Maulana Abul Kalam Azad University of Technology

Contact Vault

A PROJECT REPORT

Submitted By

BASUDEB PAUL CHOWDHURY

Masters of Computer Application (M.C.A)

Department name: Information Technology

Roll Number: 10071023017

Reg. Number: 231000510711

Paper Code: MCAS281

Paper Name: Web Design and Development

Year: 2023-25

CONTACT VAULT

SAVE YOUR CONTACTS

TABLE OF CONTENT

ABSTRACT

The Contact Vault is a web-based application. This web site provides users an easy way to search and find any phone number that "Contact Vault" has information about it. Also Contact Vault provides a feature to store the phone number of a user along with much more information that might be use later.

The main objective of our web site is facilitating users to provide an online contact storage with a feature to identify unknown calls.

INTRODUCTION

Project Name: Contact Vault

Description: Contact Vault is an application designed to manage contacts efficiently. It allows users to search for contacts, display contact details, and find information about unknown phone numbers from a global database.

Project Objectives

- Management: Provide a seamless way to add, edit, and delete contacts.
- Search Functionality: Enable users to quickly search for contacts by name or phone number.
- Global Search: Allow users to search for information about unknown phone numbers.
- User Authentication: Ensure secure access to the application through login and logout mechanisms.
- Responsive Design: Ensure the application works well on both desktop and mobile devices.
- Profile Images: Allow users to upload and display profile pictures for their contacts.
- Session Management: Securely maintain user sessions.

FEATURES

i.   Contact Management
- Users can add, edit, and delete their contacts.
- Profile pictures can be uploaded and displayed for each contact.

ii.   Search Functionality
- Users can search for contacts by entering their name or phone number.
- The application displays contact details based on the search query.

iii.   Global Search
- Users can search for unknown phone numbers to find information from a global database.
- This feature helps users identify unknown callers or contacts.

iv.   User Authentication
- Secure login and logout mechanisms ensure that only authorized users can access the application.
- Sessions are managed to keep users logged in securely.

v.   Responsive Design
- The application uses Bootstrap 5 to provide a responsive design.
- The interface adapts to different screen sizes, ensuring a good user experience on both desktop and mobile devices.

USED TECHNOLOGIES

Frontend: HTML, CSS, Bootstrap 5, JavaScript, jQuery

Backend: PHP

Database: MySQL

Server: XAMPP (for local development)

Version Control: Git and Git Hub

IDE: VS Code

Browser: Chrome

SYSTEM ARCHITECTURE AND REQUIREMENTS

Client-Server Architecture:
The application follows a client-server architecture where the client (browser) interacts with the server (PHP scripts and MySQL database) to perform various operations. The client sends requests to the server, which processes them and returns the appropriate response.

Requirements to use this application:

i.   **Hardware Requirement:**
- A browser supported device.
- Internet Connection.

ii.   **Software Requirement:**
- Any internet browsing application

PROJECT IMPLEMENTETION USING SPIRAL MODEL

SPIRAL MODEL Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using the spiral model. The Spiral Model is a software development life cycle (SDLC) model that provides a systematic and iterative approach to software development. It is based on the idea of a spiral, with each iteration of the spiral representing a complete software development cycle, from requirements gathering and analysis to design, implementation, testing, and maintenance.
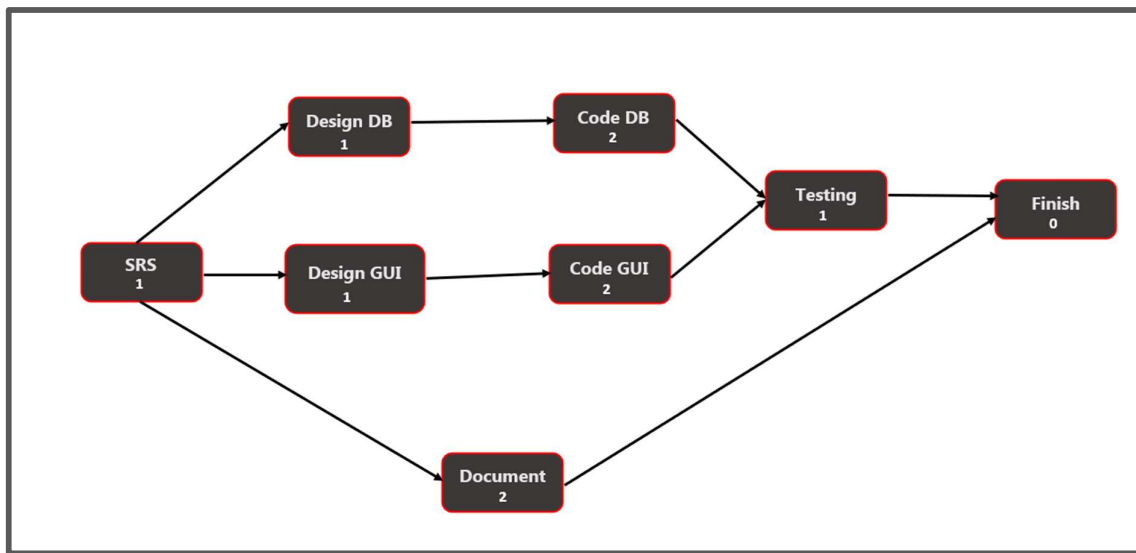
SYSTEM PLANNING

System Planning Project Planning: The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, cost, and schedule, These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses, In addition, estimates should attempt to define best case worst case scenarios that project outcomes can be bounded, The planning objective is achieved through a process of information discovery that leads to reasonable estimates.

Project Scheduling: Project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. The number of basic principles guide the project scheduling is as follows:

- **Compartmentalization:** The project must be compartmentalized into a number of manageable activities and tasks. To accomplish compartmentalization, both the product and the process are decomposed.
- **Interdependence:** The interdependence of each compartmentalized activity or task must be determined. Some tasks must occur in-sequence while others can occur in parallel. Some activities cannot commence until the work product produced by another is available. Other activities can occur independently.
- **Time allocation:** Each task is scheduled must be allocated some number of work units. In addition, each task must be assigned a start date and a completion date.

- **Defined outcomes:** Every task that is scheduled should have a defined outcome. For software projects the outcome is normally a work product or a part of a work product.
- **Defined milestones:** Every tasks or group of tasks should be associated with a project milestone. A milestone is accomplished when one or more work products has been reviewed for quality and has been approved.

PERT CHART

Methodology

Development Process

1. **Research and Planning**: Identifying requirements and defining the project scope.

2. **Design**: Creating the system architecture and database schema.

3. **Development**: Implementing the front-end and back-end functionalities.

4. **Testing**: Conducting unit testing, integration testing, and user acceptance testing.

5. **Deployment**: Setting up the server environment and deploying the application.

USER MANUAL

Installation

1. Download and install XAMPP.

2. Clone the project repository.

3. Import the database schema.

4. Configure the connection settings in connection.php.

5. Start the Apache and MySQL services in XAMPP.

6. Access the application via the browser.

Usage

1. **Login**: Use your phone number and password to log in.

2. **Add Contacts**: Navigate to the "Add" section to add new contacts.

3. **Search Contacts**: Use the search functionality to find contacts by name or phone number.

4. **Global Search**: Enter unknown phone numbers to search the global database.

5. **Manage Contacts**: Edit or delete contacts as needed.

PROJECT MANAGEMENT

Timeline

- **Research and Planning**: 1 days

- **Design**: 2 days

- **Development**: 3 days

- **Testing**: 1 days

- **Deployment**: 1 week

GNATT CHART
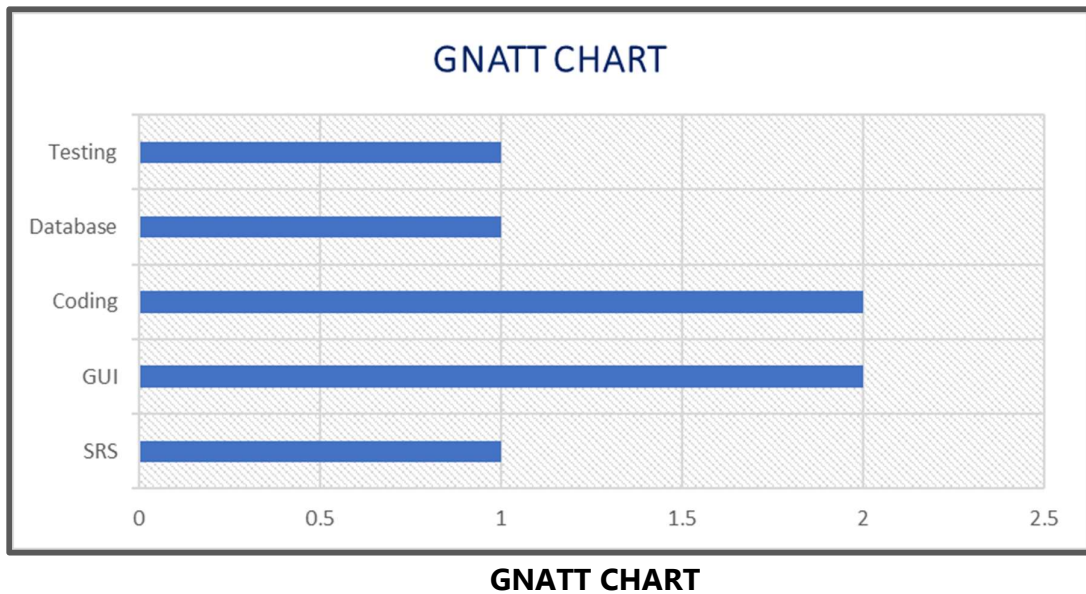
DATABASE DESIGN

There are mainly 3 tables are used in our application:

i. **user_signup_tbl:** To store data about the application user.
ii. **add_to_contacts_tbl:** To save contacts that user wish to store.
iii. **global_contacts:** To save all the contact details those users are saved on database along with the data that saved during user signup.

user_signup_tbl:

Column Name	Data Type	NULL	Default	Constraints
user_name	varchar(30)	NO		
user_phone	varchar(13)	NO		Primary
user_email	varchar(50)	NO		Unique
user_password	varchar(20)	NO		
user_subscription	tinyint(1)	YES	NULL	
user_birthdate	date	YES	NULL	
user_address	varchar(20)	YES	NULL	
date_of_join	datetime	NO	current_timestamp()	
user_img	varchar(255)	YES	NULL	

add_to_contacts_tbl

Column Name	Data Type	NULL	Default	Constraints
contact_id	varchar(26)	NO		Primary
user_phno	varchar(13)	NO		Foreign
contact_phno	varchar(13)	NO		
contact_email	varchar(40)	NO		
contact_name	varchar(50)	NO		
contact_address	varchar(50)	NO		
contact_dob	date	YES	NULL	
contact_img	varchar(255)	YES	NULL	

global_contacts:

Column Name	Data Type	NULL	Default	Constraints
contact_phone	varchar(13)	NO		Primary
contact_name	varchar(255)	NO		
contact_email	varchar(255)	YES	NULL	
contact_address	text	YES	NULL	
contact_dob	date	YES	NULL	
contact_img	varchar(255)	YES	NULL	

CODE SNIPPETS

Connection File code:

```php
<?php
$servername = "localhost";
$username = "root";
$pass = "";
$database = "contact_vault";
$con = mysqli_connect($servername, $username, $pass, $database);
```

Queries Used In Application:

```php
<?php
function userSignUp($name, $phno, $email, $pass)

{

    return "INSERT INTO `user_signup_tbl` (`user_name`, `user_phone`, `user_email`, `user_password`) VALUES ('$name', '$phno', '$email', '$pass')";

}

function isUserExsist($phno, $email)

{

    return "SELECT `user_phone`, `user_email` FROM `user_signup_tbl` WHERE  `user_phone` = '$phno' OR `user_email` = '$email'";

}

function checkLoginDetails($phno)

{

    return "SELECT `user_password`  FROM `user_signup_tbl` WHERE `user_phone` = $phno";

}

function checkContactsExist($cid)

{

    return "SELECT `contact_id` FROM `add_to_contacts_tbl` WHERE `contact_id` = $cid";

}


function getUserDetails($phno)

{

    return "SELECT * FROM `user_signup_tbl` WHERE `user_phone` = $phno";

}

function addToContacts($contact_id, $user_phone, $email, $phno, $name, $add, $dob, $imagePath)

{

    return "INSERT INTO add_to_contacts_tbl (contact_id, user_phno, contact_email, contact_phno, contact_name, contact_address, contact_dob, contact_img) VALUES ('$contact_id', '$user_phone', '$email', '$phno', '$name', '$add', '$dob', '$imagePath')";

}

function addToGlobalContacts($phno, $name, $email, $add, $dob)
```

```php
{
    return "INSERT INTO `global_contacts`(`contact_phone`, `contact_name`, `contact_email`,
`contact_address`, `contact_dob`) VALUES ('$phno','$name','$email','$add','$dob')";
}
function getGlobalContacts($phno)
{
    return "SELECT * FROM global_contacts WHERE contact_phone = '$phno'";
}
function updateGlobalContactName($updated_name, $email, $add, $phno)
{
    return "UPDATE global_contacts SET contact_name = '$updated_name', contact_email =
'$email', contact_address = '$add' WHERE contact_phone = '$phno'";
}
function searchGlobalContactByPhno($phno)
{
    return "SELECT * FROM `global_contacts` WHERE `contact_phone` = $phno";
}
function deleteContactFromMe($contact_id)
{
    return "DELETE FROM add_to_contacts_tbl WHERE contact_id = '$contact_id'";
}


function selectContactFromMe($contact_id)
{
    return "SELECT * FROM `add_to_contacts_tbl` WHERE `contact_id` = '$contact_id'";
}
function updateUserProfile($email, $name, $address, $dob, $imagePath, $user_phone)
{
    return "UPDATE user_signup_tbl SET user_email = '$email', user_name = '$name', user_address
= '$address', user_birthdate = '$dob', user_img = '$imagePath' WHERE user_phone =
'$user_phone'";
}
function countProfileContacts($user_phno)
```

```php
{
    return "SELECT COUNT(*) AS row_count FROM `add_to_contacts_tbl` WHERE `user_phno` = '$user_phno'";
}

function updateMyContact($email, $name, $add, $dob, $img, $phno)
{
    return "UPDATE `add_to_contacts_tbl` SET `contact_email` = '$email', `contact_name` = '$name', `contact_address` = '$add', `contact_dob` = '$dob', `contact_img` = '$img' WHERE `contact_phno` = '$phno'";
}
```

Login Frontend Code:

```php
<?php
session_start();
$msg = "";
if (isset($_SESSION['msg'])) {
    $msg = $_SESSION['msg'];
    unset($_SESSION['msg']);
}
?>
<!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">
    <link rel="stylesheet" href="../css/bootstrap.css">
    <link rel="stylesheet" href="style.css">
    <link rel="shortcut icon" href="../img/logo-round.png" type="x-icon">
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.6.0/dist/umd/popper.min.js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    <title>Contact Vault | Login</title>
</head>
<body>
    <?php if ($msg != ''): ?>
        <script>
            $(document).ready(function () {
                $('#myModal').modal('show');
            });
        </script>
```

```php
    <?php endif; ?>
  <div class="container d-flex justify-content-center align-items-center min-vh-100">
    <div class="row border rounded-5 p-3 bg-white shadow box-area">
      <div class="col-md-6 rounded-4 d-flex justify-content-center align-items-center flex-column left-box"
          style="background: #103cbe;">
        <div class="featured-image mb-3">
          <img src="http://localhost/GetMyContact/img/logo.png" class="img-fluid"
style="width: 250px;">
        </div>
        <p class="text-white fs-2" style="font-family: 'Courier New', Courier, monospace; font-weight: 600;">
          Login now</p>
        <small class="text-white text-wrap text-center"
          style="width: 17rem;font-family: 'Courier New', Courier, monospace;">Hello, user
login now on your
          id.</small>
      </div>
      <div class="col-md-6 right-box">
        <div class="row align-items-center">
          <div class="header-text mb-4">
            <h2>Welcome, Back</h2>
          </div>
          <form action="http://localhost/GetMyContact/backends/login.php" method="post">
            <div class="input-group mb-3">
              <input type="number" class="form-control form-control-lg bg-light fs-6"
                placeholder="Enter phone number" name="phno" required>
            </div>
            <div class="input-group mb-5">
              <input type="password" name="pass" class="form-control form-control-lg bg-light fs-6"
                placeholder="Enter password" required>
            </div>
```

```html
                <div class="input-group mb-3">

                    <input class="btn btn-lg btn-primary w-100 fs-6" type="submit" name="Login"
value="Login">

                </div>

            </form>

            <div class="row">

                <small>Don't have an account? <a
href="http://localhost/GetMyContact/html/signup.php">Signup

                    now</a></small>

            </div>

        </div>

      </div>

    </div>

    <?php if ($msg != ''): ?>

      <div class="modal fade" id="myModal" tabindex="-1" aria-
labelledby="messageModalLabel" aria-hidden="true"

          data-backdrop="static">

        <div class="modal-dialog modal-dialog-centered">

          <div class="modal-content">

            <div class="modal-header">

              <h5 class="modal-title text-primary" id="messageModalLabel">Contact Vault
says....</h5>

            </div>

            <div class="modal-body">

              <?php echo "$msg"; ?>

            </div>

            <div class="modal-footer">

              <a href="http://localhost/GetMyContact/html/login.php" class="btn btn-success
text-white">Login</a>

              <a href="http://localhost/GetMyContact/html/signup.php"

                class="btn btn-primary text-white">Signup</a>
```

```
                </div>
            </div>
          </div>
        </div>
    <?php endif; ?>
    <?php if ($msg != ''): ?>
      <script>
        $(document).ready(function () {
          $('#myModal').modal('show');
        });
      </script>
    <?php endif; ?>
</body>
</html>
```

Login Backend Code

```php
<?php
include 'connection.php';
include 'query.php';
include 'message.php';
if (isset($_POST['Login'])) {
    session_start();
    $phno = $_POST['phno'];
    $pass = $_POST['pass'];
    $qry = checkLoginDetails($phno);
    $result = mysqli_query($con, $qry);
    if (mysqli_num_rows($result) == 0) {
        $_SESSION["msg"] = "Your phone number is not register login again or signup.";
        header("Location: http://localhost/GetMyContact/html/login.php");
        exit();
    } else {
        $row = mysqli_fetch_assoc($result);
        $stored_pass = $row['user_password'];
        if ($pass == $stored_pass) {
            $_SESSION["isLoggedin"] = true;
            $_SESSION["phno"] = $phno;
            $qry = getUserDetails($phno);
            $result = mysqli_query($con, $qry);
            $userDetails = mysqli_fetch_assoc($result);
            $_SESSION['user_name'] = $userDetails['user_name'];
            $_SESSION['user_phone'] = $userDetails['user_phone'];
            $_SESSION['user_email'] = $userDetails['user_email'];
            $_SESSION['user_subscription'] = $userDetails['user_subscription'];
            $_SESSION['user_birthdate'] = $userDetails['user_birthdate'];
            $_SESSION['user_address'] = $userDetails['user_address'];
```

```php
            $_SESSION['date_of_join'] = $userDetails['date_of_join'];

            $_SESSION['user_img'] = $userDetails['user_img'];

            header("Location: http://localhost/GetMyContact/html/profile.php");

            exit();

        } else {

            $_SESSION["msg"] = "Incorrect password. Please try again.";

            header("Location: http://localhost/GetMyContact/html/login.php");

            exit();

        }

    }

}
```

Logout Code:

```php
<?php

session_start();

$_SESSION = array();

if (ini_get("session.use_cookies")) {

    $params = session_get_cookie_params();

    setcookie(session_name(),'',

        time() - 42000,

        $params["path"],

        $params["domain"],

        $params["secure"],

        $params["httponly"]

    );

}

session_destroy();


header("Location: http://localhost/GetMyContact/html/login.php");

exit();
```

SCREENSHOTS-Large Screen
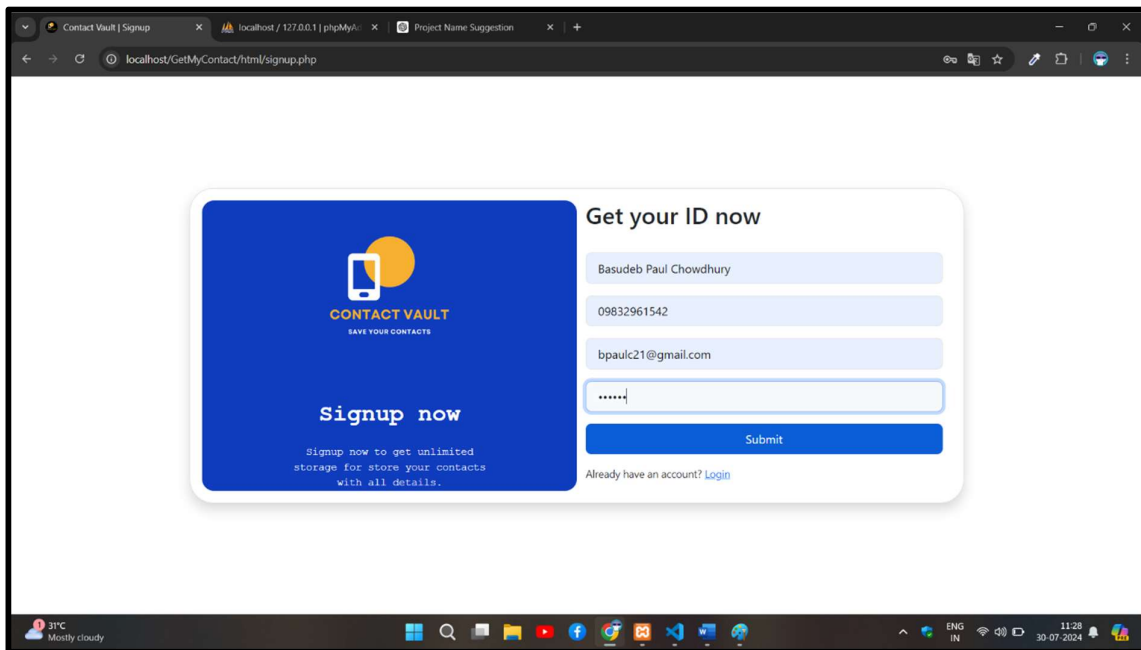
Signup page of a user.
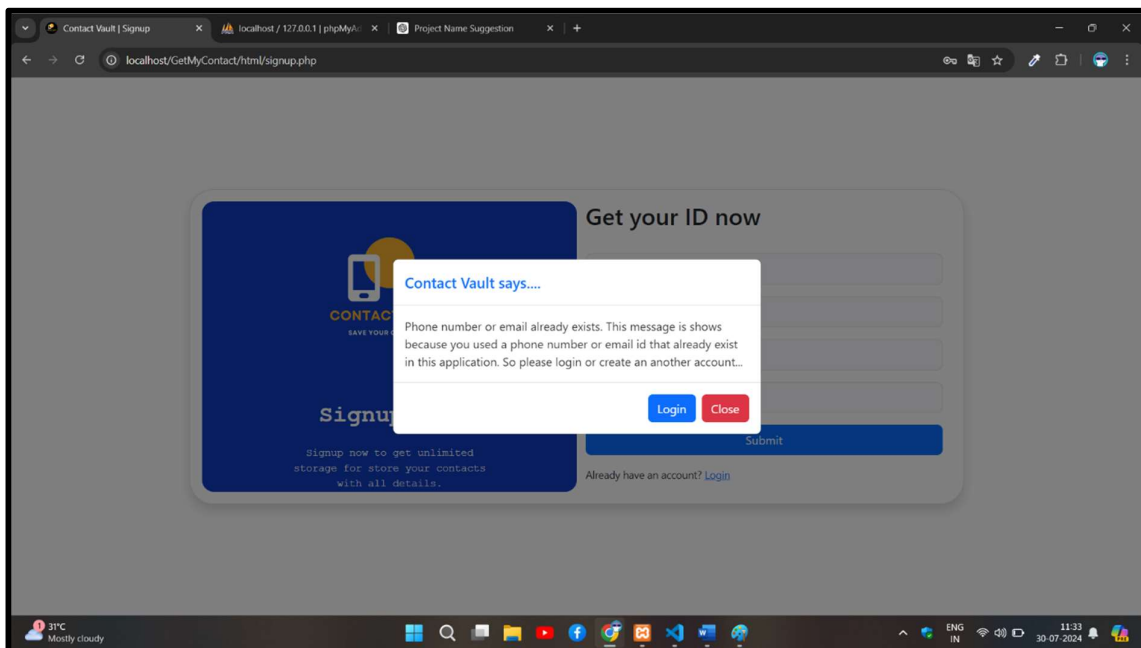
Signup page if user already exists.
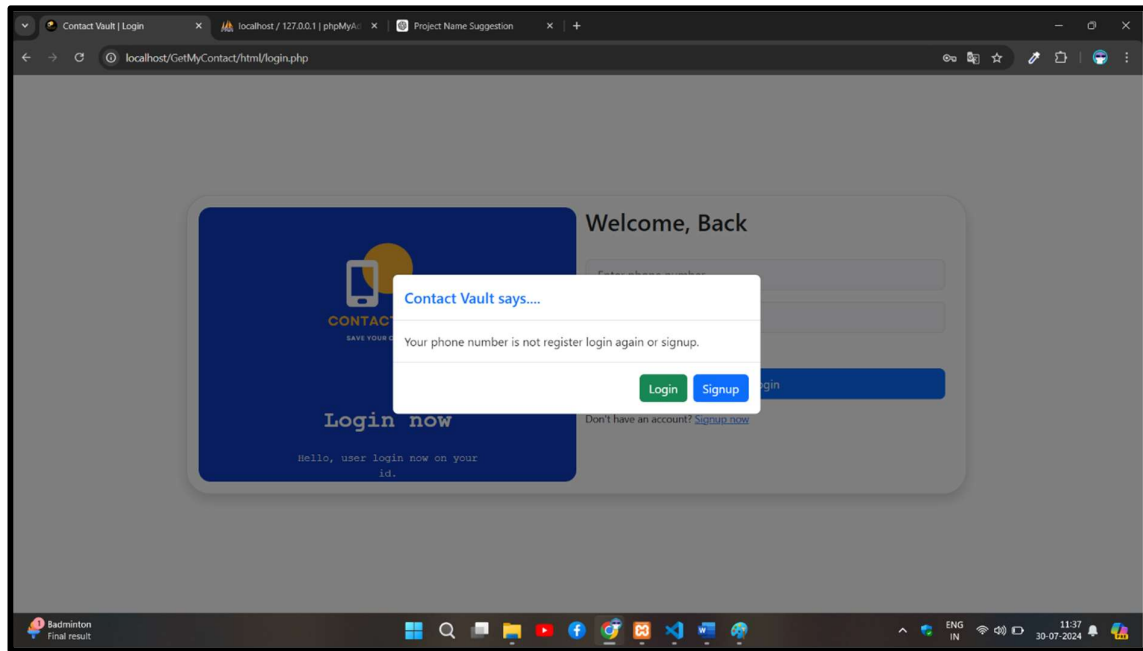
Login page if the user not exists.
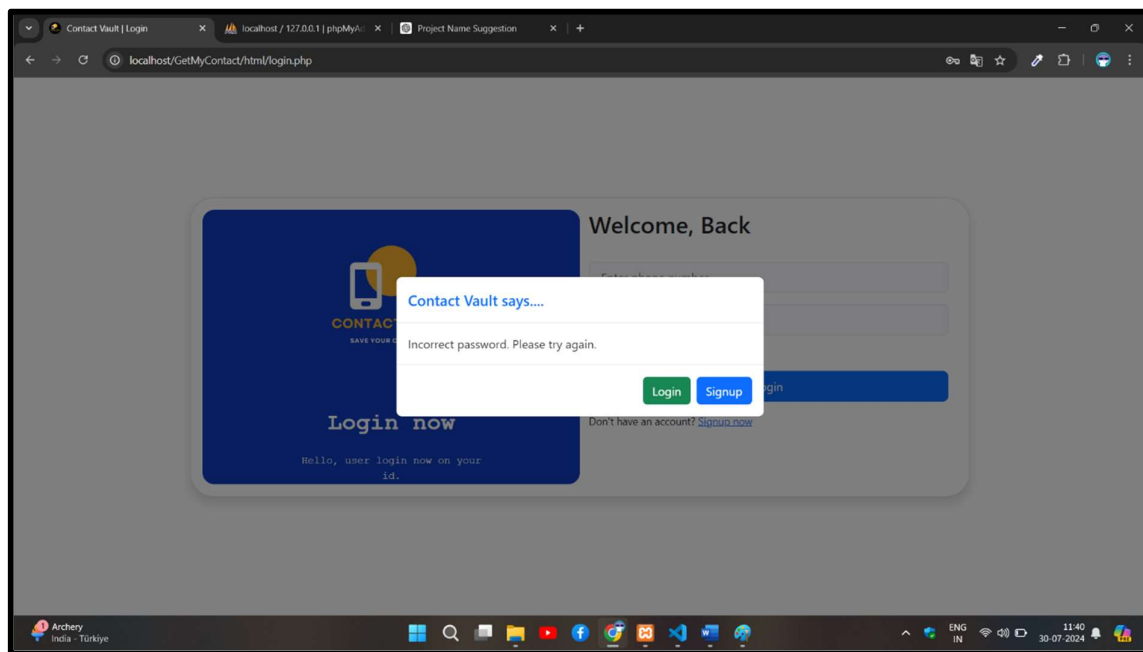
Login page if password is wrong.
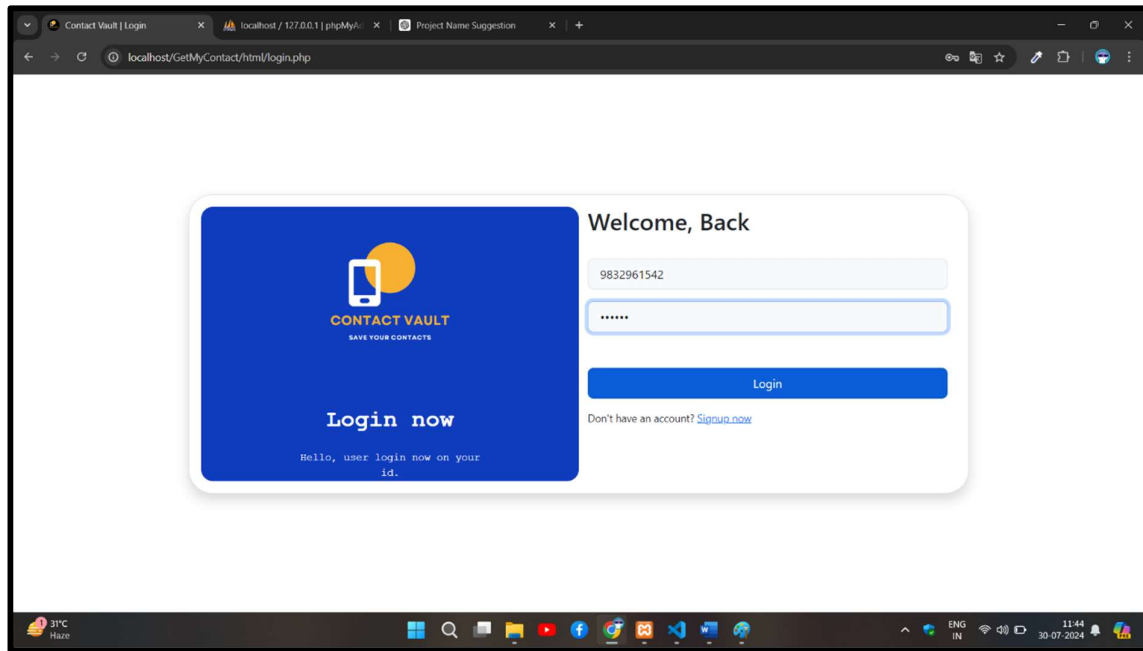
Login page.
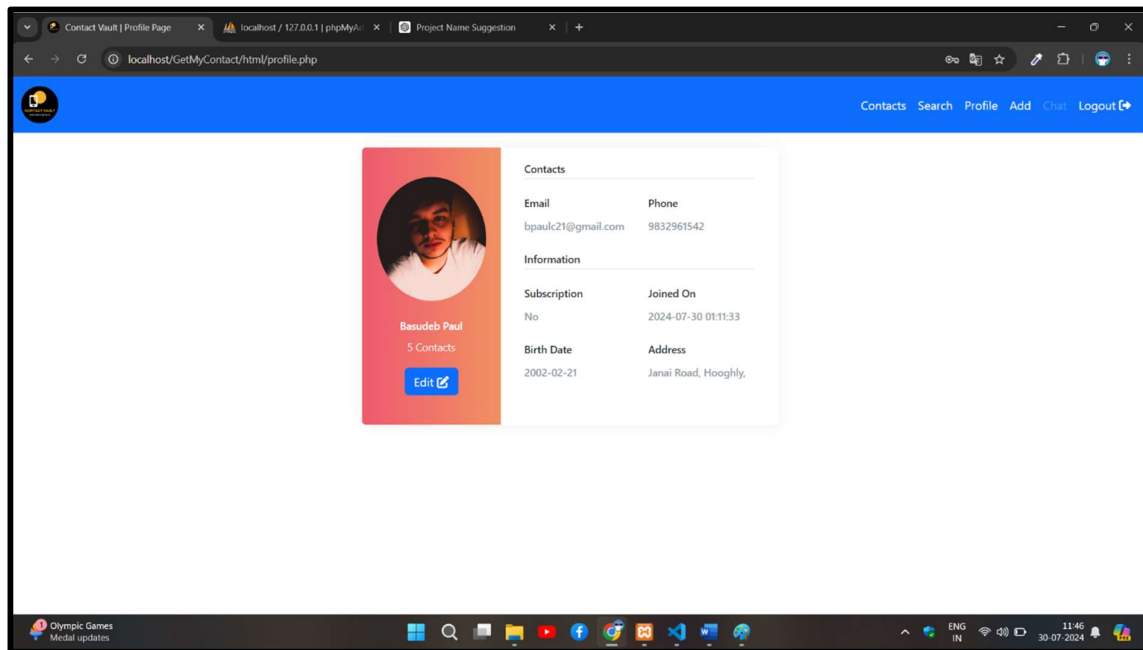
Profile page of the user who logged in.
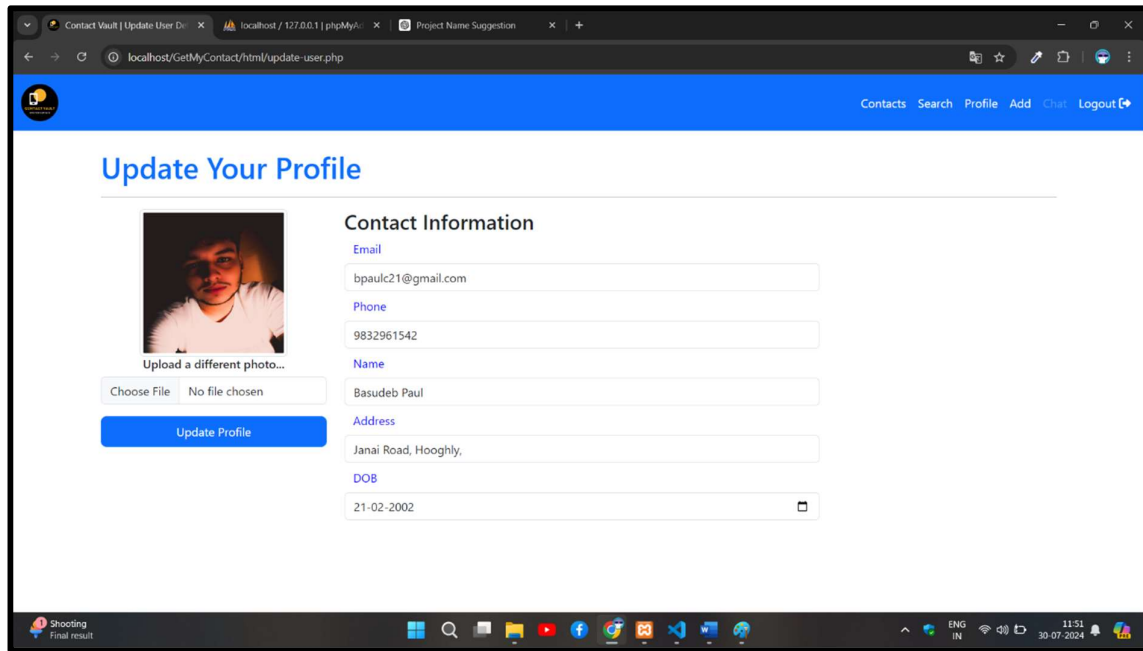
Profile update page of the user who logged in.
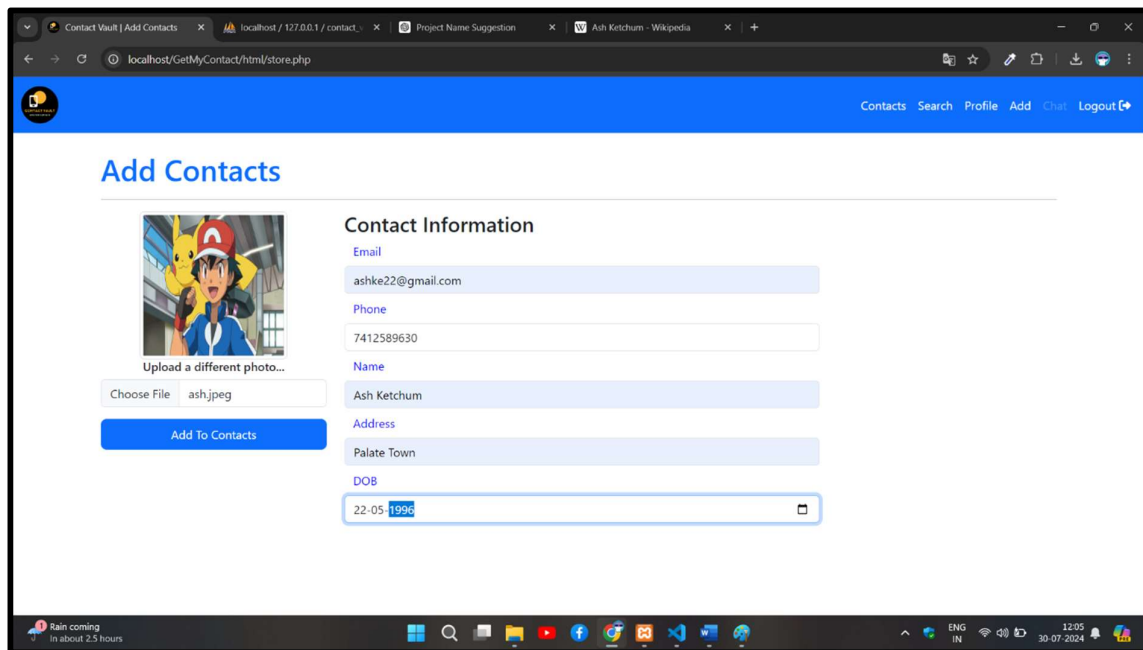
Add New Contact.
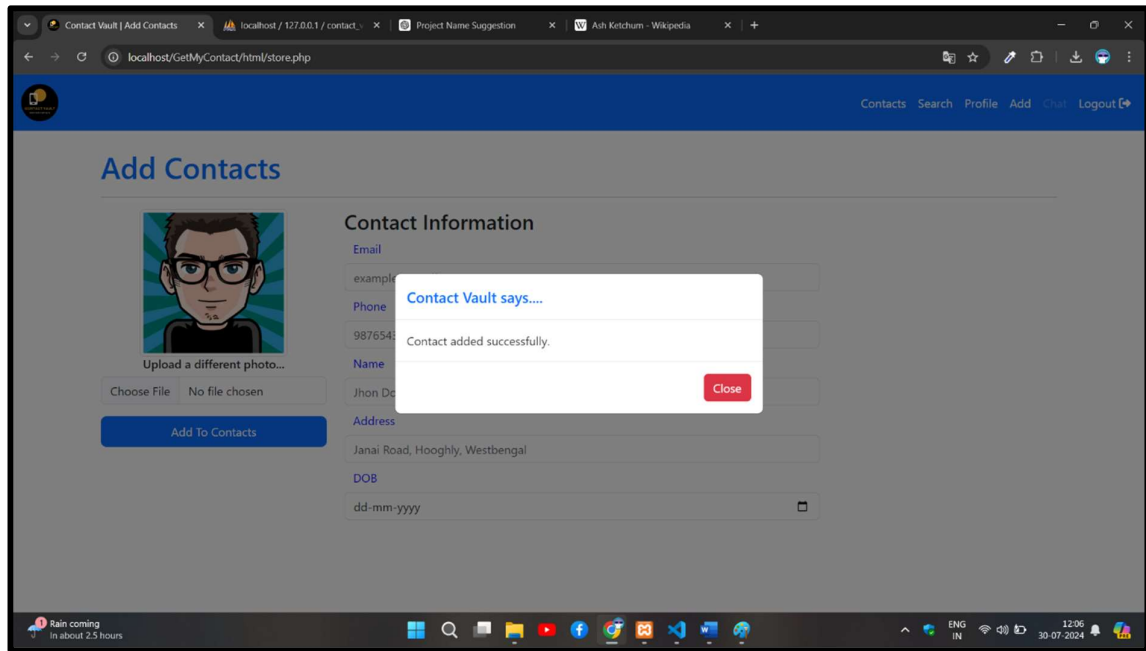
Contact successfully added message.
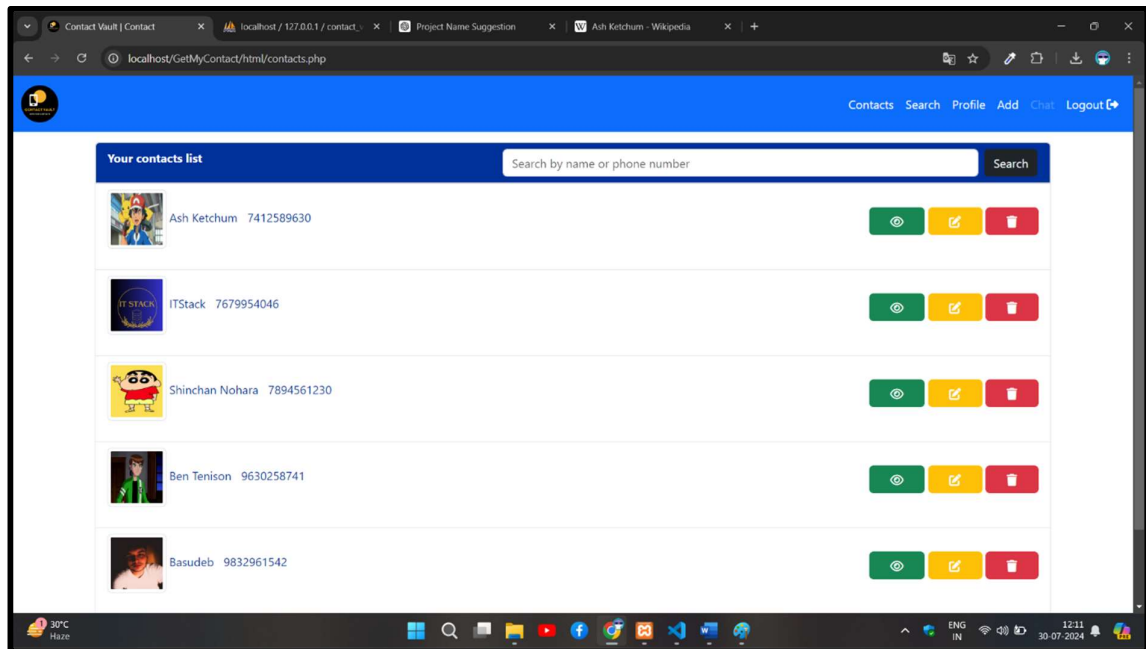
All contacts saved by logged in users.
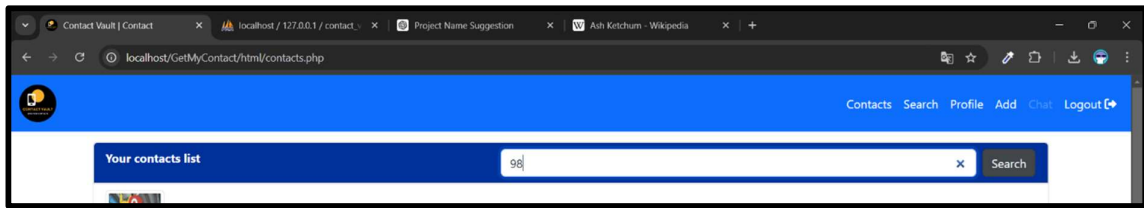
Searching all contacts phone number have 98
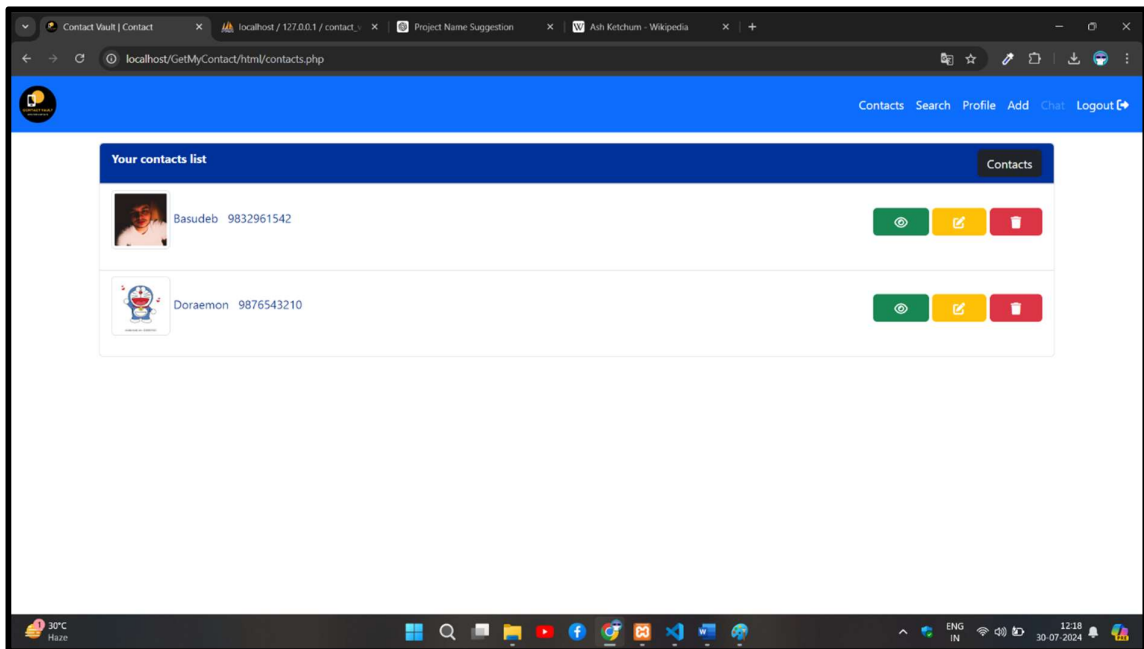
Output of searched number.
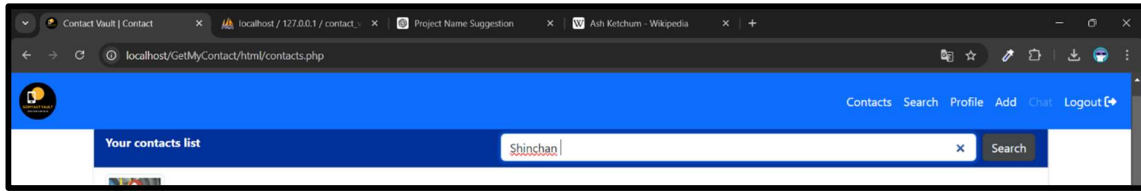
Searching all contacts with name Shinchan
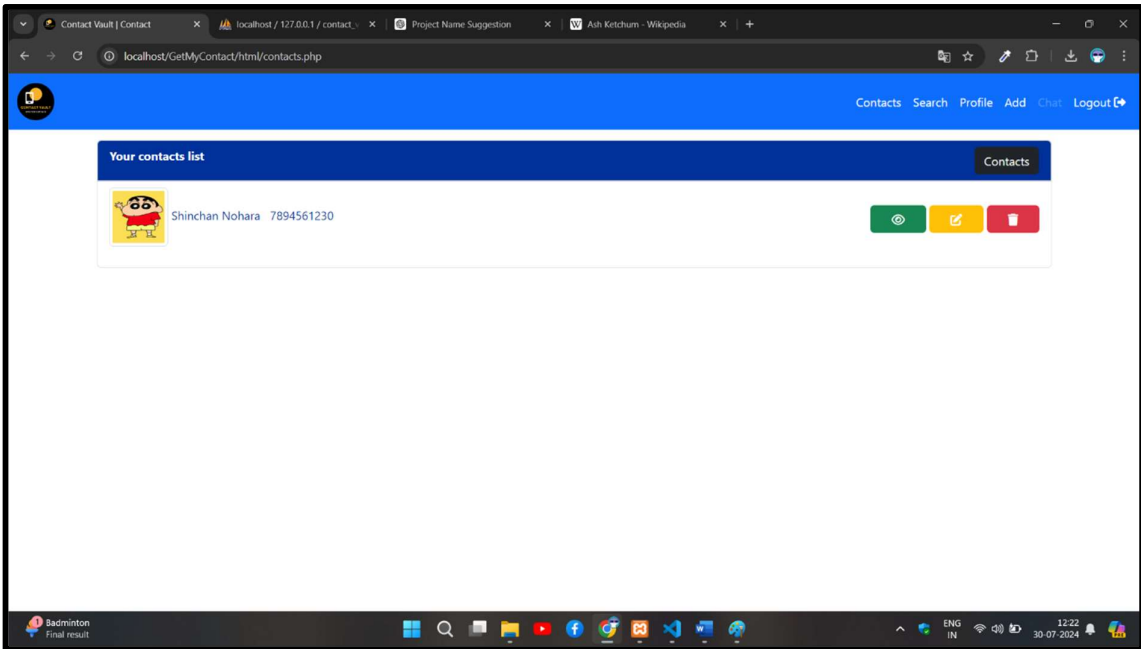
Output of searched name.
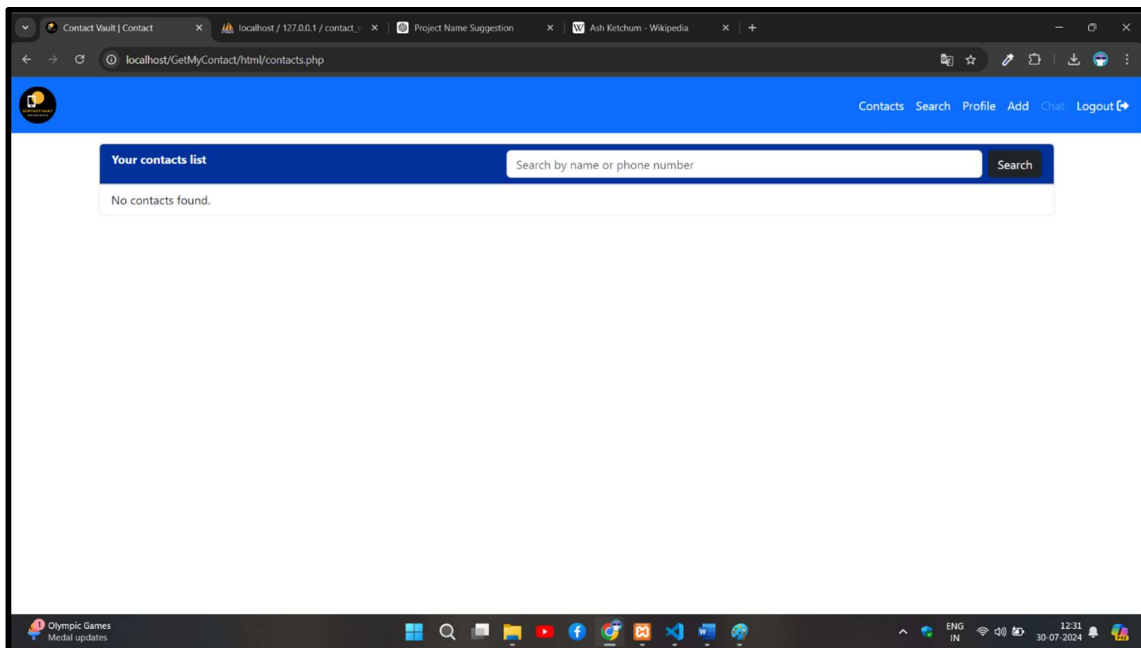
If contacts are not saved by user.
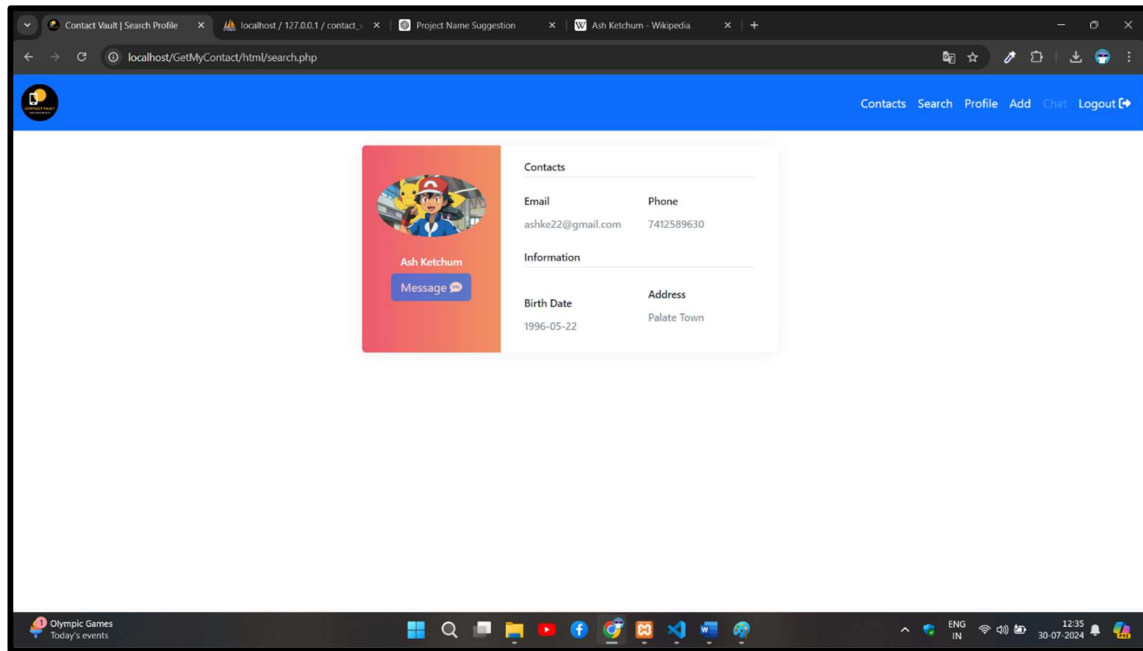
View a contact profile by clicking view button of contact page (Green Button).
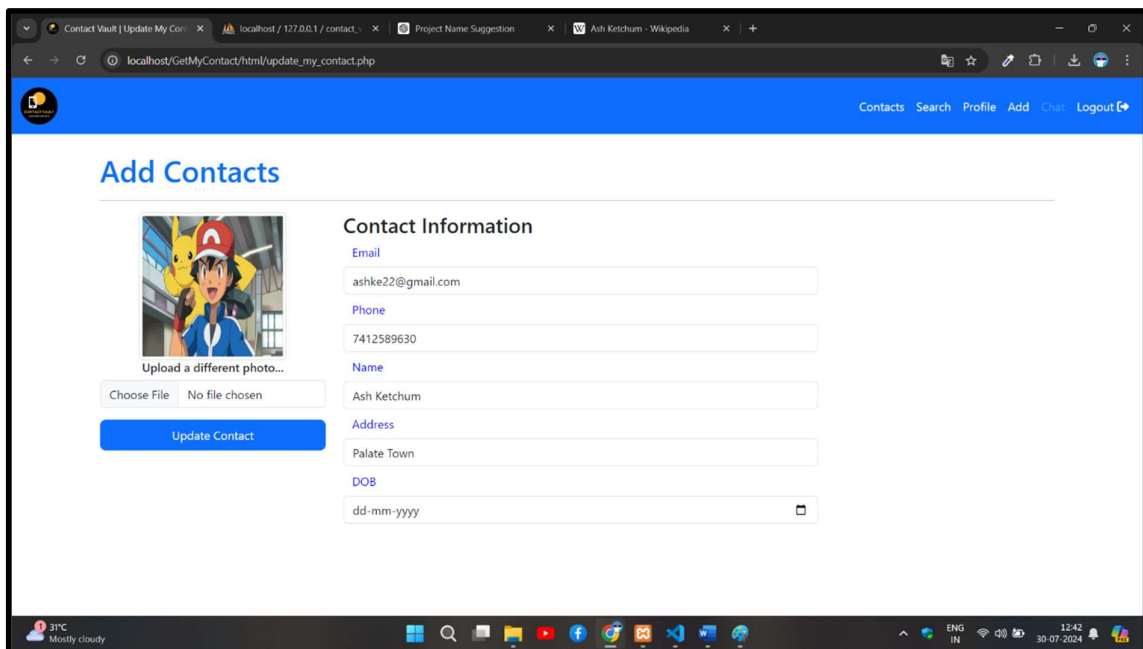
Update a saved contact details by clicking update button of contact page (Yellow Button).
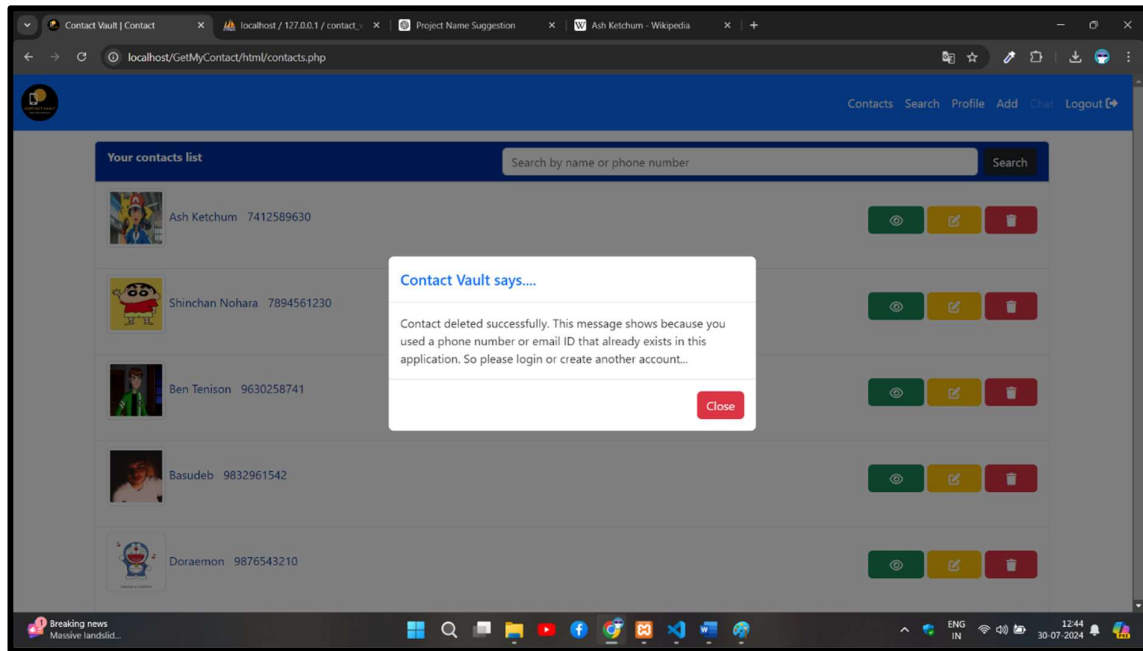
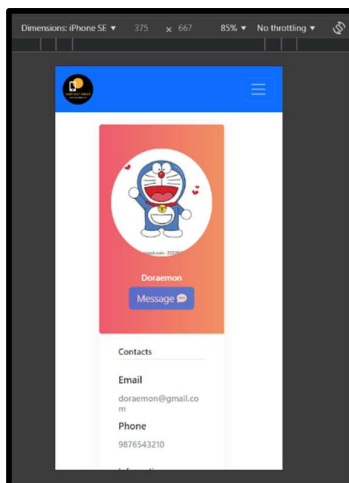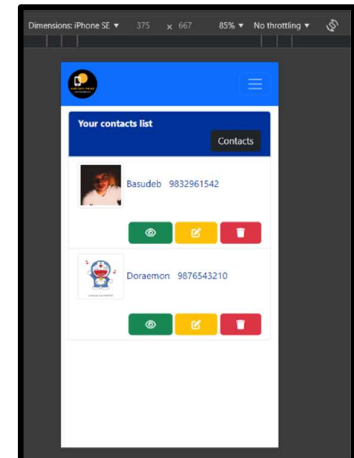**Delete a saved contact by clicking delete button of contact page (Red Button).**
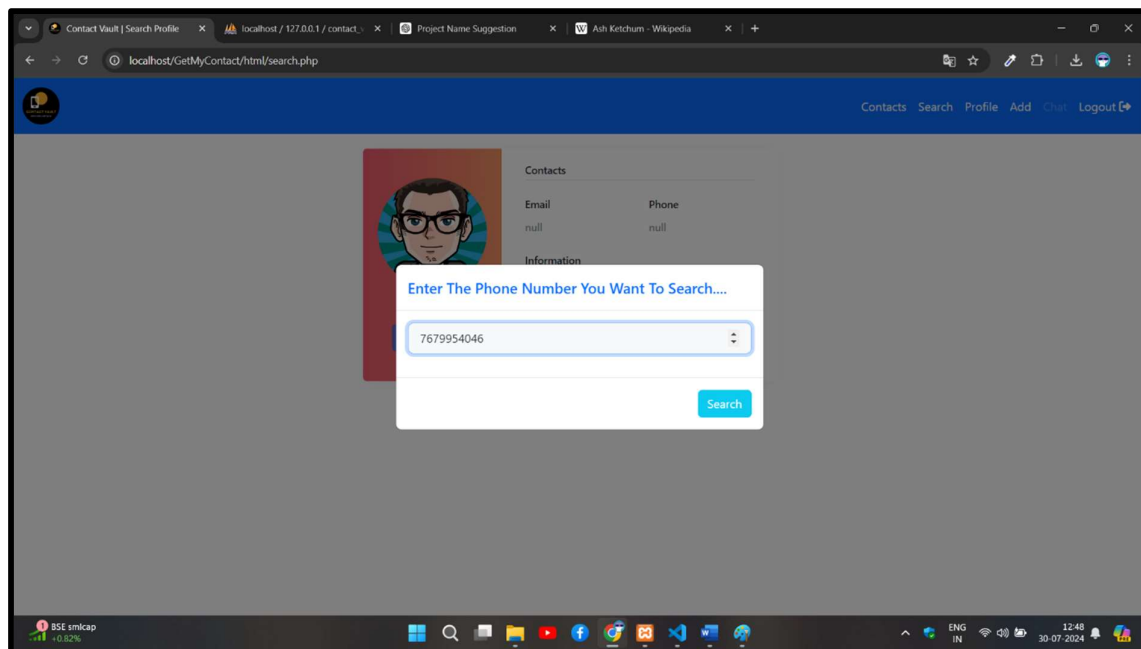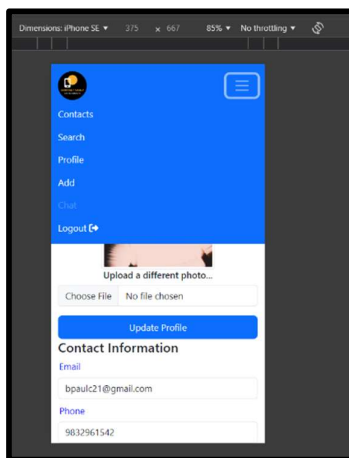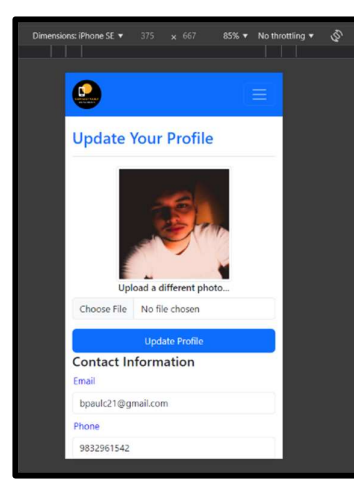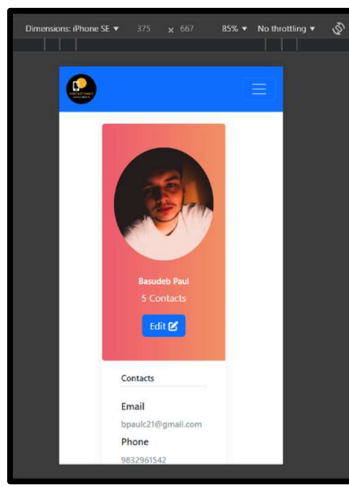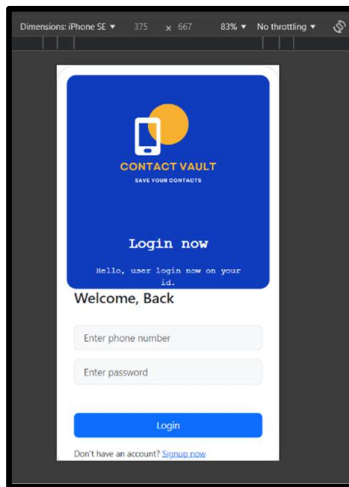

Search unknown phone number details.
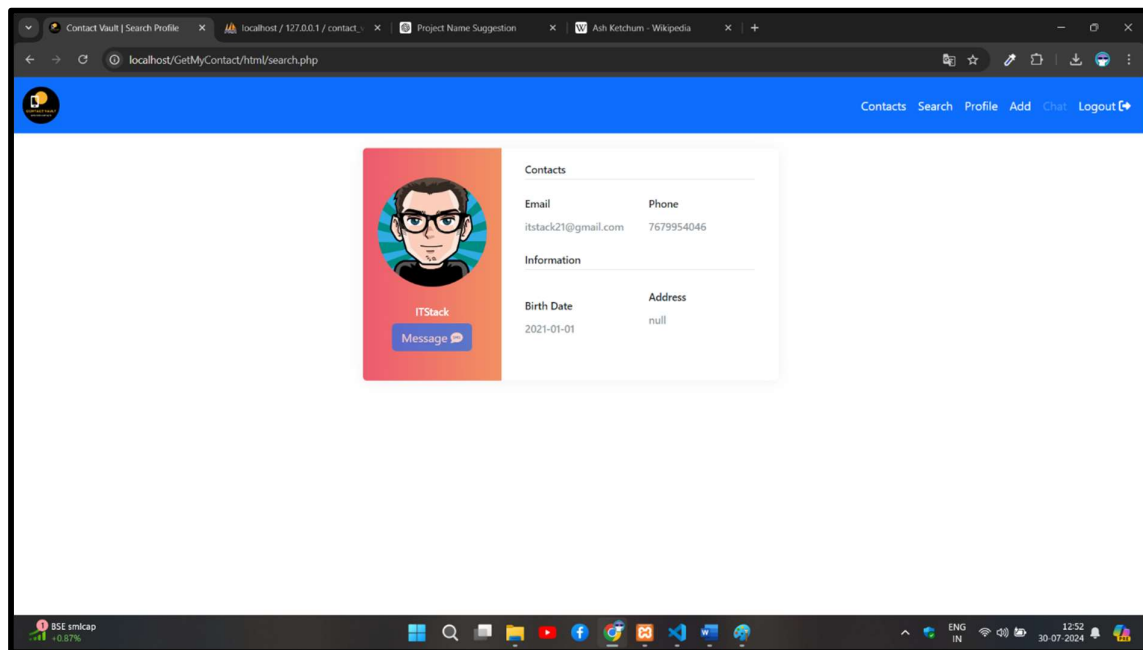
Unknown contact details the contact name is mostly saved by users.

SCREENSHOTS – Mobile Screen
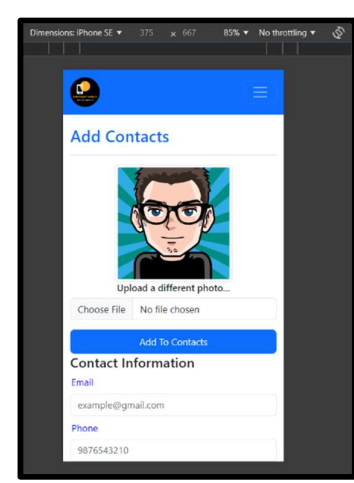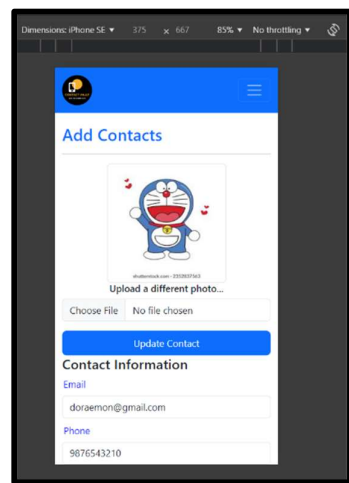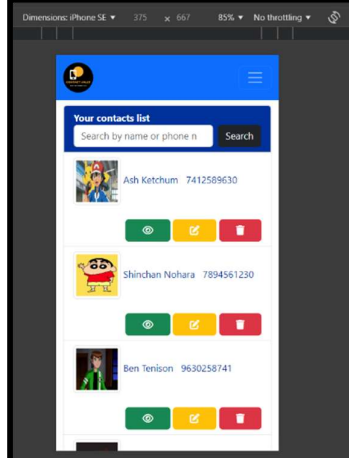
FUTURE SCOPE

i.   Additional Features:

- Implement chat functionality for users.

- Enhance global search capabilities to provide more detailed information.

- Implement additional search filters for better contact retrieval.

- Allow users to import contacts from social media platforms.
- Develop a notification system to alert users about updates and important events related to their contacts.
- Extend the web application to a mobile platform for better accessibility.

This project represents a significant step forward in the realm of web-based contact management solutions, offering a robust and user-friendly application that addresses common needs in today's digital world.

ii.   Performance Improvements:

- Optimize database queries to improve performance.

- Implement caching mechanisms to reduce server load.

CONCLUSION

The Contact Vault project is a comprehensive web-based application designed to streamline the management of contact information. This project, developed as part of the MCA curriculum, incorporates several advanced web technologies and robust features to meet the needs of users.

Key highlights of the project include:

- **User-Friendly Interface**: Utilizing HTML, CSS, Bootstrap 5, JavaScript, and jQuery, the application provides a responsive and intuitive user interface that works seamlessly across various devices.

- **Efficient Backend Management**: With a backend built on PHP and a MySQL database, the application ensures secure and efficient handling of user data and contact information.

- **Comprehensive Contact Management**: Users can add, edit, delete, and search for contacts, with the additional functionality of identifying unknown phone numbers through a global database search.

- **Secure User Authentication**: The application employs secure login and logout mechanisms along with session management to protect user data and maintain privacy.

- **Profile Management**: Users can upload and manage profile images, enhancing the personal touch and visual appeal of their contact lists.

The development and deployment of Contact Vault demonstrate a well-rounded understanding of web design and development principles, database management, and user-centric design. This project not only meets the academic requirements but also provides practical solutions to real-world problems related to contact management and unknown caller identification.