Session: 2021 – 2025

Submitted by:

Abdul Mateen	2021-CS-190
Khadim Hussain	2021-CS-204

Supervised By:

Dr. Maida Shahid

Department of Computer Science

University of Engineering and Technology, Lahore Pakistan

Table of Contents

Introduction

This Project is a web based Social Media Platform where you can add other people as your friend and share your beautiful memories with them. You can also view what they are doing in their life. You can share your review by liking and disliking there post. You can also message them and ask them how there life is going.

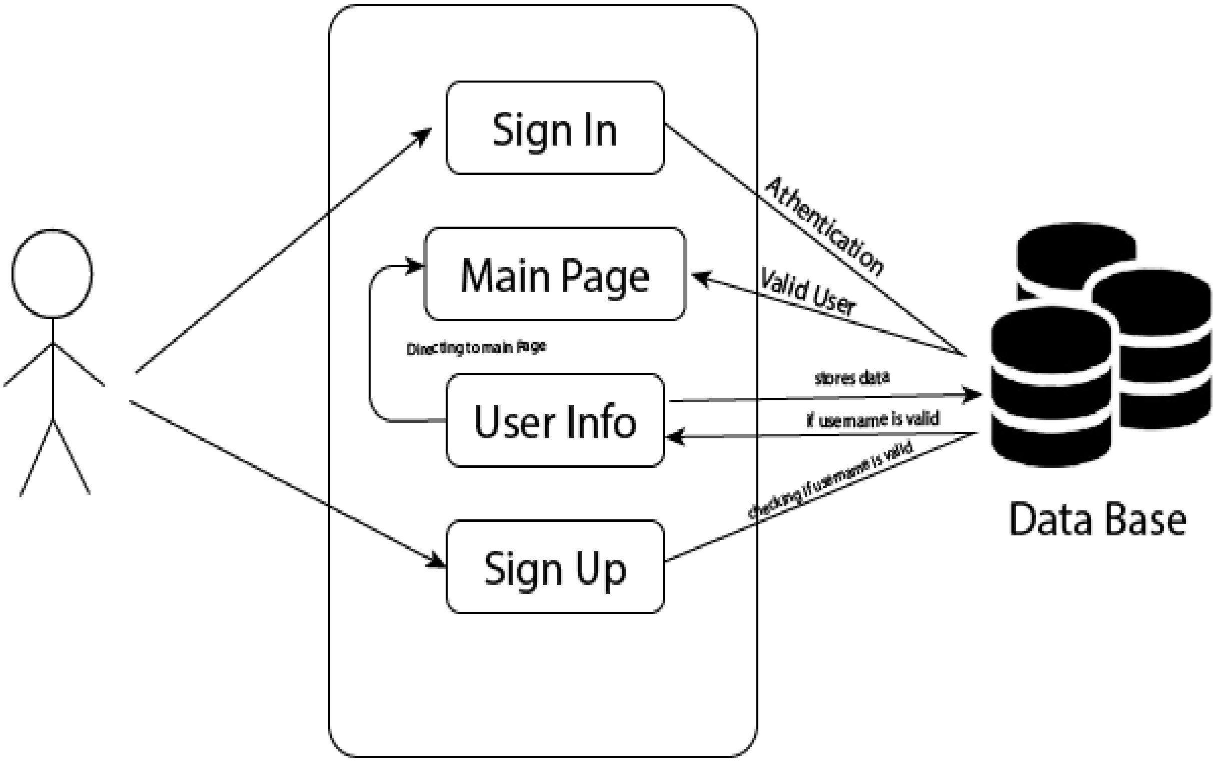
Project Features

Most Important features are mentioned below:

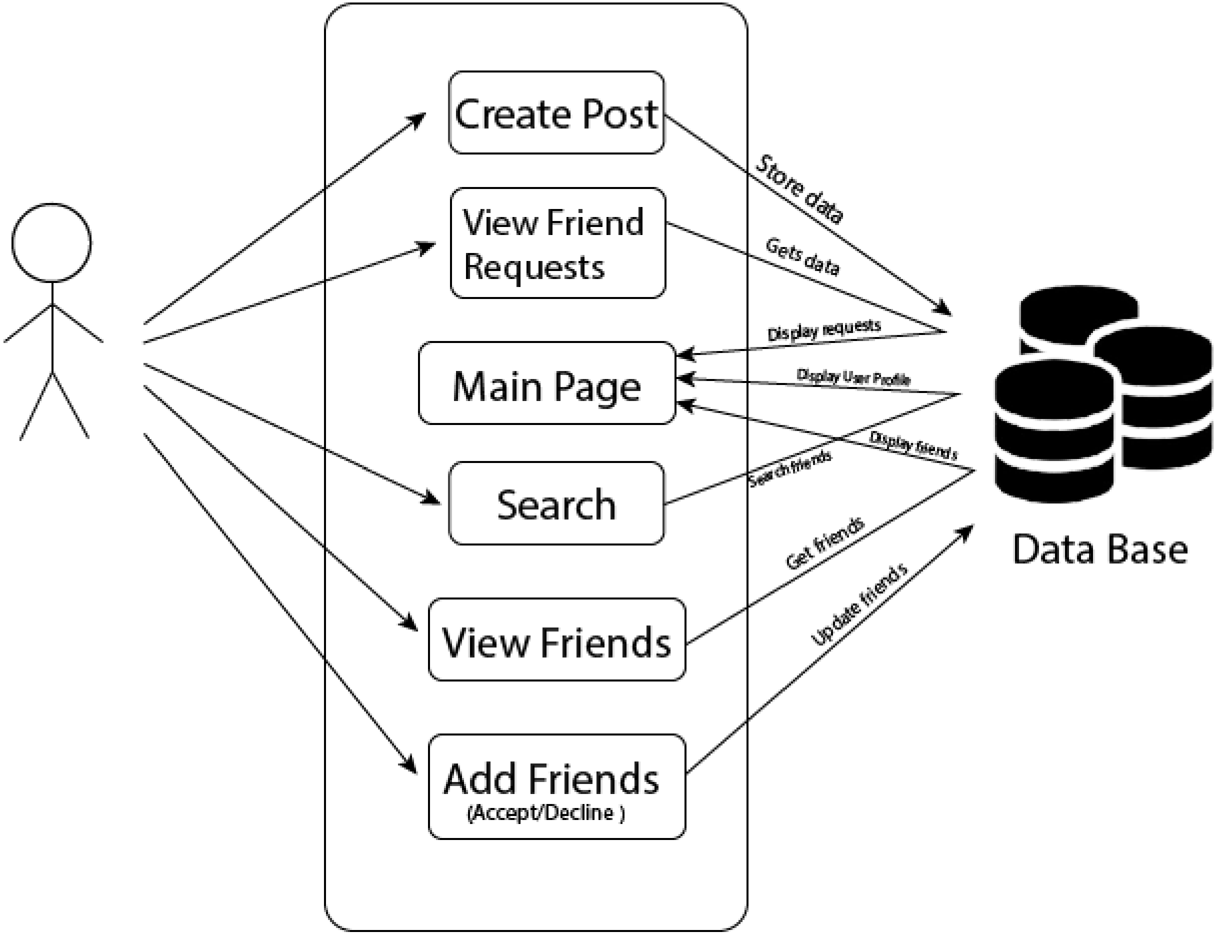
- Post your Picture and Caption it
- Add Friends
- Like and Dislike Posts of others
- Message your Friends

Use Cases

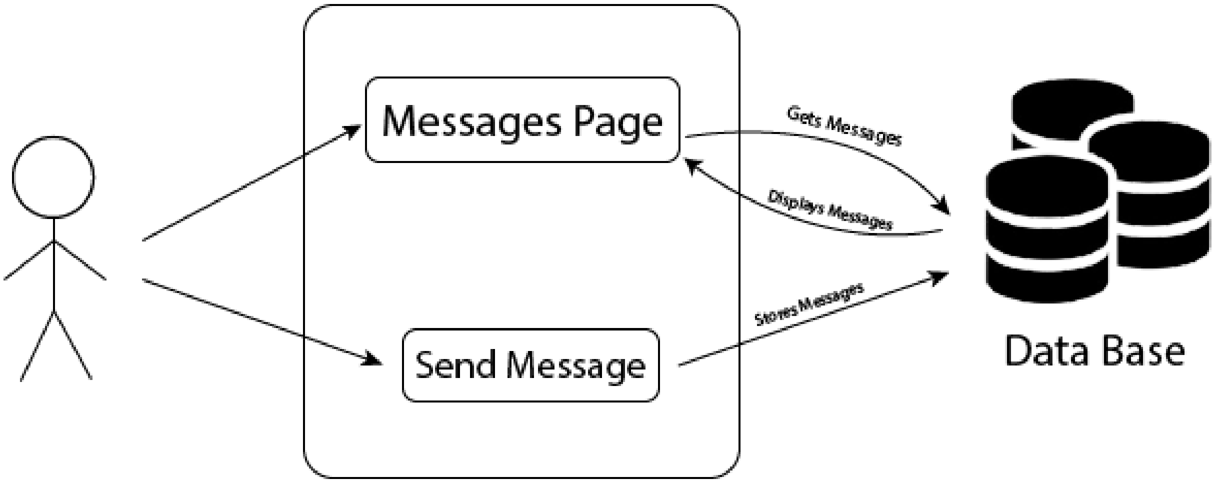
SignIn/ SignUp



Main Page

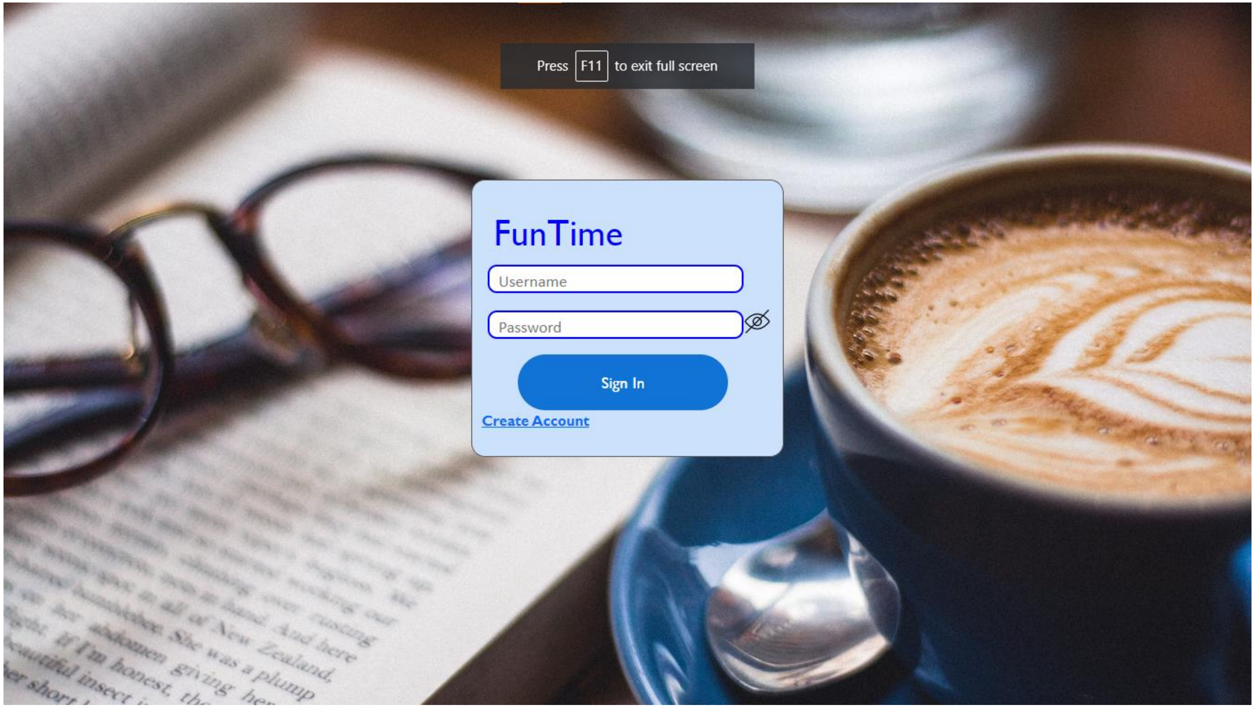


Messages

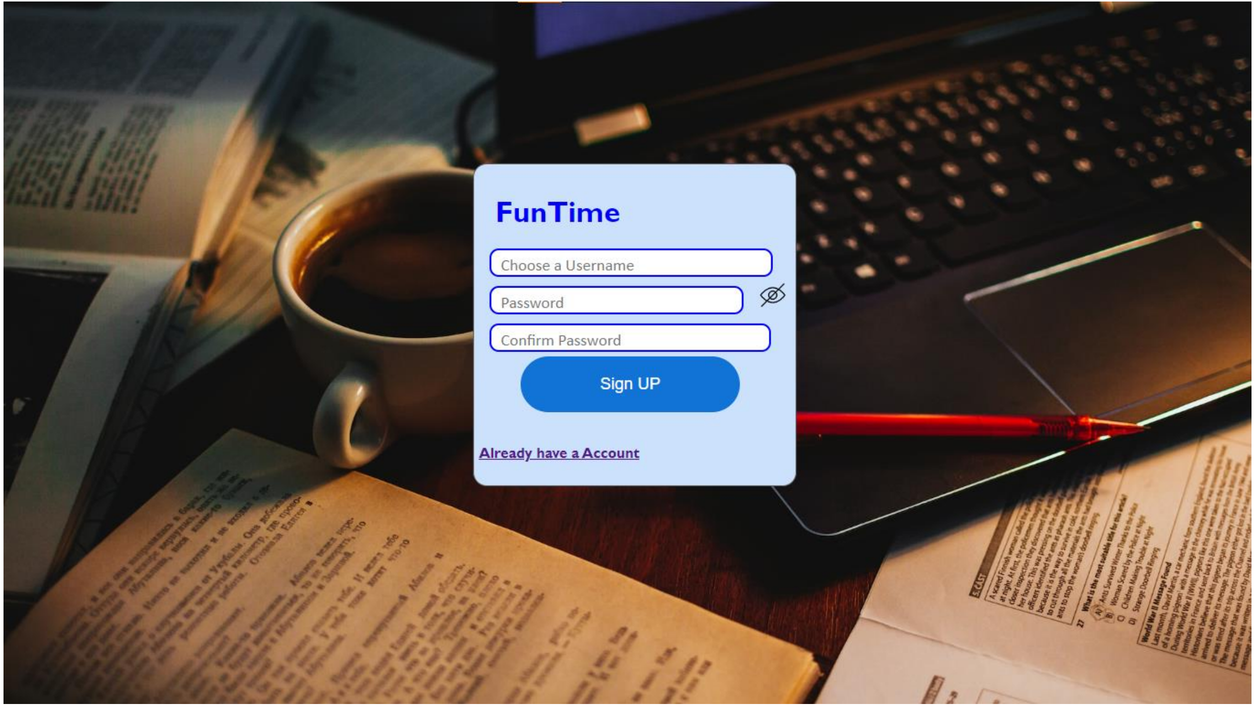


Wire Frames

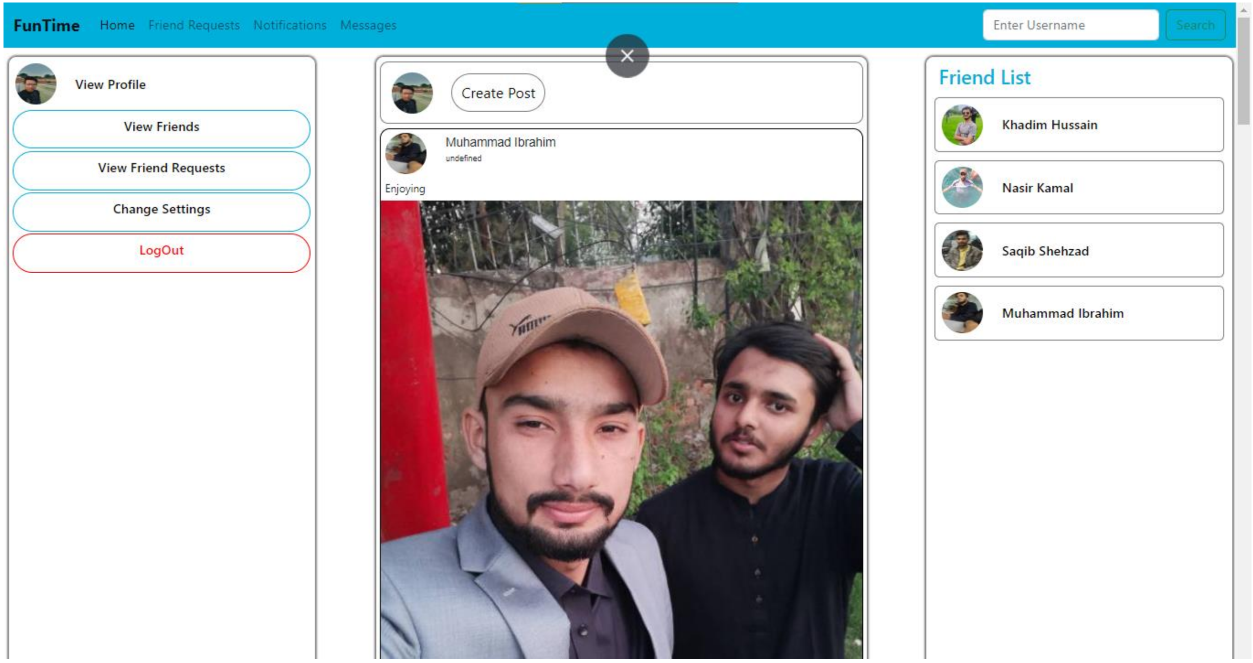
Login Page



Sign Up Page:



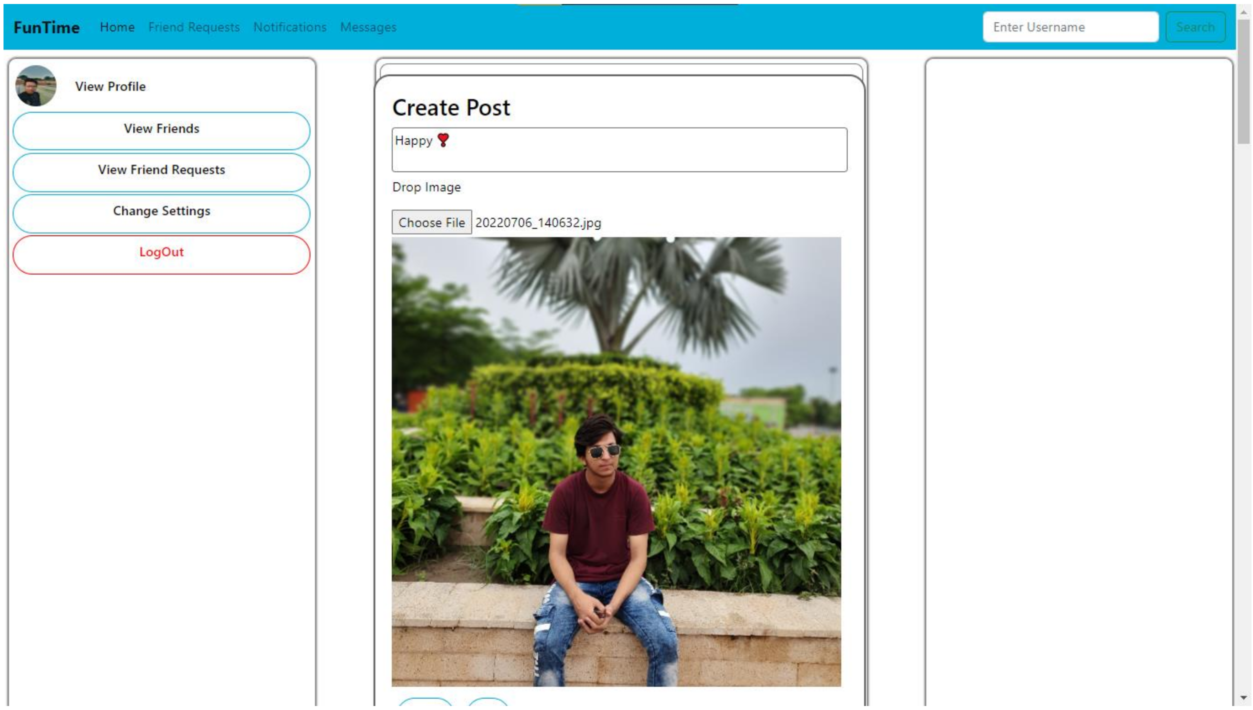
Main Page:



Muhammad Ibrahim
undefined
Enjoying

Khadim Hussain
Nasir Kamal
Saqib Shehzad
Muhammad Ibrahim

Create Post:



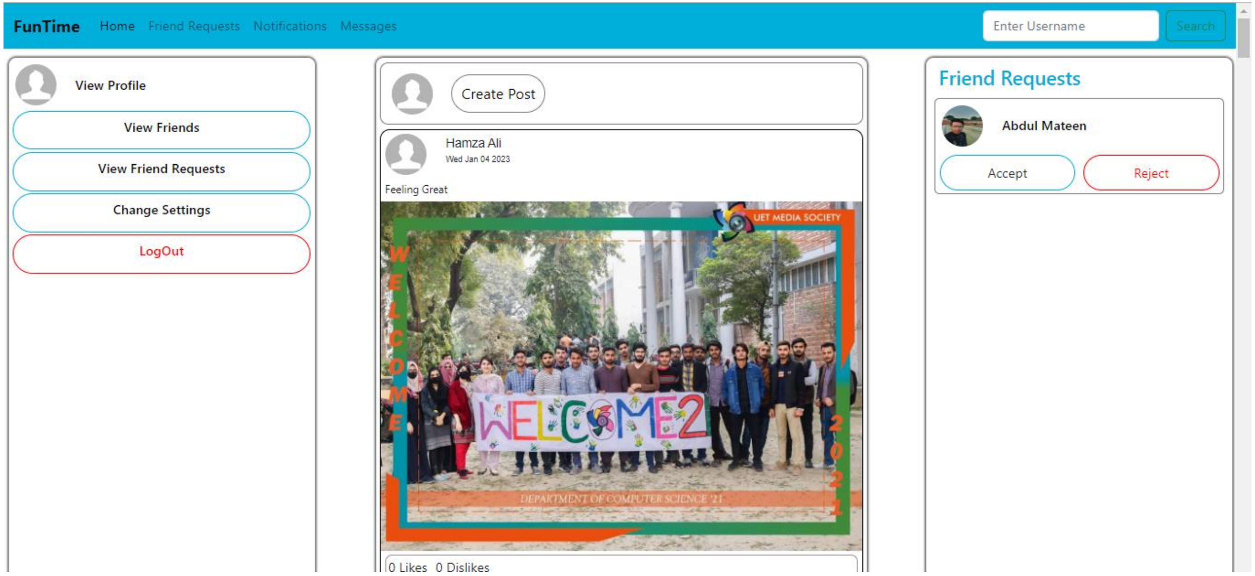
Create Post

Happy 🌹

Drop Image

20220706_140632.jpg

Friend Requests:



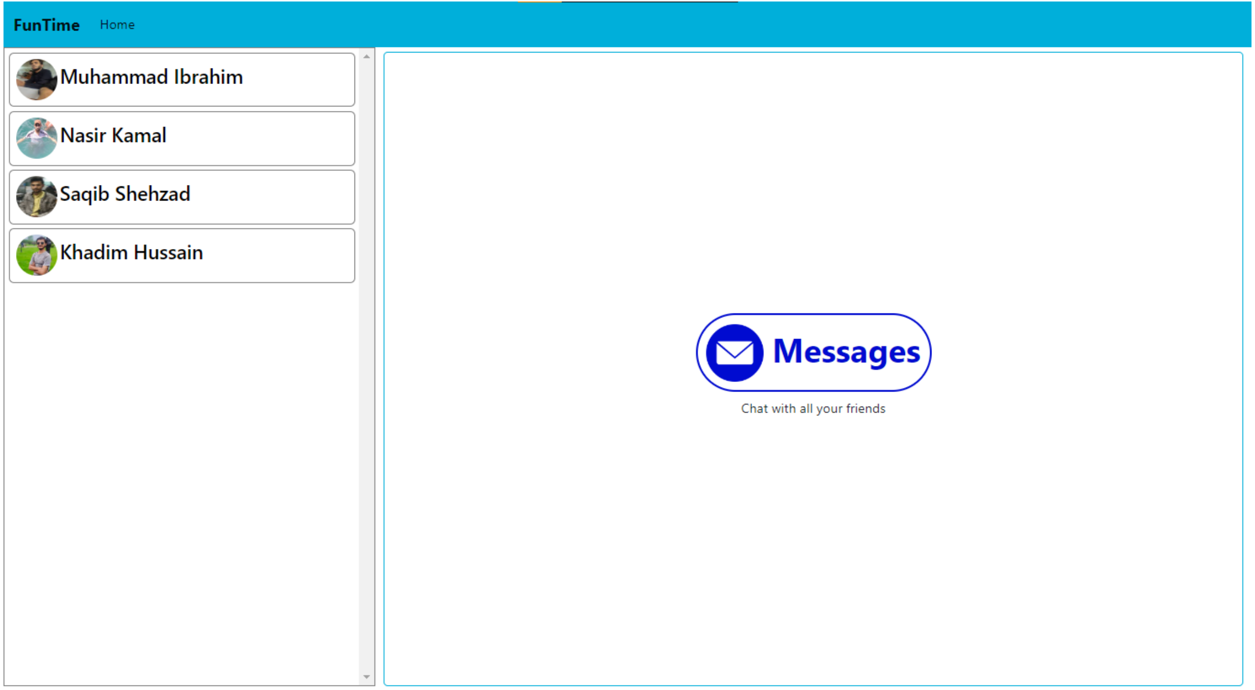
Hamza Ali
Wed Jan 04 2023

Feeling Great

0 Likes 0 Dislikes

Abdul Mateen

Messages:



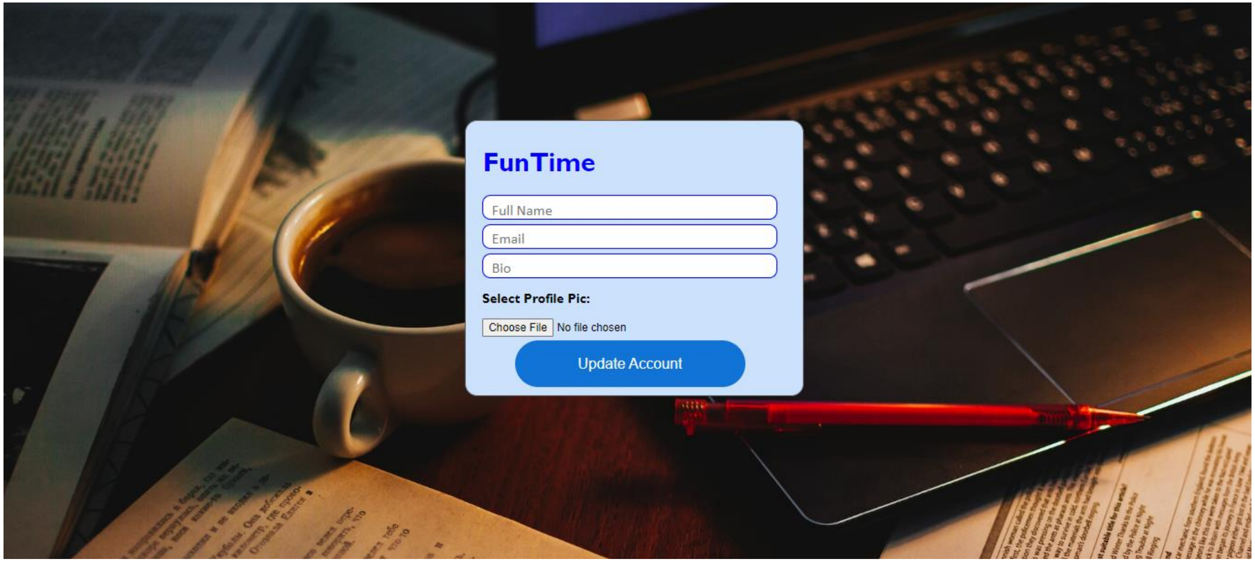
Muhammad Ibrahim

Nasir Kamal

Saqib Shehzad

Khadim Hussain

Change Settings



Technology and Resources

This is web based project so we used

- HTML
- CSS
- JavaScript

The database we used to store data of users and their posts is Firebase Real-time database.

Code

Main-page.html:

```
<!DOCTYPE html>

<html lang="en">


  <head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css">

    <link rel="stylesheet" href="/CSS/post.css">

    <link rel="stylesheet" href="/CSS/main-page.css">
    <link rel="stylesheet" href="/CSS//notification.css">
</head>


<body>
    <nav class="navbar fixed-top navbar-expand-lg " style="background-color: #00afda; position: sticky;">
        <div class="container-fluid">
            <a class="navbar-brand fw-bold" href="/Pages/Main-Page.html">FunTime</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarScroll"
                aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarScroll">
                <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scroll-height: 100px;">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="/Pages/Main-Page.html">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#" id="friend_requests_noti">Friend Requests</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" id="notification_btn" href="#">Notifications</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" id="msgs_option" href="/Pages/messages.html">Messages</a>
                    </li>
                    <div id="unread_msgs">
                    </div>
                    <li class="nav-item">
                        <!-- <a class="nav-link disabled">Link</a> -->
                    </li>
                </ul>
                <div class="d-flex">
                <input class="form-control me-2" id="search_input" type="text" placeholder="Enter Username"
                  aria-label="Search">
                <button class="btn btn-outline-success" id="search_btn">Search</button>
              </div>
            </div>
          </div>
        </nav>
        <div id="whole-screen">
          <!-- Pop up Menu -->
          <div id="new-post-popup" class="popup">
            <h3>Create Post</h3>
            <textarea name="Caption" id="caption-area" cols=auto rows=auto placeholder="Add
Caption"></textarea>
            <p>Drop Image</p>
            <input id="img-input" type="file" accept=".jpg,.png,.heic" />
            <div id="root"></div>
            <button id="cancle_btn" class="setting-card" style="margin-top: 5px;">Cancel</button>
            <button id="post_btn" class="setting-card" style="margin-top: 5px;">Post</button>
          </div>
          <div id="div-setting">
            <a href="/Pages/Profile.html" class="no-link">
              <div id="view-profile">
                <img id="profile_pic" class="profile_pic"
                  src="https://uxwing.com/wp-content/themes/uxwing/download/peoples-avatars/no-profile-
picture-icon.png"
                  alt="">
                <h6 style="text-align: center;">View Profile</h6>
              </div>
            </a>
            <a href="#" class="no-link">
              <div id="view_friends" class="setting-card">
                <h6>View Friends</h6>
              </div>
            </a>
            <a href="#" class="no-link">
              <div id="friend-requests" class="setting-card">
                <h6>View Friend Requests</h6>
              </div>
            </a>
            <div class="no-link">
              <div id="change_setting" class="setting-card">
                <h6>Change Settings</h6>
              </div>
            </div>
            <a href="login_page.html" class="no-link">
              <div id="logout">
                <h6>LogOut</h6>
              </div>
            </a>
          </div>
          <div id="main-timeline">
            <div id="new-post">
              <img id="profile_pic_createPost" class="profile_pic"
                src="https://uxwing.com/wp-content/themes/uxwing/download/peoples-avatars/no-profile-picture-icon.png"
                alt="">
              <button id="createPost_btn" class="setting-card">Create Post</button>
            </div>
            <div id="posts_section"></div>
            <!-- <button id="load_posts">Load More Posts</button> -->
          </div>
          <div id="extra-features" onclick="">
          </div>
        </div>
        </div>
```

```
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
        <script type="module" src="/js/main-page.js"></script>
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
    </body>

</html>
```

Main-page.js:

```
let username = "123",
        postCount = 0,
        imgurl,
        messageRead = true;
const MAX_WIDTH = 1280;
const MAX_HEIGHT = 720;
const MIME_TYPE = "image/jpeg";
const QUALITY = 0.7;
let post_imgurl;
//<------------------------------Bachy is code se door rahy------------------------------------->

//<----------------------------------Linking With DataBase------------------------------------->

import { initializeApp } from "https://www.gstatic.com/firebasejs/9.15.0/firebase-app.js";
import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.15.0/firebase-analytics.js";

const firebaseConfig = {
        apiKey: "AIzaSyDgjxC3pegDxwodMo_tBF6og-5NAtccu_U",
        authDomain: "justteatingfirebase.firebaseapp.com",
        databaseURL: "https://justteatingfirebase-default-rtdb.firebaseio.com",
        projectId: "justteatingfirebase",
        storageBucket: "justteatingfirebase.appspot.com",
        messagingSenderId: "1055889204987",
        appId: "1:1055889204987:web:33450c37ba9cf8d798b25e",
        measurementId: "G-LQLCQBVPS3",
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);

import {
    getDatabase,
    get,
    set,
    ref,
    child,
    update,
    onValue,
} from "https://www.gstatic.com/firebasejs/9.15.0/firebase-database.js";
const db = getDatabase();

async function SelectData() {
    // console.log("Dp values Assigend" + username);
    const dbref = ref(db);
    await get(child(dbref, "Usernames/" + username)).then((snapshot) => {
        if (snapshot.exists()) {
            imgurl = snapshot.val().imgurl;
            document.getElementById("profile_pic").src = imgurl;
            document.getElementById("profile_pic_createPost").src = imgurl;
            messageRead = snapshot.val().messagesRead;
        }
    });
    newMessage();
}
function newMessage() {
    if (messageRead == false) {
```

```
                document.getElementById("unread_msgs").classList += "color_red";
        } else {
                document.getElementById("unread_msgs").classList -= "color_red";
        }
}
function storePostInDataBase() {
        document.getElementById("posts_section").innerHTML = "";
        let text = document.getElementById("caption-area");
        let author = username;
        let likes = "";
        let dislikes = "";
        let comments = "";
        let date1 = new Date();
        let hours = date1.getHours();
        let min = date1.getMinutes();
        let am = "AM";
        if (hours > 12) {
                hours = hours - 12;
                am = "PM";
        }
        let date = `${hours}:${min} ${am} , ${date1.toDateString()}`;
        console.log(date);
        set(ref(db, "Posts/" + author + postCount), {
                text: text.value,
                imgurl: post_imgurl,
                author: author,
                likes: likes,
                dislikes: dislikes,
                comments: comments,
                date: date,
        })
                .then(() => {
                    alert("Data stored suxxessfully");
                })
                .catch((error) => {
                    alert("Error: " + error);
                });
        postCount = postCount + 1;
        updatevalues();
        post_imgurl = undefined;
        text.value = "";
}
function updatevalues() {
        set(ref(db, "PostCount/" + username), {
                postCount: postCount,
        })
                .then(() => {
                    alert("Data stored suxxessfully");
                })
                .catch((error) => {
                    alert("Error: " + error);
                });
}
function addPosts() {
        const Posts = ref(db, "Posts/");
        onValue(Posts, (snapshot) => {
                const data = snapshot.val();
                let post = [];
                Object.keys(data).forEach((key) => {
                    post.push({
                        name: key,
                        data: data[key],
                    });
                });
```

```
// console.log(post);
post.forEach(async (element) => {
    let authorDp =
            "https://uxwing.com/wp-content/themes/uxwing/download/peoples-avatars/no-profile-picture-icon.png",
        authorName = "Unknown";
    const author = ref(db, "Usernames/" + element.data.author);
    await new Promise((resolve) => {
        onValue(author, (snapshot) => {
            if (snapshot.exists()) {
                authorDp = snapshot.val().imgurl;
                authorName = snapshot.val().fullName;
            }
            resolve();
        });
    });
    let likeBtnPicSrc = "/Pngs/heart-unlike.png";
    // console.log(element.data.likes+"  "+username);
    if (element.data.likes.includes(username)) {
        likeBtnPicSrc = "/Pngs/heart-liked.png";
    }
    let likes = 0;
    if (element.data.likes.length > 0) {
        likes = element.data.likes.trim().split(" ").length;
    }
    if (likes < 0) likes = 0;
    let dislikes = 0;
    // console.log(element.data)
    if (element.data.dislikes.length > 0) {
        dislikes = element.data.dislikes.trim().split(" ").length;
    }
    if (dislikes < 0) dislikes = 0;
```

```
                    let postImgurl = element.data.imgurl,
                        date = element.data.date,
                        text = element.data.text,
                        noOfLikes = likes,
                        noOfDislikes = dislikes,
                        noOfComments = element.data.comments,
                        id = element.name;
                    insertpost(
                        authorDp,
                        authorName,
                        date, //"06:37 AM , 2 Dec",
                        postImgurl,
                        text, //"Coming Soon...",
                        noOfLikes, //37,
                        noOfDislikes, //2,
                        noOfComments, //58
                        id,
                        likeBtnPicSrc
                    );
                });
        });
}
// <------------------------------------------Linking Ends Here------------------------------------------>

const input = document.getElementById("img-input");
input.onchange = function (ev) {
        const file = ev.target.files[0]; // get the file
        const blobURL = URL.createObjectURL(file);
        const img = new Image();
        img.src = blobURL;
        img.onerror = function () {
                URL.revokeObjectURL(this.src);
                // Handle the failure properly
                // console.log("Cannot load image");
        };
        img.onload = function () {
                URL.revokeObjectURL(this.src);
                const [newWidth, newHeight] = calculateSize(img, MAX_WIDTH, MAX_HEIGHT);
                const canvas = document.createElement("canvas");
                canvas.width = newWidth;
                canvas.height = newHeight;
                const ctx = canvas.getContext("2d");
                ctx.drawImage(img, 0, 0, newWidth, newHeight);
                canvas.toBlob(
                    (blob) => {
                        // Handle the compressed image. es. upload or save in local state
                        //console.log(blob);
                        imageUploaded(blob);
                    },
                    MIME_TYPE,
                    QUALITY
                );
                document.getElementById("root").innerHTML = "";
                document.getElementById("root").append(canvas);
        };
};
function calculateSize(img, maxWidth, maxHeight) {
        let width = img.width;
        let height = img.height;
        // calculate the width and height, constraining the proportions
        if (width > height) {
                if (width > maxWidth) {
                    height = Math.round((height * maxWidth) / width);
                    width = maxWidth;
```

```
                }
        } else {
                if (height > maxHeight) {
                        width = Math.round((width * maxHeight) / height);
                        height = maxHeight;
                }
        }
        return [width, height];
}
// Utility functions for demo purpose
//this funtion will be removed later
let base64String = "";
function imageUploaded(myfile) {
        var file = myfile;
        var reader = new FileReader();
        reader.onload = function () {
                base64String = reader.result.replace("data:", "").replace(/^.+,/, "");
                post_imgurl = "data:image/png;base64," + base64String;
        };
        reader.readAsDataURL(file);
}

const dbref = ref(db);
window.onload = async () => {
        await get(child(dbref, "loggedInuser/")).then((snapshot) => {
                if (snapshot.exists()) {
                        username = snapshot.val().username;
                }
        });
        get(child(dbref, "PostCount/" + username)).then((snapshot) => {
                if (snapshot.exists()) {
                        postCount = snapshot.val().postCount;
        }
    });
    SelectData();
    addPosts();
};
let search_btn = document.getElementById("search_btn");
let search_input = document.getElementById("search_input");
search_btn.onclick = () => {
    let searchUsername = search_input.value;
    if (searchUsername.length > 0) {
        const searchedUser = ref(db, "Usernames/" + searchUsername);
        onValue(searchedUser, (snapshot) => {
            {
                if (snapshot.exists()) {
                    let dp = snapshot.val().imgurl,
                        name = snapshot.val().fullName,
                        email = snapshot.val().email,
                        bio = snapshot.val().bio,
                        friends = snapshot.val().friends,
                        friendRequests = snapshot.val().friendRequests,
                        status = friends.includes(username),
                        id = searchUsername;
                    let friendShipStatus = "Add Friend";
                    if (status == true) {
                        friendShipStatus = "You are Friends";
                    } else if (friendRequests.includes(username)) {
                        friendShipStatus = "Cancel Request";
                    } else if (searchUsername == username) {
                        friendShipStatus = "Its You";
                    }
                    showSearchedUser(
                        dp,
                                                name,
                                                searchUsername,
                                                email,
                                                bio,
                                                friendShipStatus,
                                                id
                                        );
                                } else {
                                        let train = `<div><p>No Such User Found</p></div>`;
                                        document.getElementById("extra-features").innerHTML = train;
                                }
                        }
                });
        }
};

function showSearchedUser(
        dp,
        fullName,
        username,
        email,
        bio,
        friendShipStatus,
        id
) {
        let train = `<div id="userProfile">
          <img id="userBanner" src="/Pngs/banner.jpg" alt="">
          <img id="userProfilePic" src="${dp}" alt="">
          <h4 id="fullname"><b>${fullName}</b></h4>
          <div id="userDetails">
            <div class="userCard">
              <h6><b> Username : &nbsp&nbsp</b></h6>
```

```
                        <h6>${username}</h6>
                    </div>
                    <hr>
                    <div class="userCard">
                        <h6><b> Email :&nbsp&nbsp</b></h6>
                        <h6>${email}</h6>
                    </div>
                    <hr>
                    <div class="userCard">
                        <h6><b> Bio :&nbsp&nbsp</b></h6>
                        <h6>${bio}</h6>
                    </div>
                </div>
                <div id="${id}" onclick="addFriend(${id})" class="setting-card" style="text-align:
center;">${friendShipStatus}</div>
            </div>`;
            document.getElementById("extra-features").innerHTML = train;
}
window.addFriend = async function addFriend(id) {
            if (id.innerHTML == "Its You") return;
            if (id.innerHTML == "Add Friend") {
                        console.log(id);
                        let friendUsername = id.id;
                        const friendRef = ref(db, "Usernames/" + friendUsername);
                        let friendRequests = "";
                        await new Promise((resolve, reject) => {
                                    onValue(friendRef, (snapshot) => {
                                                friendRequests = snapshot.val().friendRequests;
                                                resolve();
                                    });
                        });
                        if (friendRequests.includes(username)) {
```

```
            alert("Request Already Exists");
        } else {
            if (friendRequests.length > 0) friendRequests += " ";
            friendRequests += username;
            console.log(friendRequests);
            await update(child(dbref, "Usernames/" + friendUsername), {
                friendRequests: friendRequests,
            });
        }
        id.innerHTML = "Cancel Request";
        return;
    }
    if (id.innerHTML == "Cancel Request") {
        let friendUsername = id.id;
        const friendRef = ref(db, "Usernames/" + friendUsername);
        let friendRequests = "";
        await new Promise((resolve, reject) => {
            onValue(friendRef, (snapshot) => {
                friendRequests = snapshot.val().friendRequests;
                resolve();
            });
        });
        //jugad it should be friendRequests= friendRequests.replace(username+" ", "");
        //possible error : mateen and mateen123 can be  different useranmes
        friendRequests = friendRequests.replace(username, "");
        friendRequests = friendRequests.replace("  ", " ");
        console.log(username + "  " + friendRequests);
        await update(child(dbref, "Usernames/" + friendUsername), {
            friendRequests: friendRequests,
        });
        id.innerHTML = "Add Friend";
        return;
    }
};
let view_friends = document.getElementById("view_friends");
view_friends.onclick = () => {
    let extraFeatures = document.getElementById("extra-features");
    extraFeatures.innerHTML = `<h4 id="extra_header">Friend List</h4>`;
    const friendRef = ref(db, "Usernames/" + username);
    let friends = "";
    onValue(friendRef, (snapshot) => {
        friends = snapshot.val().friends;
    });
    let friendArr = [];
    if (friends.length > 0) friendArr = friends.trim().split(" ");
    friendArr.forEach((element) => {
        const friendRef2 = ref(db, "Usernames/" + element);
        onValue(friendRef2, (snapshot) => {
            ShowFriendCard(snapshot.val().imgurl, snapshot.val().fullName);
        });
    });
};
let view_friends_requests_notifi = document.getElementById(
    "friend_requests_noti"
);
view_friends_requests_notifi.onclick = () => {
    viewFriendsRequests();
};
let view_friends_requests = document.getElementById("friend-requests");
view_friends_requests.onclick = () => {
    viewFriendsRequests();
};
function viewFriendsRequests() {
    let extraFeatures = document.getElementById("extra-features");
    extraFeatures.innerHTML = `<h4 id="extra_header">Friend Requests</h4>`;
    const friendRef = ref(db, "Usernames/" + username);
    let friendRequests = "";
    onValue(friendRef, (snapshot) => {
        friendRequests = snapshot.val().friendRequests;
    });
    let friendArr = [];
    if (friendRequests.length > 0) friendArr = friendRequests.split(" ");
    else {
        extraFeatures.innerHTML += `<h5 >No Pending Requests</h5>`;
    }
    friendArr.forEach((element) => {
        const friendRef2 = ref(db, "Usernames/" + element);
        onValue(friendRef2, (snapshot) => {
            ShowPersonCard(
                snapshot.val().imgurl,
                snapshot.val().fullName,
                snapshot.val().username
            );
        });
    });
}
let createPost_btn = document.getElementById("createPost_btn");
// createPost_btn.addEventListener('click',show_newpost_popup())
let cancle_btn = document.getElementById("cancle_btn");
// cancle_btn.addEventListener('click',show_newpost_popup())
createPost_btn.onclick = () => {
    show_newpost_popup();
    clearPostDetails();
};
cancle_btn.onclick = () => {
    show_newpost_popup();
```

```
};
function show_newpost_popup() {
            let btn = document.getElementById("new-post-popup");
            btn.classList.toggle("show-newpost-popup");
}

let post_btn = document.getElementById("post_btn");
post_btn.onclick = () => {
            storePostInDataBase();
            show_newpost_popup();
            clearPostDetails();
};
function clearPostDetails() {
            document.getElementById("root").innerHTML = "";
            document.getElementById("caption-area").value = "";
            document.getElementById("img-input").value = null;
}
// let loadPost_btn = document.getElementById("load_posts");
// loadPost_btn.onclick = () => {
//          addPosts();
// };

//                      Adding it to insert friend on btn click

// let viewFriends = document.getElementById("view_friends");
// viewFriends.onclick = () => {
//          let extraFeatures = document.getElementById("extra-features");
//          extraFeatures.innerHTML = `<h4 id="extra_header">Friend List</h4>`;
//          let i = 0;
//          for (i = 0; i < 10; i++) {
//                      ShowFriendCard();
//          }
// };
function ShowFriendCard(dp, name) {
            let extraFeatures = document.getElementById("extra-features");
            extraFeatures.innerHTML += `<div id="person_card">
            <a href="#" class="no-link">
                <div id="view-profile" >
                    <img id="profile_pic" class="profile_pic"
                        src="${dp}"
                        alt="">
                    <h6 style="text-align: center;">${name}</h6>
                </div>
            </a>
</div>`;
            // Added this so when user clicks on the profile of his friend it will show his profile
            // let get = document.getElementById("person_card");
            // get.onclick = () => {
            //  showSearchedUser(dp, name, "n", "@", "cool", true, "id");
            //  Call function you call when someone search for username
            //  complete this by matching the user name with the data on db
            // };
}

function ShowPersonCard(dp, name, username) {
            let extraFeatures = document.getElementById("extra-features");
            extraFeatures.innerHTML += `<div id="person_card"'>
            <a href="#" class="no-link">
                <div id="view-profile">
                    <img id="profile_pic" class="profile_pic"
                        src="${dp}"
                        alt="">
                    <h6 style="text-align: center;">${name}</h6>
                </div>
            </a>
            <div id="add_friend" class="flex-row">
                <div id="${username}" onclick="acceptRequest(${username})" class="accept_request"
style="width: 50%;">Accept</div>
                <div id="${username}"  onclick="rejectRequest(${username})" class="declineRequest"
">Reject</div>

            </div>
</div>`;
}
window.rejectRequest = async function (user) {
            let requestedUsername = user[0].id;
            console.log(requestedUsername);

            let friendRef = ref(db, "Usernames/" + username);
            let friendRequests = "";
            await new Promise((resolve, reject) => {
                onValue(friendRef, (snapshot) => {
                    friendRequests = snapshot.val().friendRequests;
                    resolve();
                });
            });
            //jugad it should be friendRequests= friendRequests.replace(username+" ", "");
            //possible error : mateen and mateen123 can be  different useranmes
            friendRequests = friendRequests.replace(requestedUsername, "");
            friendRequests = friendRequests.replace("  ", " ");
            friendRequests = friendRequests.trim();
            //console.log(username + "  " + friendRequests);
            await update(child(dbref, "Usernames/" + username), {
                friendRequests: friendRequests,
            });
            viewFriendsRequests();
};
```

```
window.acceptRequest = async function (user) {
    let requestedUsername = user[0].id;
    console.log(requestedUsername);

    let friendRef = ref(db, "Usernames/" + username);
    let friendRequests = "",
        friends = "";
    await new Promise((resolve, reject) => {
        onValue(friendRef, (snapshot) => {
            friendRequests = snapshot.val().friendRequests;
            friends = snapshot.val().friends;
            resolve();
        });
    });
    //jugad it should be friendRequests= friendRequests.replace(username+" ", "");
    //possible error : mateen and mateen123 can be  different useranmes
    friendRequests = friendRequests.replace(requestedUsername, "");
    friendRequests = friendRequests.replace("  ", " ");
    friendRequests = friendRequests.trim();
    if (friends.length > 0) friends += " ";
    friends += requestedUsername;
    //console.log(username + "  " + friendRequests);
    await update(child(dbref, "Usernames/" + username), {
        friendRequests: friendRequests,
        friends: friends,
    });
    //get friends of request and add in them
    friends = "";
    friendRef = ref(db, "Usernames/" + requestedUsername);
    await new Promise((resolve, reject) => {
        onValue(friendRef, (snapshot) => {
            friends = snapshot.val().friends;
            resolve();
        });
    });
    friends = friends.trim();
    if (friends.length > 0) friends += " ";
    friends = friends.replace("  ", " ");
    friends += username;
    await update(child(dbref, "Usernames/" + requestedUsername), {
        friends: friends,
    });

    //code
    viewFriendsRequests();
};

//  notification

let notiBtn = document.getElementById("notification_btn");
notiBtn.onclick = () => {
    let extraFeatures = document.getElementById("extra-features");
    extraFeatures.innerHTML = `<h4 id="extra_header">Notifications</h4>`;
    extraFeatures.innerHTML += `<div id="notification">
    <div id="notification-img">
                <img src="/Pngs/pic.jpg" alt="">
            </div>
            <div id="text-content">
                Ali liked your photo.
            </div>
        </div > `;
};

//  /   /   /   /   /   /   /   /   /   /   /   /
```

```
function insertpost(
        dp,
        name,
        date,
        img,
        text,
        likes,
        dislikes,
        comments,
        id,
        likeBtnPicSrc
) {
        let train = `<div class="post" >
    <div class="post-details">
      <img src=${dp} class="profile_pic"
        alt="">
      <div class="name-time">
        <h4>${name}</h4>
        <p>${date}</p>
      </div>
    </div>
    <div class="post-data" >
      <p>${text}</p>
      <img src=${img} ondblclick="likePost(${id})"
        alt="">


    </div>
    <div class="reaction-section">
      <div class="reaction-section-details">
        <p id="${id}+likes">${likes} Likes</p>
        <p>${dislikes} Dislikes</p>
      </div>
      <div class="reaction-option">
        <div class="love-reaction" onclick="likePost(${id})">
          <img id="${id}" src="${likeBtnPicSrc}"  alt="">
          <p>Love</p>
        </div>
        <div class="comment-section">
          <img src="/Pngs/dislike_btn.png" alt="">
          <p>Dislike</p>
        </div>
      </div>
    </div>
  </ > `;
        document.getElementById("posts_section").innerHTML += train;
}
window.likePost = async function likePost(post) {
        console.log(post);
        let id = post.id;
        if (id === undefined) {
                id = post[0].id;
                post = post[0];
        }
        let likes = "12";
        console.log(id);
        let Ref = ref(db, "Posts/" + id);
        await new Promise((resolve) => {
                onValue(Ref, (snapshot) => {
                        if (snapshot.exists()) {
                                likes = snapshot.val().likes;
                        }
                        resolve();
                });
        });
        if (likes.includes(username)) {
                likes = likes.replace(username + " ", "");
                post.src = "/Pngs/heart-unlike.png";
        } else {
                likes += username + " ";
                post.src = "/Pngs/heart-liked.png";
        }
        updateLikes(likes, id);
        // if (post.src == "http://127.0.0.1:5501/Pngs/heart-liked.png") {
        //      post.src = "/Pngs/heart-unlike.png";
        //      return;
        // }
        // post.src = "/Pngs/heart-liked.png";
};
function updateLikes(likes, id) {
        update(ref(db, "Posts/" + id), {
                likes: likes,
        });
        console.log(likes);
        let count = likes.trim().split(" ").length;
        console.log(count);
        if (likes == "") count = 0;
        document.getElementById(id + "+likes").innerHTML = count + " Likes";
}
let changeSetting = document.getElementById("change_setting");
changeSetting.addEventListener("click", changeSettingClick);
function changeSettingClick() {
        window.location.href = "../Pages/setting.html?" + username;
}
```

<---END-->