\_string($con, $\_POST['message\_body']);

$date = date("Y-m-d H:i:s");

$message\_obj->sendMessage($user\_to, $body, $date);

}

}

?>

<div class="user\_details column" >

<a href="<?php echo $userLoggedIn; ?>"> <img src="<?php echo

$user['profile\_pic']; ?>"> </a>

<div class="user\_details\_left\_right">

<a href="<?php echo $userLoggedIn; ?>">

<?php

echo $user['first\_name'] . " " . $user['last\_name'];

?>

</a>

<br>

<?php echo "Posts: " . $user['num\_posts']. "<br>"; echo "Likes: " . $user['num\_likes'];

?>

</div>

</div>

<div class="main\_column column" id="main\_column">

<?php

if($user\_to != "new"){

echo "<h4>You and <a href='$user\_to'>" . $user\_to\_obj-

>getFirstAndLastName() . "</a></h4><hr><br>";

echo "<div class='loaded\_messages' id='scroll\_messages'>"; echo $message\_obj->getMessages($user\_to);

echo "</div>";

}

else {

echo "<h4>New Message</h4>";

}

?>

<div class="message\_post">

<form action="" method="POST">

<?php

if($user\_to == "new") {

echo "Select the friend you would like to message

<br><br>";

?>

To: <input type='text' onkeyup='getUsers(this.value,

"<?php echo $userLoggedIn; ?>")' name='q' placeholder='Name' autocomplete='off' id='seach\_text\_input'>

<?php

echo "<div class='results'></div>";

}

else {

echo "<textarea name='message\_body' id='message\_textarea' placeholder='Write your message ...'></textarea>";

echo "<input type='submit' name='post\_message' class='info' id='message\_submit' value='Send'>";

}

?>

</form>

</div>

<script>

var div = document.getElementById("scroll\_messages"); div.scrollTop = div.scrollHeight;

</script>

</div>

<div class="user\_details column" id="conversations">

<h4>Conversations</h4>

<div class="loaded\_conversations">

<?php echo $message\_obj->getConvos(); ?>

</div>

<br>

<a href="messages.php?u=new">New Message</a>

</div>

### profile.php

<?php

include("includes/header.php");

$message\_obj = new Message($con,$userLoggedIn); if (isset($\_GET['profile\_username'])) {

$username = $\_GET['profile\_username'];

$user\_details\_query = mysqli\_query($con, "SELECT \* FROM users WHERE username='$username'");

$user\_array = mysqli\_fetch\_array($user\_details\_query);

$num\_friends = (substr\_count($user\_array['friend\_array'], ",")) - 1;

}

if (isset($\_POST['remove\_friend'])) {

$user = new User($con, $userLoggedIn);

$user->removeFriend($username);

}

if (isset($\_POST['add\_friend'])) {

$user = new User($con, $userLoggedIn);

$user->sendRequest($username);

}

if (isset($\_POST['respond\_request'])) { header("Location: requests.php");

}

if(isset($\_POST['post\_message'])){ if(isset($\_POST['message\_body'])){

$body = mysqli\_real\_escape\_string($con,$\_POST['message\_body']);

$date = date("Y-m-d H:i:s");

$message\_obj -> sendMessage($username, $body, $date);

}

$link = '#profileTabs a[href="#messages\_div"]'; echo "<script>

$(function() {

$('" . $link . "').tab('show');

});

</script>";

}

?>

<style type="text/css">

.wrapper {

margin-left: 0px; padding-left: 0px;

}

</style>

<div class="profile\_left">

<img src="<?php echo $user\_array['profile\_pic']; ?>">

<div class="profile\_info">

<p>

<?php echo "Name: " . $user\_array['first\_name'] . " " .

$user\_array['last\_name']; ?>

</p>

<p>

<?php echo "Posts: " . $user\_array['num\_posts']; ?>

</p>

<p>

<?php echo "Likes: " . $user\_array['num\_likes']; ?>

</p>

<p>

<?php echo "Friends: " . $num\_friends ?>

</p>

</div>

<form action="<?php echo $username; ?>" method="POST">

<?php

$profile\_user\_obj = new User($con, $username); if ($profile\_user\_obj->isClosed()) {

header("Location: user\_closed.php");

}

$logged\_in\_user\_obj = new User($con, $userLoggedIn); if ($userLoggedIn != $username) {

if ($logged\_in\_user\_obj->isFriend($username)) {

echo '<input type="submit" name="remove\_friend" class="danger" value="Remove Friend"><br>';

} else if ($logged\_in\_user\_obj->didReceiveRequest($username)) { echo '<input type="submit" name="respond\_request" class="warning"

value="Respond to Request"><br>';

} else if ($logged\_in\_user\_obj->didSendRequest($username)) {

echo '<input type="submit" name="" class="default" value="Request Sent"><br>';

} else

echo '<input type="submit" name="add\_friend" class="success" value="Add Friend"><br>';

}

?>

</form>

<input type="submit" class="deep\_blue" data-toggle="modal" data- target="#post\_form" value="Post Something">

<?php

if ($userLoggedIn != $username) {

echo '<div class="profile\_info\_bottom">';

echo $logged\_in\_user\_obj->getMutualFriends($username) . " Mutual friends"; echo '</div>';

}

?>

</div>

<div class="profile\_main\_column column">

<ul class="nav nav-tabs" role="tablist" id="profileTabs">

<li role="presentation" class="active"><a href="#newsfeed\_div" aria- controls="newsfeed\_div" role="tab"

data-toggle="tab">Newsfeed</a></li>

<li role="presentation"><a href="#messages\_div" aria- controls="messages\_div" role="tab"

data-toggle="tab">Messages</a></li>

</ul>

<div class="tab-content">

<div role="tabpanel" class="tab-pane fade in active" id="newsfeed\_div">

<div class="posts\_area"></div>

<img id="loading" src="assets/images/icons/loading.gif">

</div>

<div role="tabpanel" class="tab-pane fade" id="messages\_div">

<?php

echo "<h4>You and <a href='" . $username . "'>" . $profile\_user\_obj-

>getFirstAndLastName() . "</a></h4><hr><br>";

echo "<div class='loaded\_messages' id='scroll\_messages'>";

echo $message\_obj->getMessages($username); echo "</div>";

?>

<div class="message\_post">

<form action="" method="POST">

<textarea name='message\_body' id='message\_textarea' placeholder='Write your message ...'></textarea>

<input type='submit' name='post\_message' class='info' id='message\_submit' value='Send'>

</form>

</div>

<script>

var div = document.getElementById("scroll\_messages"); div.scrollTop = div.scrollHeight;

</script>

</div>

</div>

<!-- Modal -->

<div class="modal fade" id="post\_form" tabindex="-1" role="dialog" aria- labelledby="postModalLabel"

aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria- label="Close"><span

aria-hidden="true">&times;</span></button>

<h4 class="modal-title" id="postModalLabel">Post something!</h4>

</div>

<div class="modal-body">

<p>This will appear on the user's profile page and also their newsfeed for your friends to see!</p>

<form class="profile\_post" action="" method="POST">

<div class="form-group">

<textarea class="form-control" name="post\_body"></textarea>

<input type="hidden" name="user\_from" value="<?php echo

$userLoggedIn; ?>">

<input type="hidden" name="user\_to" value="<?php echo $username;

?>">

</div>

</form>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data- dismiss="modal">Close</button>

<button type="button" class="btn btn-primary" name="post\_button" id="submit\_profile\_post">Post</button>

</div>

</div>

</div>

</div>

<script>

var userLoggedIn = '<?php echo $userLoggedIn; ?>'; var profileUsername = '<?php echo $username; ?>';

$(document).ready(function () {

$('#loading').show();

//Original ajax request for loading first posts

$.ajax({

url: "includes/handlers/ajax\_load\_profile\_posts.php", type: "POST",

data: "page=1&userLoggedIn=" + userLoggedIn + "&profileUsername=" + profileUsername,

cache: false,

success: function (data) {

$('#loading').hide();

$('.posts\_area').html(data);

}

});

$(window).scroll(function () {

var height = $('.posts\_area').height(); //Div containing posts var scroll\_top = $(this).scrollTop();

var page = $('.posts\_area').find('.nextPage').val();

var noMorePosts = $('.posts\_area').find('.noMorePosts').val();

if ((document.body.scrollHeight == document.body.scrollTop + window.innerHeight) && noMorePosts == 'false') {

$('#loading').show();

var ajaxReq = $.ajax({

url: "includes/handlers/ajax\_load\_profile\_posts.php", type: "POST",

data: "page=" + page + "&userLoggedIn=" + userLoggedIn + "&profileUsername=" + profileUsername,

cache: false,

.nextpage

success: function (response) {

$('.posts\_area').find('.nextPage').remove(); //Removes current

$('.posts\_area').find('.noMorePosts').remove(); //Removes

current .nextpage

$('#loading').hide();

$('.posts\_area').append(response);

}

});

} //End if return false;

}); //End (window).scroll(function())

});

</script>

</div>

</body>

</html>

### register.php

<?php

require 'config/config.php';

require 'includes/form\_handlers/register\_handler.php'; require 'includes/form\_handlers/login\_handler.php';

?>

<html>

<head>

<title>Welcome to The Social Net!</title>

<link rel="stylesheet" type="text/css" href="assets/css/register\_style.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></scrip t>

<script src="assets/js/register.js"></script>

</head>

<body>

<?php if(isset($\_POST['register\_button'])){

echo '

<script>

$(document).ready(function(){

$("#first").hide();

$("#second").show();

});

</script>

';

}

?>

<div class="wrapper">

<div class="login\_box">

<div class="login\_header">

<h1>The Social Net</h1> Login or signup

</div>

<div id="first">

<form action="register.php" method="POST">

<input type="email" name="log\_email" placeholder="Email Address" value="<?php

if (isset($\_SESSION['log\_email'])) { echo $\_SESSION['log\_email'];

}

?>" required>

<br>

<input type="password" name="log\_password" placeholder="Password">

<br>

<?php if (in\_array("Email or password was incorrect<br>",

$error\_array))

Register now!</a>

echo "Email or password was incorrect<br>"; ?>

<input type="submit" name="login\_button" value="Login">

<br>

<a href="#" id="signup" class="signup">Need an account?

</form>

</div>

<div id="second">

<form action="register.php" method="POST">

<input type="text" name="reg\_fname" placeholder="First

Name" value="<?php

if (isset($\_SESSION['reg\_fname'])) { echo $\_SESSION['reg\_fname'];

}

?>" required>

<br>

<?php if (in\_array("Your first name must be between 2 and 25 characters<br>", $error\_array))

echo "Your first name must be between 2 and 25

characters<br>"; ?>

Name" value="<?php

<input type="text" name="reg\_lname" placeholder="Last

if (isset($\_SESSION['reg\_lname'])) { echo $\_SESSION['reg\_lname'];

}

?>" required>

<br>

<?php if (in\_array("Your last name must be between 2 and 25 characters<br>", $error\_array))

echo "Your last name must be between 2 and 25

characters<br>"; ?>

value="<?php

<input type="email" name="reg\_email" placeholder="Email" if (isset($\_SESSION['reg\_email'])) {

echo $\_SESSION['reg\_email'];

}

?>" required>

<br>

Email" value="<?php

<input type="email" name="reg\_email2" placeholder="Confirm

if (isset($\_SESSION['reg\_email2'])) { echo $\_SESSION['reg\_email2'];

$error\_array))

$error\_array))

}

?>" required>

<br>

<?php if (in\_array("Email already in use<br>",

echo "Email already in use<br>";

else if (in\_array("Invalid email format<br>",

echo "Invalid email format<br>";

else if (in\_array("Emails don't match<br>", $error\_array)) echo "Emails don't match<br>"; ?>

<input type="password" name="reg\_password" placeholder="Password" required>

<br>

<input type="password" name="reg\_password2" placeholder="Confirm Password" required>

<br>

<?php if (in\_array("Your passwords do not match<br>",

$error\_array))

echo "Your passwords do not match<br>";

// else if (in\_array("Your password can only contain

english characters or numbers<br>", $error\_array))

// echo "Your password can only contain english characters or numbers<br>";

// else if (in\_array("Your password must be betwen 5 and 30 characters<br>", $error\_array))

// echo "Your password must be betwen 5 and 30

characters<br>";

value="Register">

?>

<input type="submit" name="register\_button"

<br>

<?php if (in\_array("<span style='color: #8a5000;'>You're all set! Go ahead and login!</span><br>", $error\_array))

echo "<span style='color: #8a5000;'>You're all set! Go ahead and login!</span><br>"; ?>

<a href="#" id="signin" class="signin">Already have an account? Sign in here!</a>

</form>

</div>

</div>

</div>

</body>

</html>

### requests.php

<?php

include("includes/header.php"); //Header

?>

<div class="main\_column column" id="main\_column">

<h4>Friend Requests</h4>

<?php

$query = mysqli\_query($con, "SELECT \* FROM friend\_requests WHERE user\_to='$userLoggedIn'");

if(mysqli\_num\_rows($query) == 0)

echo "You have no friend requests at this time!"; else {

while($row = mysqli\_fetch\_array($query)) {

$user\_from = $row['user\_from'];

$user\_from\_obj = new User($con, $user\_from);

echo $user\_from\_obj->getFirstAndLastName() . " sent you a friend

request!";

$user\_from\_friend\_array = $user\_from\_obj->getFriendArray(); if(isset($\_POST['accept\_request' . $user\_from ])) {

$add\_friend\_query = mysqli\_query($con, "UPDATE users SET friend\_array=CONCAT(friend\_array, '$user\_from,') WHERE username='$userLoggedIn'");

$add\_friend\_query = mysqli\_query($con, "UPDATE users SET friend\_array=CONCAT(friend\_array, '$userLoggedIn,') WHERE username='$user\_from'");

$delete\_query = mysqli\_query($con, "DELETE FROM friend\_requests WHERE user\_to='$userLoggedIn' AND user\_from='$user\_from'");

echo "You are now friends!"; header("Location: requests.php");

}

if(isset($\_POST['ignore\_request' . $user\_from ])) {

$delete\_query = mysqli\_query($con, "DELETE FROM friend\_requests WHERE user\_to='$userLoggedIn' AND user\_from='$user\_from'");

echo "Request ignored!"; header("Location: requests.php");

}

?>

<form action="requests.php" method="POST">

<input type="submit" name="accept\_request<?php echo

$user\_from; ?>" id="accept\_button" value="Accept">

<input type="submit" name="ignore\_request<?php echo

$user\_from; ?>" id="ignore\_button" value="Ignore">

</form>

<?php

}

}

?>

</div>

### search.php

<?php include("includes/header.php");

if(isset($\_GET['q'])) {

$query = $\_GET['q'];

}

else {

$query = "";

}

if(isset($\_GET['type'])) {

$type = $\_GET['type'];

}

else {

$type = "name";

}

?>

<div class="main\_column column" id="main\_column">

<?php

if($query == "")

echo "You must enter something in the search box."; else {

//If query contains an underscore, assume user is searching for usernames

if($type == "username")

$usersReturnedQuery = mysqli\_query($con, "SELECT \* FROM users WHERE username LIKE '$query%' AND user\_closed='no' LIMIT 8");

//If there are two words, assume they are first and last names respectively

else {

$names = explode(" ", $query); if(count($names) == 3)

$usersReturnedQuery = mysqli\_query($con, "SELECT \* FROM users WHERE (first\_name LIKE '$names[0]%' AND last\_name LIKE '$names[2]%') AND user\_closed='no'");

//If query has one word only, search first names or last names else if(count($names) == 2)

$usersReturnedQuery = mysqli\_query($con, "SELECT \* FROM users WHERE (first\_name LIKE '$names[0]%' AND last\_name LIKE '$names[1]%') AND user\_closed='no'");

else

$usersReturnedQuery = mysqli\_query($con, "SELECT \* FROM users WHERE (first\_name LIKE '$names[0]%' OR last\_name LIKE '$names[0]%') AND user\_closed='no'");

}

<br>";

//Check if results were found if(mysqli\_num\_rows($usersReturnedQuery) == 0)

echo "We can't find anyone with a " . $type . " like: " .$query; else

echo mysqli\_num\_rows($usersReturnedQuery) . " results found: <br>

echo "<p id='grey'>Try searching for:</p>";

echo "<a href='search.php?q=" . $query ."&type=name'>Names</a>, <a href='search.php?q=" . $query ."&type=username'>Usernames</a><br><br><hr id='search\_hr'>";

while($row = mysqli\_fetch\_array($usersReturnedQuery)) {

$user\_obj = new User($con, $user['username']);

$button = "";

$mutual\_friends = "";

if($user['username'] != $row['username']) {

//Generate button depending on friendship status if($user\_obj->isFriend($row['username']))

$button = "<input type='submit' name='" . $row['username']

. "' class='danger' value='Remove Friend'>";

else if($user\_obj->didReceiveRequest($row['username']))

$button = "<input type='submit' name='" . $row['username']

. "' class='warning' value='Respond to request'>";

else if($user\_obj->didSendRequest($row['username']))

$button = "<input type='submit' class='default' value='Request Sent'>";

else

$button = "<input type='submit' name='" . $row['username']

. "' class='success' value='Add Friend'>";

$mutual\_friends = $user\_obj-

>getMutualFriends($row['username']) . " friends in common";

//Button forms if(isset($\_POST[$row['username']])) {

if($user\_obj->isFriend($row['username'])) {

$user\_obj->removeFriend($row['username']); header("Location:

http://$\_SERVER[HTTP\_HOST]$\_SERVER[REQUEST\_URI]");

}

else if($user\_obj->didReceiveRequest($row['username'])) { header("Location: requests.php");

}

else if($user\_obj->didSendRequest($row['username'])) {

}

else {

$user\_obj->sendRequest($row['username']); header("Location:

http://$\_SERVER[HTTP\_HOST]$\_SERVER[REQUEST\_URI]");

}

}

}

echo "<div class='search\_result'>

<div class='searchPageFriendButtons'>

<form action='' method='POST'> " . $button . "

<br>

</form>

</div>

<div class='result\_profile\_pic'>

<a href='" . $row['username'] ."'><img src='".

$row['profile\_pic'] ."' style='height: 100px;'></a>

</div>

<a href='" . $row['username'] ."'> " .

$row['first\_name'] . " " . $row['last\_name'] . "

<p id='grey'> " . $row['username'] ."</p>

</a>

<br>

" . $mutual\_friends ."<br>

</div>

<hr id='search\_hr'>";

} //End while

}

?>

</div>

### settings.php

<?php include("includes/header.php");

include("includes/form\_handlers/settings\_handler.php");

?>

<div class="main\_column column">

<h4>Account Settings</h4>

<?php

echo "<img src='" . $user['profile\_pic'] ."' class='small\_profile\_pic'>";

?>

<br>

<a href="upload.php">Upload new profile picture</a> <br><br><br> Modify the values and click 'Update Details'

<?php

$user\_data\_query = mysqli\_query($con, "SELECT first\_name, last\_name, email FROM users WHERE username='$userLoggedIn'");

$row = mysqli\_fetch\_array($user\_data\_query);

$first\_name = $row['first\_name'];

$last\_name = $row['last\_name'];

$email = $row['email'];

?>

<form action="settings.php" method="POST">

First Name: <input type="text" name="first\_name" value="<?php echo

$first\_name; ?>" id="settings\_input"><br>

Last Name: <input type="text" name="last\_name" value="<?php echo

$last\_name; ?>" id="settings\_input"><br>

Email: <input type="text" name="email" value="<?php echo $email; ?>" id="settings\_input"><br>

<?php echo $message; ?>

<input type="submit" name="update\_details" id="save\_details" value="Update Details" class="info settings\_submit"><br>

</form>

<h4>Change Password</h4>

<form action="settings.php" method="POST">

Old Password: <input type="password" name="old\_password" id="settings\_input"><br>

New Password: <input type="password" name="new\_password\_1" id="settings\_input"><br>

Confirm New Password : <input type="password" name="new\_password\_2" id="settings\_input"><br>

<?php echo $password\_message; ?>

<input type="submit" name="update\_password" id="save\_details" value="Update Password" class="info settings\_submit"><br>

</form>

<h4>Close Account</h4>

<form action="settings.php" method="POST">

<input type="submit" name="close\_account" id="close\_account" value="Close Account" class="danger settings\_submit">

</form>

</div>

### upload.php

<?php include("includes/header.php");

$profile\_id = $user['username'];

$imgSrc = "";

$result\_path = "";

$msg = "";

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. - Remove The Temp image if it exists

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if (!isset($\_POST['x']) && !isset($\_FILES['image']['name']) ){

//Delete users temp image

$temppath = 'assets/images/profile\_pics/'.$profile\_id.'\_temp.jpeg';

if (file\_exists ($temppath)){ @unlink($temppath); }

}

if(isset($\_FILES['image']['name'])){

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. - Upload Original Image To Server

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Get Name | Size | Temp Location

$ImageName = $\_FILES['image']['name'];

$ImageSize = $\_FILES['image']['size'];

$ImageTempName = $\_FILES['image']['tmp\_name'];

//Get File Ext

$ImageType = @explode('/', $\_FILES['image']['type']);

$type = $ImageType[1]; //file type

//Set Upload directory

$uploaddir =

$\_SERVER['DOCUMENT\_ROOT'].'/Demo/assets/images/profile\_pics';

//Set File name

$file\_temp\_name = $profile\_id.'\_original.'.md5(time()).'n'.$type;

//the temp file name

$fullpath = $uploaddir."/".$file\_temp\_name; // the temp file path

$file\_name = $profile\_id.'\_temp.jpeg'; //$profile\_id.'\_temp.'.$type;

// for the final resized image

$fullpath\_2 = $uploaddir."/".$file\_name; //for the final resized image

//Move the file to correct location

$move = move\_uploaded\_file($ImageTempName ,$fullpath) ; chmod($fullpath, 0777);

//Check for valid uplaod if (!$move) {

die ('File didnt upload');

} else {

$imgSrc= "assets/images/profile\_pics/".$file\_name; // the image to display in crop area

$msg= "Upload Complete!"; //message to page

$src = $file\_name; //the file name to post from cropping form to the resize

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2 - Resize The Image To Fit In Cropping Area

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//get the uploaded image size clearstatcache();

$original\_size = getimagesize($fullpath);

$original\_width = $original\_size[0];

$original\_height = $original\_size[1];

// Specify The new size

$main\_width = 500; // set the width of the image

$main\_height = $original\_height / ($original\_width /

$main\_width); // this sets the height in ratio

//create new image using correct php func if($\_FILES["image"]["type"] == "image/gif"){

$src2 = imagecreatefromgif($fullpath);

}elseif($\_FILES["image"]["type"] == "image/jpeg" ||

$\_FILES["image"]["type"] == "image/pjpeg"){

$src2 = imagecreatefromjpeg($fullpath);

}elseif($\_FILES["image"]["type"] == "image/png"){

$src2 = imagecreatefrompng($fullpath);

}else{

$msg .= "There was an error uploading the file. Please upload a .jpg, .gif or .png file. <br />";

}

//create the new resized image

$main = imagecreatetruecolor($main\_width,$main\_height); imagecopyresampled($main,$src2,0, 0, 0,

0,$main\_width,$main\_height,$original\_width,$original\_height);

//upload new version

$main\_temp = $fullpath\_2; imagejpeg($main, $main\_temp, 90); chmod($main\_temp,0777);

//free up memory imagedestroy($src2); imagedestroy($main);

//imagedestroy($fullpath);

@ unlink($fullpath); // delete the original

upload

}//ADD Image

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 3- Cropping & Converting The Image To Jpg

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ if (isset($\_POST['x'])){

//the file type posted

$type = $\_POST['type'];

//the image src

$src = 'assets/images/profile\_pics/'.$\_POST['src'];

$finalname = $profile\_id.md5(time());

if($type == 'jpg' || $type == 'jpeg' || $type == 'JPG' || $type == 'JPEG'){

//the target dimensions 150x150

$targ\_w = $targ\_h = 150;

//quality of the output

$jpeg\_quality = 90;

//create a cropped copy of the image

$img\_r = imagecreatefromjpeg($src);

$dst\_r = imagecreatetruecolor( $targ\_w, $targ\_h );

imagecopyresampled($dst\_r,$img\_r,0,0,$\_POST['x'],$\_POST['y'],

$targ\_w,$targ\_h,$\_POST['w'],$\_POST['h']);

//save the new cropped version imagejpeg($dst\_r,

"assets/images/profile\_pics/".$finalname."n.jpeg", 90);

}else if($type == 'png' || $type == 'PNG'){

//the target dimensions 150x150

$targ\_w = $targ\_h = 150;

//quality of the output

$jpeg\_quality = 90;

//create a cropped copy of the image

$img\_r = imagecreatefrompng($src);

$dst\_r = imagecreatetruecolor( $targ\_w, $targ\_h ); imagecopyresampled($dst\_r,$img\_r,0,0,$\_POST['x'],$\_POST['y'],

$targ\_w,$targ\_h,$\_POST['w'],$\_POST['h']);

//save the new cropped version imagejpeg($dst\_r,

"assets/images/profile\_pics/".$finalname."n.jpeg", 90);

}else if($type == 'gif' || $type == 'GIF'){

//the target dimensions 150x150

$targ\_w = $targ\_h = 150;

//quality of the output

$jpeg\_quality = 90;

//create a cropped copy of the image

$img\_r = imagecreatefromgif($src);

$dst\_r = imagecreatetruecolor( $targ\_w, $targ\_h ); imagecopyresampled($dst\_r,$img\_r,0,0,$\_POST['x'],$\_POST['y'],

$targ\_w,$targ\_h,$\_POST['w'],$\_POST['h']);

//save the new cropped version imagejpeg($dst\_r,

"assets/images/profile\_pics/".$finalname."n.jpeg", 90);

}

//free up memory

imagedestroy($img\_r); // free up memory imagedestroy($dst\_r); //free up memory

@ unlink($src); // delete the original upload

//return cropped image to page

$result\_path ="assets/images/profile\_pics/".$finalname."n.jpeg";

//Insert image into database

$insert\_pic\_query = mysqli\_query($con, "UPDATE users SET profile\_pic='$result\_path' WHERE username='$userLoggedIn'");

header("Location: ".$userLoggedIn);

}// post x

?>

<div id="Overlay" style=" width:100%; height:100%; border:0px #990000 solid; position:absolute; top:0px; left:0px; z-index:2000; display:none;"></div>

<div class="main\_column column">

<div id="formExample">

<p><b> <?=$msg?> </b></p>

data">

<form action="upload.php" method="post" enctype="multipart/form-

Upload something<br /><br />

<input type="file" id="image" name="image" style="width:200px;

height:30px; " /><br /><br />

<input type="submit" value="Submit" style="width:85px; height:25px;" />

</form><br /><br />

</div> <!-- Form-->

<?php

if($imgSrc){ //if an image has been uploaded display cropping area?>

<script>

$('#Overlay').show();

$('#formExample').hide();

</script>

<div id="CroppingContainer" style="width:800px; max-height:600px; background-color:#FFF; margin-left: -200px; position:relative; overflow:hidden; border:2px #666 solid; z-index:2001; padding-bottom:0px;">

<div id="CroppingArea" style="width:500px; max-height:400px; position:relative; overflow:hidden; margin:40px 0px 40px 40px; border:2px #666 solid; float:left;">

<img src="<?=$imgSrc?>" border="0" id="jcrop\_target" style="border:0px #990000 solid; position:relative; margin:0px 0px 0px 0px; padding:0px; " />

</div>

<div id="InfoArea" style="width:180px; height:150px; position:relative; overflow:hidden; margin:40px 0px 0px 40px; border:0px #666 solid; float:left;">

<p style="margin:0px; padding:0px; color:#444; font-

size:18px;">

click save.

<b>Crop Profile Image</b><br /><br />

<span style="font-size:14px;">

Crop / resize your uploaded profile image. <br /> Once you are happy with your profile image then please

</p>

</div>

<br />

</span>

<div id="CropImageForm" style="width:100px; height:30px; float:left; margin:10px 0px 0px 40px;" >

<form action="upload.php" method="post" onsubmit="return checkCoords();">

$type ?>

height:30px;" />

<input type="hidden" id="x" name="x" />

<input type="hidden" id="y" name="y" />

<input type="hidden" id="w" name="w" />

<input type="hidden" id="h" name="h" />

<input type="hidden" value="jpeg" name="type" /> <?php //

<input type="hidden" value="<?=$src?>" name="src" />

<input type="submit" value="Save" style="width:100px;

</form>

</div>

<div id="CropImageForm2" style="width:100px; height:30px; float:left; margin:10px 0px 0px 40px;" >

<form action="upload.php" method="post" onsubmit="return

cancelCrop();">

<input type="submit" value="Cancel Crop"

style="width:100px; height:30px;" />

</form>

</div>

</div><!-- CroppingContainer -->

<?php

} ?>

</div>

<?php if($result\_path) {

?>

<img src="<?=$result\_path?>" style="position:relative; margin:10px auto; width:150px; height:150px;" />

<?php } ?>

<br /><br />

### User\_closed.php

<?php include("includes/header.php");

?>

<div class="main\_column column" id="main\_column">

<h4>User Closed</h4>

This account is closed.

<a href="index.php"> Click here to go back.</a>

</div>

**6.1 TEST CASES**

# CHAPTER – 6 TESTING

### Test Case to check if an account is already created

|  |  |
| --- | --- |
| **Test Case ID:** | 1 |
| Test Case Name: | Creating a user |
| Purpose: | To create an account in the website. |
| Input: | Username, email, password, confirm password. |
| Expected Result: | An account should be created |
| Actual Result: | An account is created and additional details are entered. |
| Failure | If any account is already created with same user name or email, then accou is taken with same username is displayed. |

**Table 6.1.1 Creating user testing**

### Test Case to check search for an user

|  |  |
| --- | --- |
| **Test Case ID:** | 2 |
| Test Case Name: | Searching for an user. |
| Purpose: | To search for an user. |
| Input: | To enter username. |
| Expected Result: | Results for requested user. |
| Actual Result: | Results for @user. |
| Failure | If any user is not found, blank screen is displayed. |

**Table 6.1.2 Searching for an user testing**

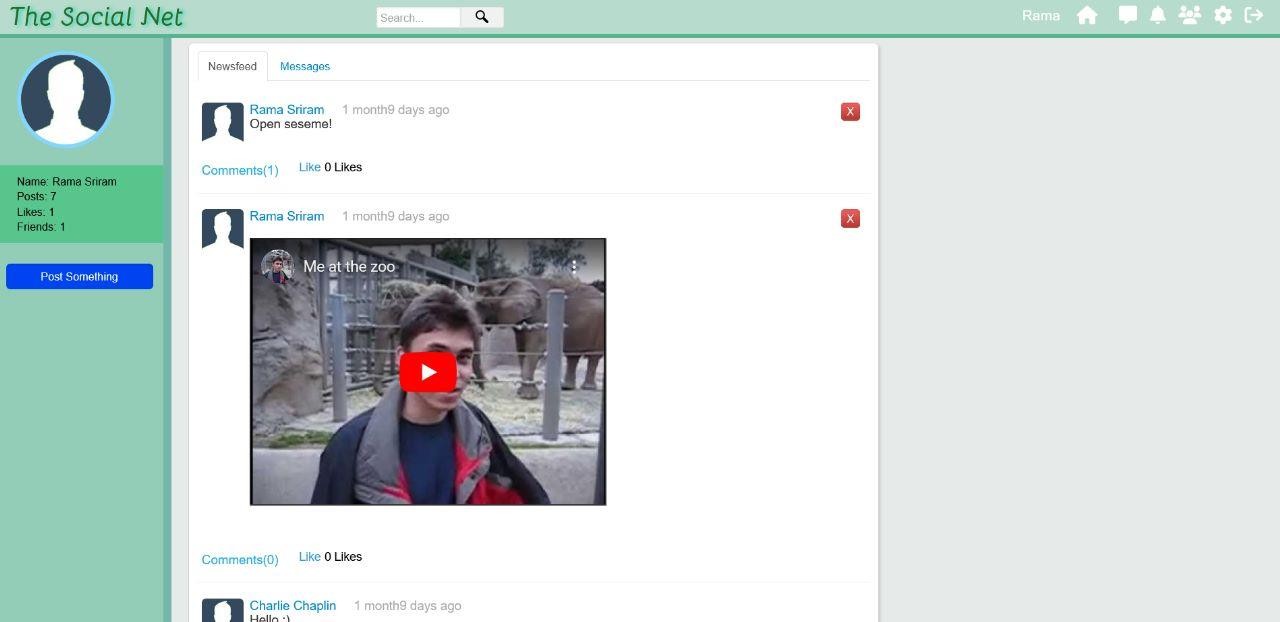
### Test Case to upload pictures

|  |  |
| --- | --- |
| **Test Case ID:** | 3 |
| Test Case Name: | Test case to upload pictures |
| Purpose: | To upload the pictures |
| Input: | A file containing the picture is selected |

|  |  |
| --- | --- |
| Expected Result: | A picture is uploaded |
| Actual Result: | Picture is uploaded and returned to homepage |
| Failure | If any errors is shown then check files path |

**Table 6.1.3 Upload pictures testing**

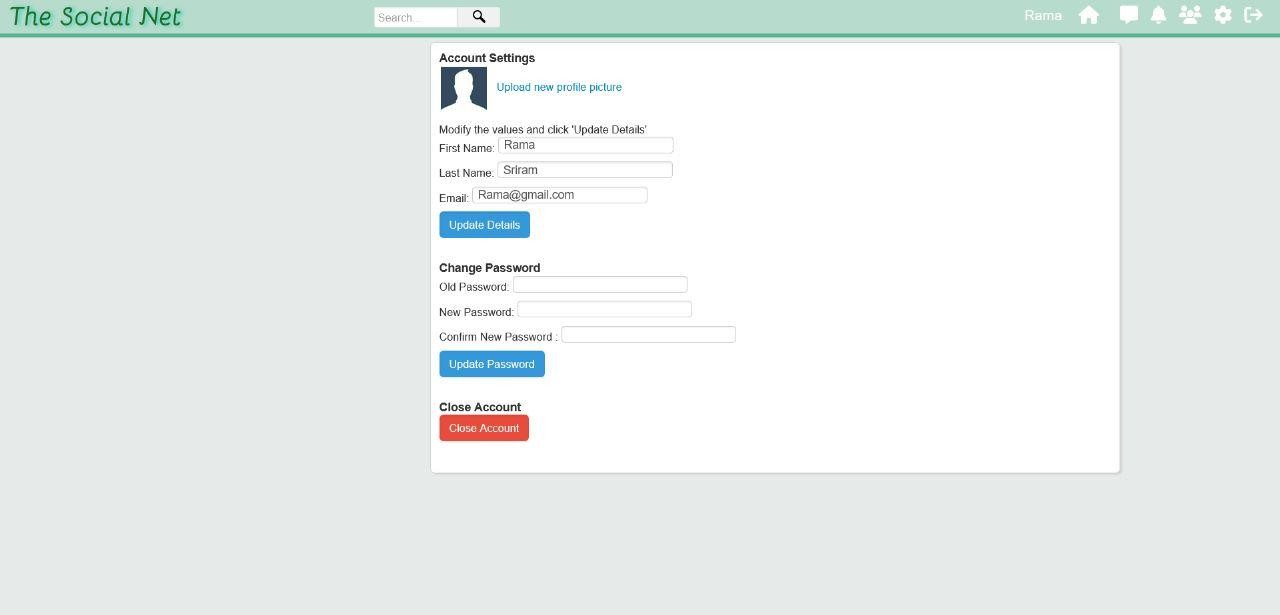
# CHAPTER - 7 SCREENSHOTS



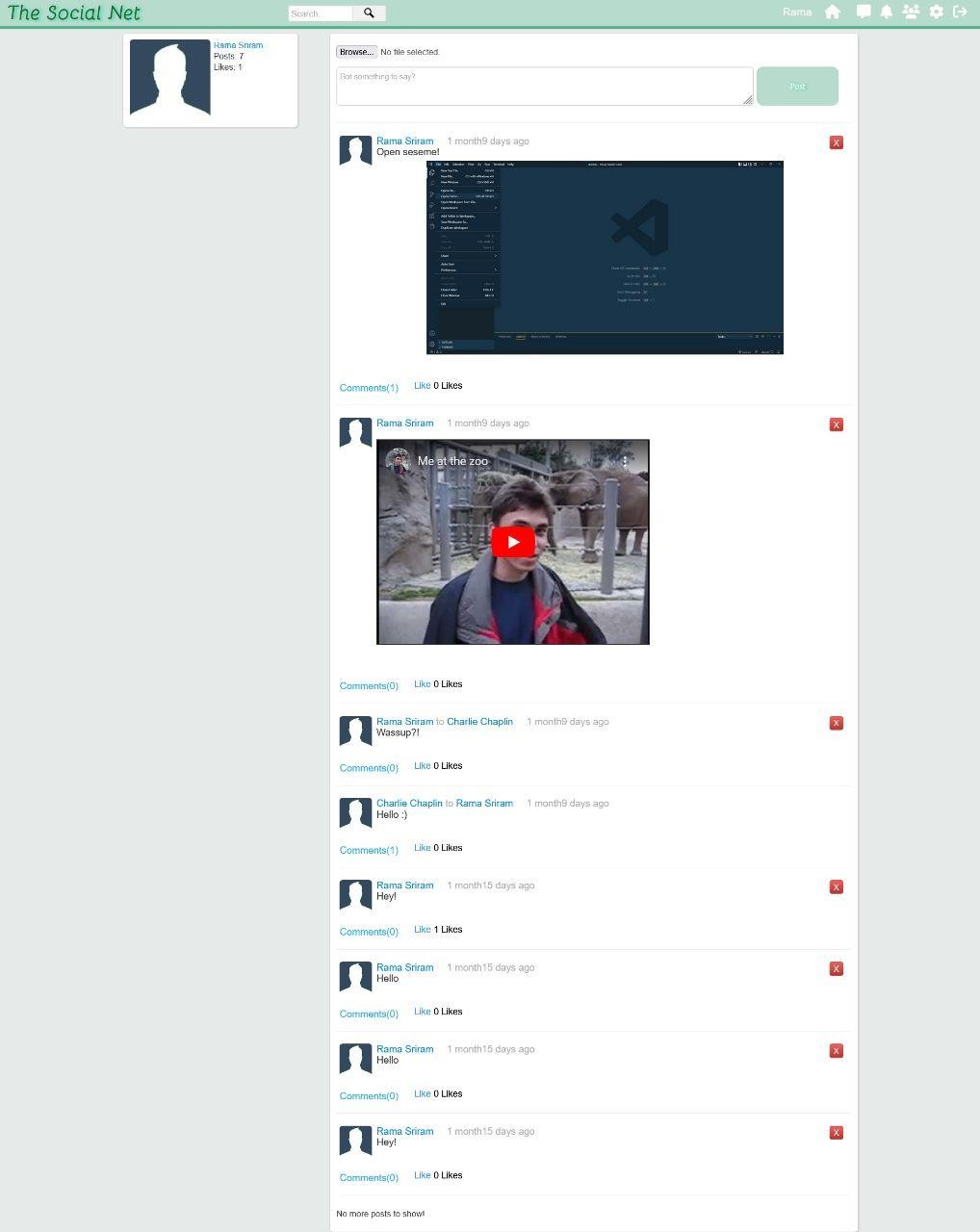
*Figure 7.1: HOME PAGE*



*Figure 7.2: LOG IN PAGE*



*Figure 7.3: USER ACCOUNT PAGE*



*Figure 7.4: POSTS PAGE*

# CHAPTER - 8 CONCLUSION AND FUTURE SCOPE

In conclusion, a social media website using PHP can be a powerful platform for connecting people and facilitating communication and collaboration. There are many features that can be implemented in a social networking application using PHP, such as user authentication and authorization, user profiles, news feeds, messaging, and notifications.

As for the future scope of such an application, there are many possibilities for further development and improvement. Some ideas might include integrating additional social media platforms and APIs, adding support for multimedia content and enhanced multimedia features, implementing analytics and reporting tools, and adding additional security and privacy measures. Additionally, the application could be extended to support additional languages and internationalization, and could be optimized for mobile devices.

# BIBLIOGRAPHY

1. <https://www.php.net/manual/en/intro-whatis.php>
2. <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
3. https://jquery.com/
4. htt[ps://www.udemy.c](http://www.udemy.com/course/make-a-social-media-website/)om[/cour](http://www.udemy.com/course/make-a-social-media-website/)se[/mak](http://www.udemy.com/course/make-a-social-media-website/)e[-a-social-media-website/](http://www.udemy.com/course/make-a-social-media-website/)

**APPENDIX A**: TOOLS AND TECHNOLOGIES

* XAMPP: XAMPP is a software package that enables users to create a local web server environment on their computer. It stands for Cross-platform, Apache, MySQL, PHP, and Perl.
* WINDOWS 11: Windows 11 was used as the operating system.
* VS Code: Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux, and macOS
* CSS: CSS is a design language that you use to make your web page look nice and presentable. CSS stands for Cascading Style Sheets, and you use it to improve the appearance of a web page.
* HTML, JavaScript: The HTML<script> tag is used to define a client-side script (JavaScript). The

<script> element either contains script statements, or it points to an external script file through the src attribute. HTML is a markup language used to format/structure a web page.

* MySQL: MySQL is an open-source relational database management system (RDBMS) that provides a comprehensive set of features for managing and organizing large amounts of data. It is one of the most popular database systems used in web applications, and it is known for its performance, reliability, and scalability.
* jQuery : jQuery is a popular JavaScript library that simplifies HTML document traversal, event handling, animating, and Ajax interactions for rapid web development. It is designed to make it easier to navigate a document, select elements, create animations, handle events, and perform Ajax requests
* Apache:e: Apache refers to the Apache HTTP Server, which is a widely used open-source web server software. It is the most popular and widely used web server software globally. Apache is known for its security, reliability, and scalability.