

RANCHI WOMEN'S COLLEGE, RANCHI

(An autonomous constituent, Affiliated to Ranchi University Ranchi)



A PROJECT REPORT ON “Women’s safety”

Submitted in Partial Fulfillment for the Degree of
B.Sc.Information Technology(IT)
(2021-2024)

Under the Supervision of :-

(Mrs.Purnima Kumari Srivastava)

Department of information technology

Ranchi Women’s College,Ranchi

Submitted by:-

Name- Yogita Bhardwaj

Vidya kumari

Roll No- 21VCBSIT007,

21VCBSIT008

ACKNOWLEDGEMENT

The successful completion of this project is solely due to the guidance and encouragement received from the people around us. I specially thank my project guide **Mrs. Purnima Kumari Srivastava** for extending her support and rendering constant help whenever needed, her helpful criticism and support inspired us to present the project in present form.

We also extended our gratitude to our institute **RANCHI WOMEN'S COLLEGE** for this encouragement and crowned our efforts with success. This report would have been impossible but for the support and guidance that we received from various people at different stages of the project.

We express our deep gratitude to all.

THANKYOU

Yogita Bhardwaj

Vidya kumari

RANCHI WOMEN'S COLLEGE,RANCHI



CERTIFICATE

This is to certify that the project report entitled **Women's Safety** Submitted to **Ranchi Women's College, Ranchi** in partial fulfillment of the requirement for the award of degree of **B.Sc.Information Technology(IT)**, is an original work carried out by **Yogita bhardwaj & Vidya kumari** Roll no. **21VCBSIT007 , 21VCBSIT008** under my direct supervision and guidance.

Signature of Guide
(Mrs.Purnima Kumari Srivastava)

External Examiner

RANCHI WOMEN'S COLLEGE,RANCHI



CERTIFICATE

This is to certify that the project report entitled “**Women’s safety**” Submitted to **Ranchi Women’s College, Ranchi** in partial fulfillment of the requirement for the award of degree of **B.Sc.Information Technology**, is an original work carried out by **Yogita bhardwaj & Vidya kumari** Roll no. **21VCBSIT007 , 21VCBSIT008** under my direct supervision and guidance.

Signature of Guide
(Mrs. Purnima Kumari Srivastava)

External Examiner

TABLE OF CONTENT

1. Introduction
2. Platform (H/w & S/W)
3. Entity Relationship Diagram (ERD)
4. Data flow Diagram (DRD)
5. Modules Description
6. Screen Shorts
7. Coding
8. Database Schema
9. Output
10. Future Scope
11. Conclusion
12. Bibliography

1. INTRODUCTION

our women's safety app, a comprehensive solution crafted to empower and protect women in their daily lives. In today's world, ensuring personal safety is paramount, and our app is designed with this in mind.

The app offers a range of features to address various safety concerns. First and foremost, it provides a real-time location sharing feature, allowing users to share their whereabouts with trusted contacts. In case of an emergency, users can quickly send out distress signals, triggering alerts to designated contacts and authorities.

Moreover, the app incorporates a discreet panic button feature, enabling users to seek help swiftly without drawing attention. This feature is especially crucial in situations where immediate action is required but overt communication may be challenging or risky.

Additionally, the app includes educational resources and tips on self-defense, empowering women with knowledge and skills to enhance their personal safety awareness.

Furthermore, the app fosters a supportive community by facilitating peer-to-peer support and information sharing among users. This community aspect enhances the overall safety net provided by the app, creating a network of solidarity and assistance.

In essence, our women's safety app is a proactive tool designed to provide peace of mind and support to women, empowering them to navigate the world with confidence and security.

For Python, I have used Python version 3.7 and MySQL version 5.6.

This is the Server –Side technologies and database. For frontend development I have used HTML, CSS, JAVASCRIPT. Some of jQuery library are also used to developed this project.

Rideon allows an user to manage the inventory with full create, retrieve, update and delete (CRUD) Functionality with regards to ride in the system. User has ability to add/delete their profile.

1.1 TOOLS AND TECHNOLOGIES USED

HTML

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications.

The major points of HTML are given below:

- HTML stands for HyperText Markup Language.
- HTML is used to create web pages and web applications.
- HTML describes the structure of a Web Page.
- HTML consists of series of elements.
- HTML elements tell the browser how to display the Content.
- HTML elements label pieces of content such as “This is a Heading”, “This is a Paragraph”, “This is a Link” etc.
- HTML is widely used language on the web.
- We can create a static website by HTML only.
- Technically, HTML is a Markup language rather than a programming language.

CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.
- You can add new looks to your old HTML documents.
- You can completely change the look of your website with only a few changes in CSS.

JAVASCRIPT

JavaScript is the world's most popular programming language.

JavaScript is the programming language of the Web.

JavaScript is easy to learn.

JavaScript is *an object-based scripting language* which is lightweight and cross-platform.

JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

- JavaScript is Lightweight, interpreted programming language.
- JavaScript is designed for creating network-centric applications.
- JavaScript is complementary to and integrated with Java.
- JavaScript is complementary to and integrated with HTML
- JavaScript is an open and cross-platform



Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

BOOTSTRAP

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs



PYTHON FLASK

Flask is a micro [webframework](#) written in [Python](#). It is classified as a [microframework](#) because it does not require particular tools or libraries. It has no [database](#) abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Applications that use the Flask framework include [Pinterest](#) and [LinkedIn](#).

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO).



MySQL is a very popular open-source relational database management system (RDBMS).

What is MySQL?

- MySQL is a relational database management system
- MySQL is open-source
- MySQL is free
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is cross-platform
- MySQL is compliant with the ANSI SQL standard
- MySQL was first released in 1995
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My

Who Uses MySQL?

- Huge websites like Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube, etc.
- Content Management Systems like WordPress, Drupal, Joomla!, Contao, etc.
- A very large number of web developers around the world

1.2 APPLICATION FUNCTIONS

- Provide a registration page for user and admin.
- Verify user and admin using login module.
- Provide a search mechanism.
- Provide complete details of the user.
- Enable user to add a contact.
- Enable user to add, update and delete contacts.
- Enable user to view and contact details and delete them if required.
- Provide a way to add a contact.
- Display total mail send to user contact Display admin review .
- Provide mechanism to logout.

1.3 PURPOSE

- **Emergency Assistance:** One of the primary purposes of a women's safety app is to provide immediate assistance in times of distress or danger. This includes features like panic buttons, which enable users to quickly send alerts to their emergency contacts or authorities, allowing for swift response and aid in critical situations.
- **Real-Time Location Sharing:** Another important purpose is to offer real-time location sharing capabilities. This feature allows users to share their location with trusted contacts, ensuring that friends, family members, or authorities can monitor their whereabouts and provide assistance if needed, especially when traveling alone or in unfamiliar environments.
- **Education and Awareness:** Women's safety apps often serve as educational platforms, offering resources, tips, and information on personal safety, self-defense techniques, and risk mitigation strategies. By raising awareness and providing valuable knowledge, these apps empower women to make informed decisions and take proactive measures to enhance their safety and well-being in various situations.

2. SYSTEM SPECIFICATIONS(Platform)

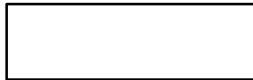
2.1 Hardware:

- Processor: Intel® Core™ i5-3000 11th gen
- Installed memory (RAM): 8 GB
- System Type: 64-bit Operating System
- SSD Storage : 512GB
- Browser: Chrome

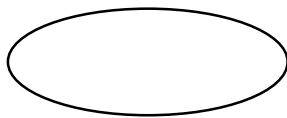
3. ENTITY RELATIONSHIP DIAGRAM (ERD)

ER-modeling is a data modeling method used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modeling method are called Entity-Relationship Diagrams or ER diagrams or ERDs. Entity Relationship (ER) diagram is used to organize data as a relation, normalizing relations and finally obtaining a relational database model.

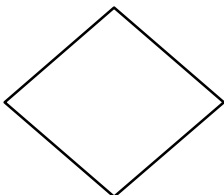
Components of ER-Diagram



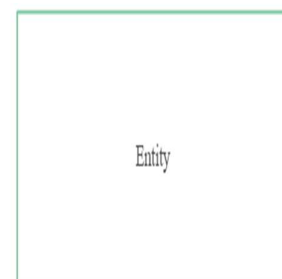
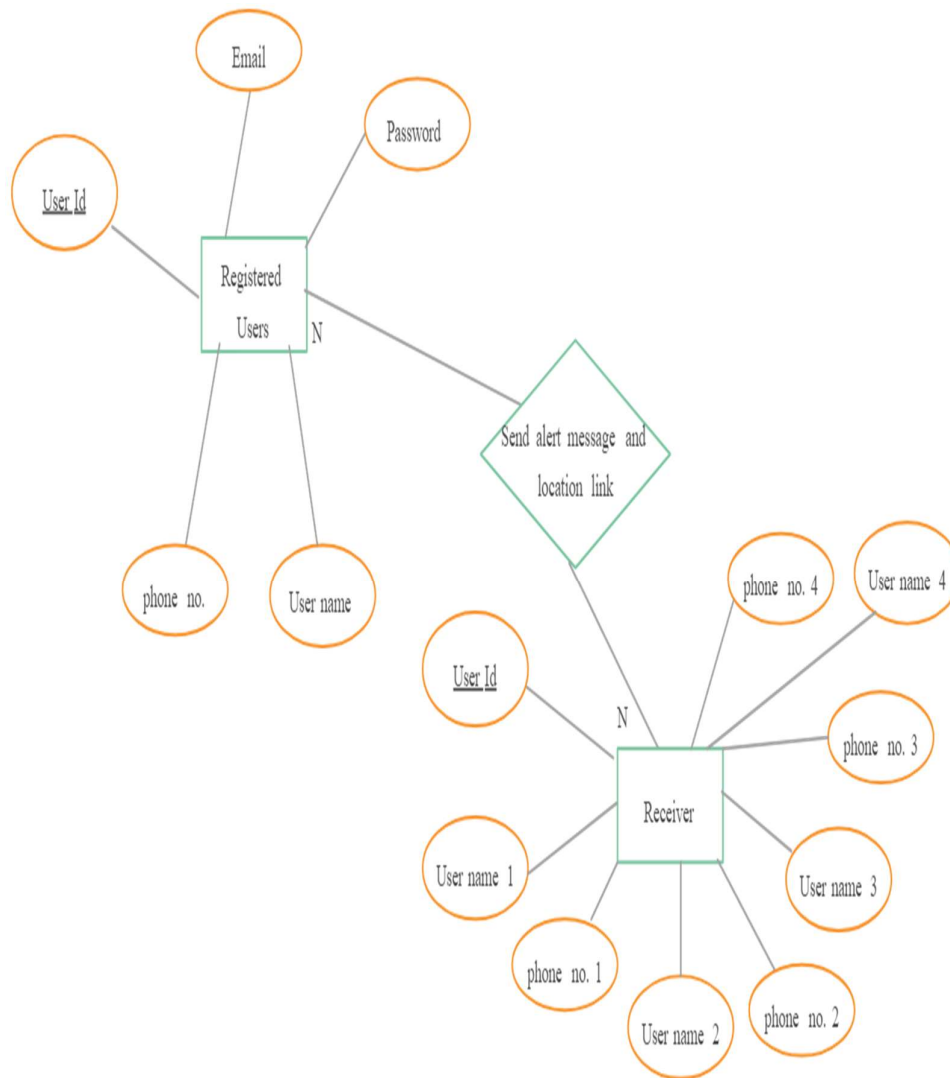
Entity: An entity can be a real-world object, either animate or inanimate, that can be merely identifiable. An entity is denoted as a rectangle in an ER diagram. For example, in a school database, students, teachers, classes, and courses offered can be treated as entities. All these entities have some attributes or properties that give them their identity.

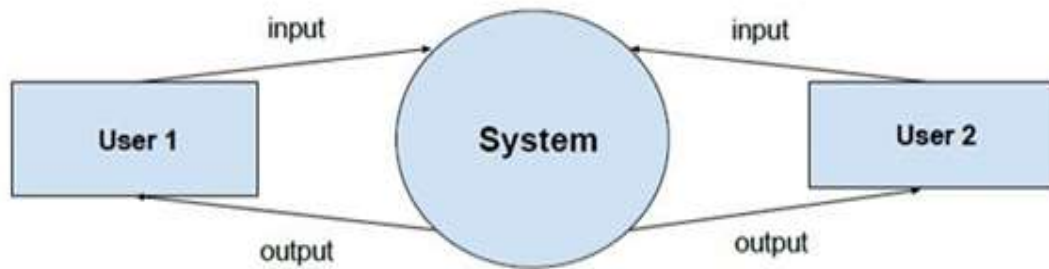


Attributes: Entities are denoted utilizing their properties, known as attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.



Relationships: The association among entities is known as relationship. Relationships are represented by the diamond-shaped box.





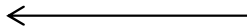
V. ARCHITECTURE



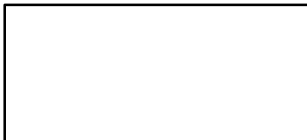
4. DATA FLOW DIAGRAM (DFD)

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

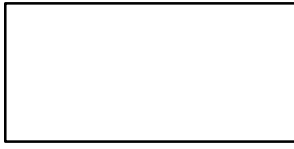
Symbols and Notations Used in DFDs:



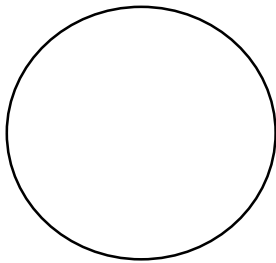
Dataflow: This arrow denotes a data object or the data item that is being sent either from one process to another or from an external entity to a process etc. Data moves in a Specific Direction from an origin to a Destination.



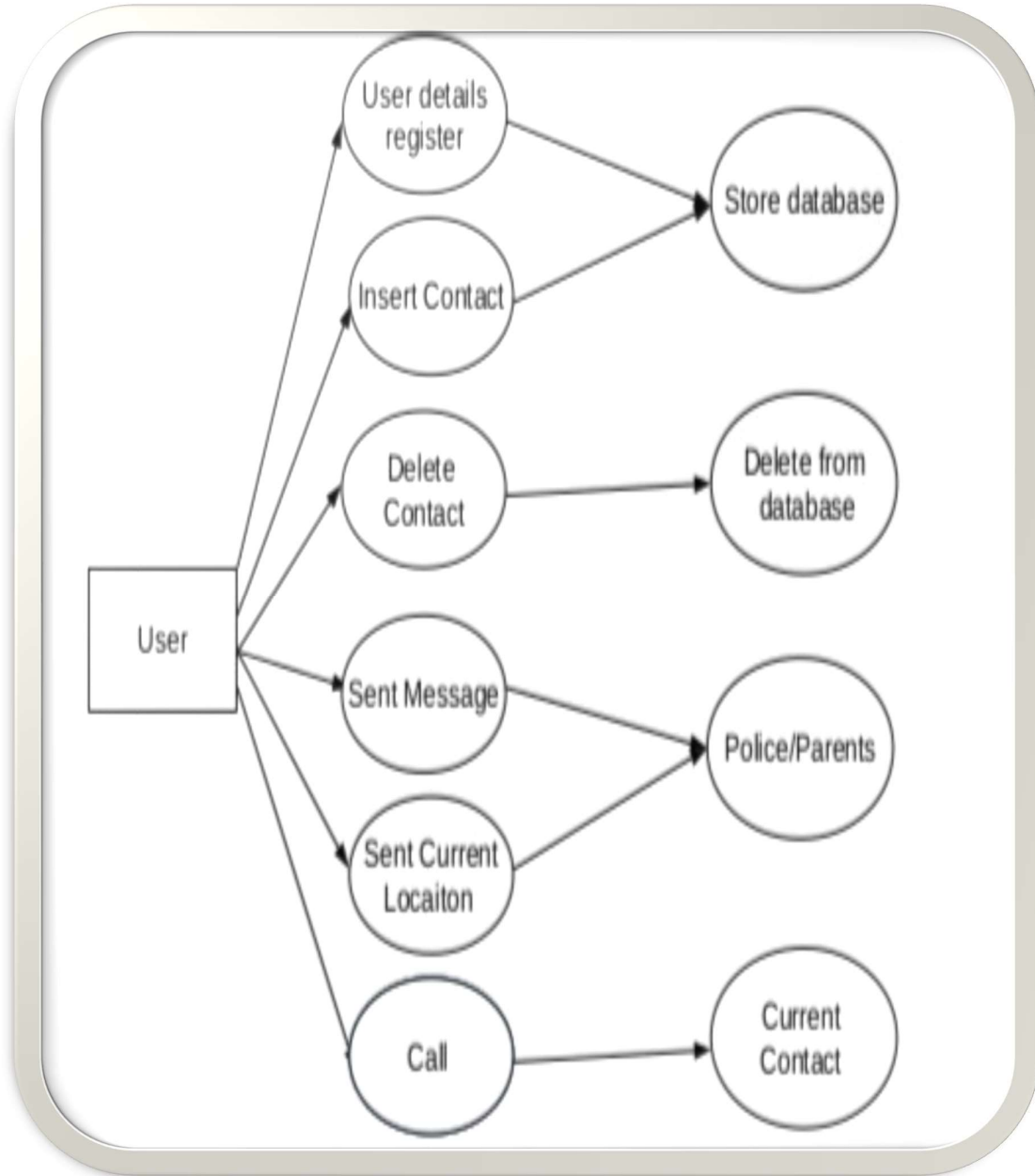
Source: External Source or destination of data, which may be People, Programs, Organization or other entities. This denotes an entity that is outside the System. It is not a Part of the System and Can be a machine , another Software System, an rganization or a person.



DataStore: Here Data are Stored or referenced by a process in the System. This represents a store or repository of data that is used by the Software.



Process: People, Procedures, or devices that uses or produce(Transformed)Data.ThePhysical Component is not identified. This is a process or transform that receives the information received from the external entity,Processes it or changes it in some way and then sends it to an external entity.



5. MODULES DESCRIPTIONS

The Women's Safety module in the Aap app aims to empower women by providing them with tools and information to enhance their safety. Here are the descriptions of the key features:

- **Emergency Button:** This feature allows users to quickly send distress signals to their emergency contacts or local authorities in case of an emergency. It can be activated with a single tap, triggering an alert with the user's location.
- **Location Sharing:** Users can share their real-time location with trusted contacts, allowing them to keep track of their whereabouts and provide assistance if needed.
- **Safety Tips:** The app offers practical advice and safety tips on various topics such as traveling alone, using public transportation, and staying safe at night. These tips are designed to help women make informed decisions and reduce the risk of encountering dangerous situations.
- **Helpline Numbers:** The app provides a comprehensive list of emergency helpline numbers for police, ambulance, fire services, and other relevant authorities. Users can quickly access these numbers in case of emergencies.
- **Safe Routes:** Users can access information about safe routes in their area, including well-lit streets, busy areas, and places with surveillance cameras. This feature helps women plan their travel routes more safely, especially when walking alone or at night.
- **Legal Rights Information:** The app provides information about women's legal rights,

Steps:

- Step 1: Start
- Step 2: user can registered with their Details
- Step 3: user login with email id and password
- Step 4: Create,Update,Delete profile
- Step 5: Viewing profile details
- Step 6: Change password
- Step 7: Viewing the user details
- Step 8: User can add their relatons contact number
- Step 9: User viewing the all contact number
- Step 10: User use Emergency button
- Step 11: Shair emergency email with current location
- Step 12: Stop

SCREEN SHORT& CODING

- INDEX.HTML



```

{% extends "main_layout.html" %}

{% block content %}
<!-- flash message show using javascript -->
{% with messages = get_flashed_messages() %}
{% if messages %}
{% for message in messages %}
<script>
    alert("{{message}}")
</script>
{% endfor %}
{% endif %}
{% endwith %}

<!-- Hero Section -->
<section id="hero" class="hero section">
    <div class="hero-bg">
        
    </div>
    <div class="container text-center">
        <div class="d-flex flex-column justify-content-center align-items-center">
            <h1 data-aos="fade-up" class="">Welcome to <span>Women's Safety </span></h1>
            <p data-aos="fade-up" data-aos-delay="100" class="">Quickly start your project now
and set the stage for success<br></p>
            <div class="d-flex" data-aos="fade-up" data-aos-delay="200">
                <a href="/user_signup" class="btn-get-started">Get Started</a>
                <a href="https://www.youtube.com/watch?v=LXb3EKWsInQ" class="lightbox btn-
watch-video d-flex align-items-center"><i class="bi bi-play-circle"></i><span>Watch
Video</span></a>
            </div>
            
        </div>
    </div>

```

</div>

</section><!-- /Hero Section -->

<!-- About Section -->

<section id="about" class="about section">

<div class="container">

<div class="row gy-4">

<div class="col-lg-6 content" data-aos="fade-up" data-aos-delay="100">

<p class="who-we-are">Who We Are</p>

<h3>Unleashing Potential with Creative Strategy</h3>

<p class="fst-italic">

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore

 magna aliqua.

</p>

 <i class="bi bi-check-circle"></i> Ullamco laboris nisi ut aliquip ex ea commodo consequat.

 <i class="bi bi-check-circle"></i> Duis aute irure dolor in reprehenderit in voluptate velit.

 <i class="bi bi-check-circle"></i> Ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate trideta storacalaperda mastiro dolore eu fugiat nulla pariatur.

 Read More<i class="bi bi-arrow-right"></i>

</div>

<div class="col-lg-6 about-images" data-aos="fade-up" data-aos-delay="200">

```

<div class="row gy-4">
  <div class="col-lg-6">
    
  </div>
  <div class="col-lg-6">
    <div class="row gy-4">
      <div class="col-lg-12">
        
      </div>
      <div class="col-lg-12">
        
      </div>
    </div>
  </div>
</div>

</div>

</div>

</div>
</section><!-- /About Section -->

<!-- Clients Section -->
<section id="clients" class="clients section">

  <div class="container" data-aos="fade-up">

    <div class="row gy-4">

      <div class="col-xl-2 col-md-3 col-6 client-logo">
        
      </div><!-- End Client Item -->

```

```

<div class="col-xl-2 col-md-3 col-6 client-logo">
  
</div><!-- End Client Item -->

<div class="col-xl-2 col-md-3 col-6 client-logo">
  
</div><!-- End Client Item -->

<div class="col-xl-2 col-md-3 col-6 client-logo">
  
</div><!-- End Client Item -->

<div class="col-xl-2 col-md-3 col-6 client-logo">
  
</div><!-- End Client Item -->

<div class="col-xl-2 col-md-3 col-6 client-logo">
  
</div><!-- End Client Item -->

</div>

</div>

</section><!-- /Clients Section -->

<!-- Features Section -->
<section id="features" class="features section">

  <!-- Section Title -->
  <div class="container section-title" data-aos="fade-up">
    <h2 class="">Features</h2>
    <p>Necessitatibus eius consequatur ex aliquid fuga eum quidem sint consectetur
    velit</p>

```

```
</div><!-- End Section Title -->
```

```
</section>
```

```
{% endblock %}
```

2. USER SIGNUP

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/user_signup". The page title is "Index - QuickStart Bootstrap Te". The website header features the logo "Q Women's Security" and a navigation menu with links: Home, About, Features, Services, Pricing, Dropdown, and Get Started.

The main content area is titled "User Signup" and contains the following form fields:

- Name:
- Email:
- Mobile:
- Password:

Below the form fields is a red "Signup Now" button. A link "Already registered? login Now" is positioned below the button.

The footer is divided into four columns:

- QuickStart**
A108 Adam Street
New York, NY 535022
Phone: +1 5589 55488 55
Email: info@example.com
- Useful Links**
[Home](#)
[About us](#)
[Services](#)
[Terms of service](#)
[Privacy policy](#)
- Our Services**
[Web Design](#)
[Web Development](#)
[Product Management](#)
[Marketing](#)
[Graphic Design](#)
- Our Newsletter**
Subscribe to our newsletter and receive the latest news about our products and services!

[Subscribe](#)

The Windows taskbar at the bottom shows the search bar, task view button, and various application icons. The system tray on the right indicates the language is "ENG IN", the time is "21:19", and the date is "13-05-2024".

```

{% extends "main_layout.html" %}

{% block content %}
    <style>

        button{
            background-color: #d87878;
            color: white;
            cursor: pointer;
            margin: 12px 0;
            width: 20%;
            border-radius: 8px ;
            padding: 7px 15px;
        }
    </style>
    <form action="/user_signup" method="post"><br><br><br><br>
        <center>

            <table>
                <tr>
                    <th colspan="2">User Signup</th>
                </tr>
                <tr>
                    <td>Name:</td>
                    <td><input name="name" type="text" placeholder="Enter your Name"
autofocus required></td>
                </tr>
                <tr>
                    <td>Email:</td>
                    <td><input name="email" type="email"> </td>
                </tr>
                <tr>
                    <td>Mobile:</td>
                    <td><input name="mobile" type="number"></td>
                </tr>
                <tr>
                    <td>Password:</td>
                    <td><input name="password" type="password" placeholder="create
password"></td>
                </tr>

            </table>
            <button type="submit">Signup Now</button>

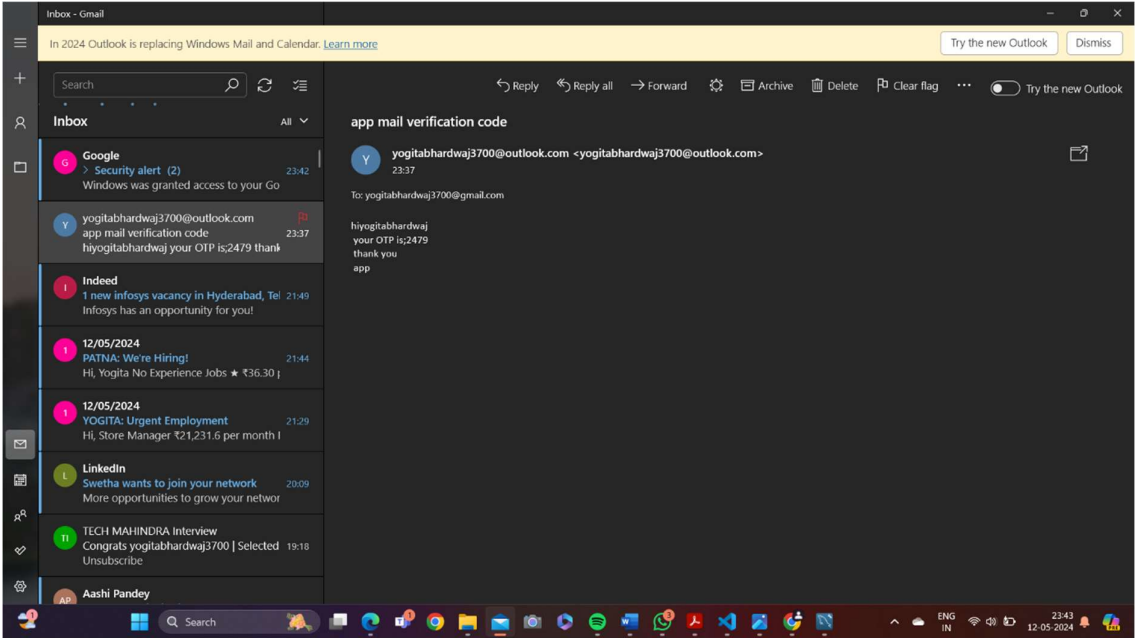
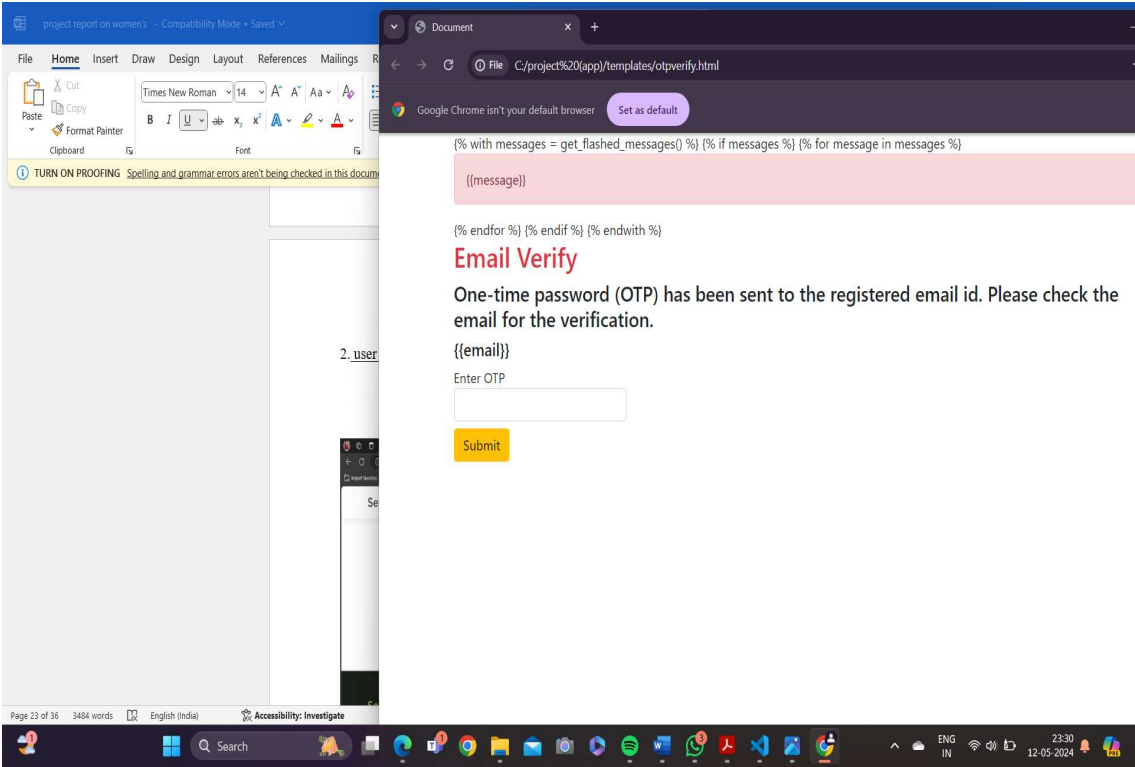
        </form action="">
        <div class="fotter">
            <span>Already registered?<a href="/user_login">login Now</a></span>
        </div>

```



```
</center>  
</table>  
{% endblock %}
```

3. EMAIL VERIFY



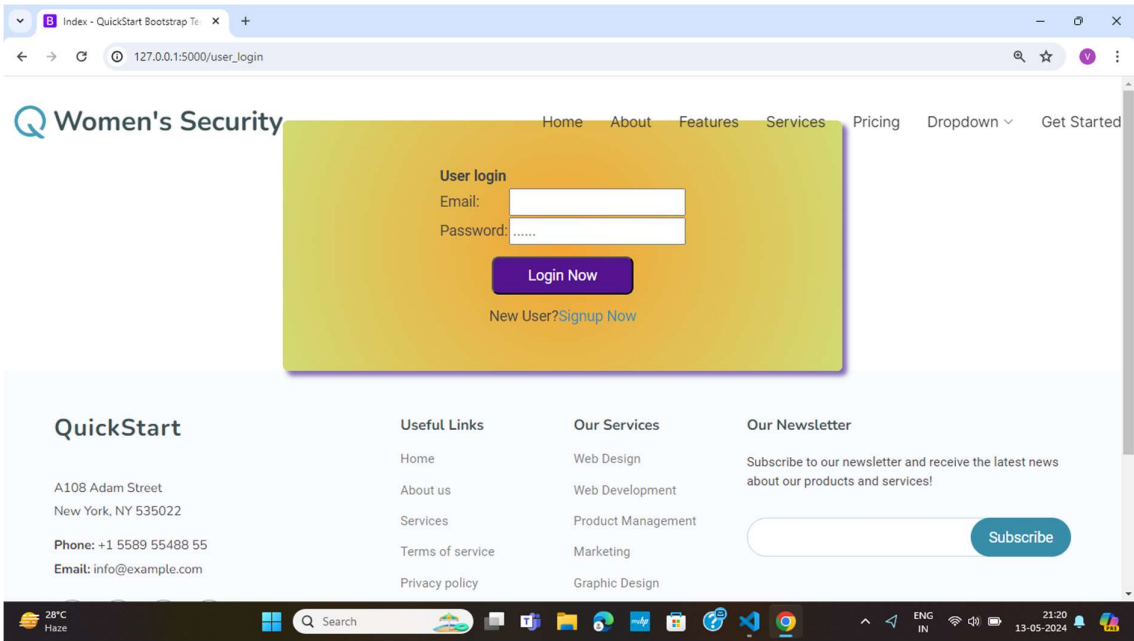
```

<!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>
  <!-- CSS only -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
</head>
<body>
  <div class="container">
    {% with messages = get_flashed_messages() %}
    {% if messages %}
    {% for message in messages %}
      <div class="alert alert-danger" role="alert">
        {{ message }}
      </div>
    {% endfor %}
    {% endif %}
    {% endwith %}
    <h2 class="text-danger">Email Verify</h2>
    <form action="/otpverify" method="POST">
    <h4> One-time password (OTP) has been sent to the registered email id. Please check the
    email for the verification.</h4>
    <h5>{{ email }}</h5>
    <div class="form-group">
      <label for="exampleInputEmail1">Enter OTP</label>
      <input type="text" name="otp" class="form-control w-25" >

```

```
</div>
<input type="hidden" name="email" value="{{email}}">
<button type="submit" class="btn btn-warning mt-2">Submit</button>
</form>
</div>
</body>
</html>
```

4. USER LOGIN



```

{% extends "main_layout.html" %}

{% block content %}

<style>
body{
    background-image: url(../static/image/login.jpeg);
    background-size: cover;
}
.mybg{
    background: radial-gradient(circle, rgb(244, 164, 52) 0%, rgb(209, 219, 115) 100%);
    box-shadow: 5px 5px 5px rgb(134, 110, 181);
}
button{
    background-color: #53148d;
    color: white;
    cursor: pointer;
    margin: 12px 0;
    width: 30%;
    border-radius: 8px ;
    padding: 7px 15px;
}
</style>

<div class="container mybg mt-5 p-5 rounded w-50">
    {% with messages = get_flashed_messages() %}
    {% if messages %}
    {% for message in messages %}

    <div class="alert alert-danger" role="alert">
        {{ message }}
    </div><br><br><br>
    {% endfor %}
    {% endif %}
    {% endwith %}
<form action="/user_login" method="post">

```

```

<center>
<table>
  <tr><th colspan="2">User login</th></tr>
  <tr>
    <td>Email:</td>
    <td><input name="email" type="email"></td>
  </tr>
  <tr>
    <td>Password:</td>
    <td><input name="password" type="password" placeholder="....."></td>
  </tr>

</table>

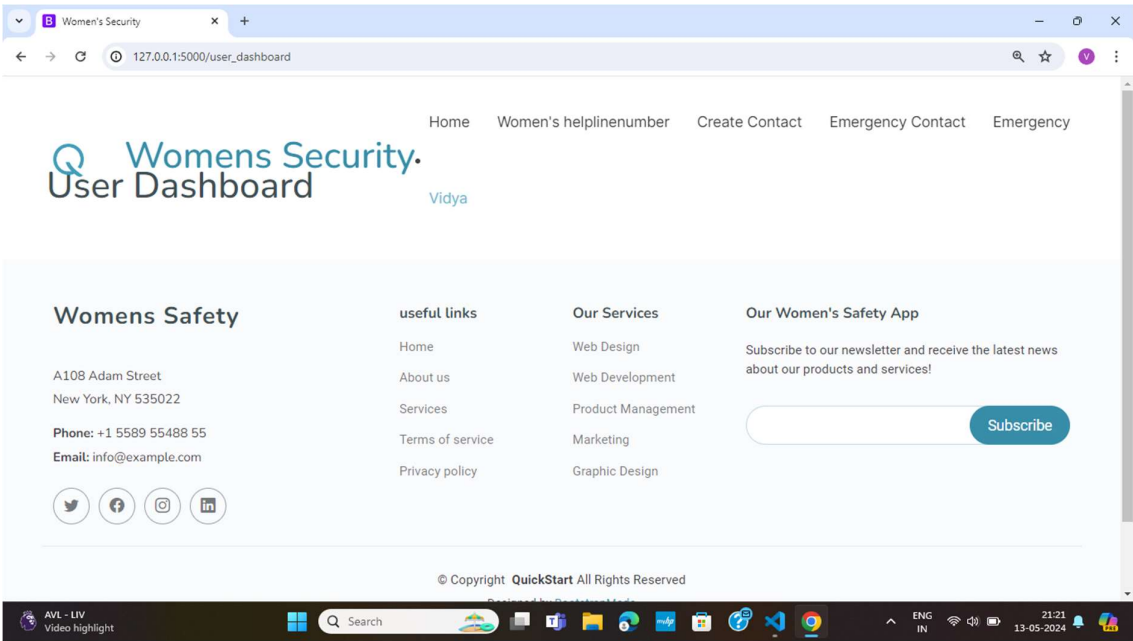
<button class="btn btn-warning mt-3" type="submit">Login Now</button>

</form action="">
<div class="fotter">
  <span> New User?<a href="/user_signup">Signup Now</a></span>
</div>

</center>
</table>
</div>
{% endblock %}

```

5. USER DASHBOARD




```
{% extends "user_layout.html" %}
{% block content %}
    <h1 class="mt-5 p-5">User Dashboard</h1>
    
{% endblock %}
```

6. HOME



```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>Women's Security</title>

  <base href="../static/">

  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="assets/img/favicon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Fonts -->
  <link href="https://fonts.googleapis.com" rel="preconnect">
  <link href="https://fonts.gstatic.com" rel="preconnect" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;
0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&family=Inter:wght@100;200;300;400;500
;600;700;800;900&family=Nunito:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,9
00;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="assets/vendor/aos/aos.css" rel="stylesheet">
  <link href="assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

```

```

<!-- Main CSS File -->
<link href="assets/css/main.css" rel="stylesheet">

<!-- =====
* Template Name: QuickStart
* Template URL: https://bootstrapmade.com/quickstart-bootstrap-startup-website-template/
* Updated: May 10 2024 with Bootstrap v5.3.3
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/
===== -->

</head>

<body class="index-page">

<header id="header" class="header d-flex align-items-center fixed-top">
  <div class="container-fluid container-xl position-relative d-flex align-items-center">

    <a href="index.html" class="logo d-flex align-items-center me-auto">
      

    <h1 class="sitename"><a href="">Women's Security</a></h1>
  </a>

  <nav id="navmenu" class="navmenu">
    <ul>
      <li><a href="/user_dashboard" class="">Home</a></li>
      <li><a href="/user_helplinenumber">Women's helplinenumber</a></li>
      <li><a href="/user_create_contact">Create Contact</a></li>
      <li><a href="/user_create_contact_show">Emergency Contact</a></li>
    </ul>
  </nav>

```

```

    <li><a href="/user_emergency">Emergency</a></li>

    </ul>

    </li>

    <li><a class="getstarted" href="/user_profile">{{session['user_name']}}</a></li>
</ul>

<i class="mobile-nav-toggle d-xl-none bi bi-list"></i>

</nav>

</div>
</header>

{% block content %}
{% endblock %}

<footer id="footer" class="footer position-relative">

<div class="container footer-top">
<div class="row gy-4">
<div class="col-lg-4 col-md-6 footer-about">
<a href="index.html" class="logo d-flex align-items-center">
    <span class="sitename">Womens Safety</span>
</a>
<div class="footer-contact pt-3">
<p>A108 Adam Street</p>
<p>New York, NY 535022</p>
<p class="mt-3"><strong>Phone:</strong> <span>+1 5589 55488 55</span></p>
<p><strong>Email:</strong> <span>info@example.com</span></p>
</div>
<div class="social-links d-flex mt-4">
<a href=""><i class="bi bi-twitter"></i></a>
<a href=""><i class="bi bi-facebook"></i></a>
<a href=""><i class="bi bi-instagram"></i></a>

```

```
<a href=""><i class="bi bi-linkedin"></i></a>
</div>
</div>
```

```
<div class="col-lg-2 col-md-3 footer-links">
  <h4>useful links</h4>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About us</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Terms of service</a></li>
    <li><a href="#">Privacy policy</a></li>
  </ul>
</div>
```

```
<div class="col-lg-2 col-md-3 footer-links">
  <h4>Our Services</h4>
  <ul>
    <li><a href="#">Web Design</a></li>
    <li><a href="#">Web Development</a></li>
    <li><a href="#">Product Management</a></li>
    <li><a href="#">Marketing</a></li>
    <li><a href="#">Graphic Design</a></li>
  </ul>
</div>
```

```
<div class="col-lg-4 col-md-12 footer-newsletter">
  <h4>Our Women's Safety App</h4>
  <p>Subscribe to our newsletter and receive the latest news about our products and
services!</p>
  <form action="forms/newsletter.php" method="post" class="php-email-form">
    <div class="newsletter-form"><input type="email" name="email"><input
type="submit" value="Subscribe"></div>
    <div class="loading">Loading</div>
```

```

        <div class="error-message"></div>
        <div class="sent-message">Your subscription request has been sent. Thank
you!</div>
    </form>
</div>

</div>
</div>

<div class="container copyright text-center mt-4">
    <p>©    <span>Copyright</span>    <strong    class="px-1
sitename">QuickStart</strong><span>All Rights Reserved</span></p>
    <div class="credits">
        <!-- All the links in the footer should remain intact. -->
        <!-- You can delete the links only if you've purchased the pro version. -->
        <!-- Licensing information: https://bootstrapmade.com/license/ -->
        <!-- Purchase the pro version with working PHP/AJAX contact form: [buy-url] -->
        Designed by <a href="https://bootstrapmade.com/">BootstrapMade</a>
    </div>
</div>

</footer>

<!-- Scroll Top -->
<a href="#" id="scroll-top" class="scroll-top d-flex align-items-center justify-content-
center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Preloader -->
<div id="preloader"></div>

<!-- Vendor JS Files -->
<script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="assets/vendor/php-email-form/validate.js"></script>
<script src="assets/vendor/aos/aos.js"></script>

```

```
<script src="assets/vendor/glightbox/js/glightbox.min.js"></script>
```

```
<script src="assets/vendor/swiper/swiper-bundle.min.js"></script>
```

```
<!-- Main JS File -->
```

```
<script src="assets/js/main.js"></script>
```

```
</body>
```

```
</html>
```


5. WOMEN'S HELPLINENUMBER

Women's Security

127.0.0.1:5000/user_helplinenumber

WOMEN'S HELPLINE NUMBER

Home

Women's helplinenumber

Create Contact

Emergency Contact

Emergency

Q

Womens Security.

Vidya

Dial "number" to get help

HELPLINE	PHONE NUMBER
Women Helpline-All India:	Call1091854685
Ranchi Commission for Women:	Call 011-96522145 Call 011-85462135
Outer Ranchi Helpline:	Call011-96584213 Call011-05469542

127.0.0.1:5000/user_dashboard

AVL - LIV

Video highlight

Search

ENG IN

21:21

13-05-2024

```

{% extends "user_layout.html" %}

{% block content %}
<style>
body{
    background-image: url(../static/image/helplinenumber.jpeg);
    background-size: cover;
    height: 90vh;
}
button{
    background-color: #d87878;
    color: white;
    cursor: pointer;
    margin: 12px 0;
    width: 20%;
    border-radius: 8px ;
    padding: 7px 15px;
}
</style>
<div class="container mybg mt-5 p-5 rounded">
    {% with messages = get_flashed_messages() %}
    {% if messages %}
    {% for message in messages %}
    <div class="alert alert-danger" role="alert">
        {{message}}
    </div>
    {% endfor %}
    {% endif %}
    {% endwith %}
    <h1> WOMEN'S HELPLINE NUMBER</h1>
    
    <h2>Dial "<u>number</u>" to get help </h2>

```

```
<form action="/user_helplinenumber" method="post" enctype="multipart/form-  
data"><br><br>
```

```
<center>
```

```
<section class="container table-section">
```

```
<table class="table table-striped table-bordered">
```

```
<thead class="table-primary"></thead>
```

```
<tr>
```

```
<th>HELPLINE</th>
```

```
<th>PHONE NUMBER</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Women Helpline-All India:</td>
```

```
<td><a href="tel:1091">Call</a>1091854685</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Ranchi Commission for Women:</td>
```

```
<td>
```

```
<a href="tel:011-96521452">Call</a>011-96522145<br>
```

```
<a href="tel:011-85462135">Call</a>011-85462135
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Outer Ranchi Helpline:</td>
```

```
<td><a href="tel:011-96584213">Call</a>011-96584213</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Student Helpline </td>
```

```
<td><a href="tel:011-95468542">Call</a>011-95468542</td>
```

```
</tr>
```

```
<tr>
```

```
<td>National Commission for Women</td>
```

```
<td>
  <a href="tel:011-23237166">Call</a>011-23237166<br>
  <a href="tel:011-23234918">Call</a>011-23234918
</td>
</tr>
```

```
</form action="">
```

```
</center>
```

```
</table>
```

```
{% endblock %}
```

7. CREATE CONTACT

The screenshot shows a web browser window with the title 'Womens Security' and a single tab. The address bar shows the URL '127.0.0.1:5000/user_create_contact'. The page has a navigation bar with links: 'Home', 'Women's helpline number', 'Create Contact', 'Emergency Contact', and 'Emergency'. The main heading is 'Womens Security. Create Contact' with a blue 'Vidya' logo. Below the heading is a form with the following fields: 'Name' (text input), 'Relation' (dropdown menu with 'Parents' selected), 'Mobile No.' (text input), and 'Email' (text input). A red 'Add Contact Now' button is at the bottom of the form. The Windows taskbar is visible at the bottom, showing the Start button, search bar, and various application icons. The system tray shows 'ENG IN', signal icons, and the date '13-05-2024'.

```
{% extends "user_layout.html" %}
```

```
{% block content %}
```

```
<div class="container mybg mt-5 p-5 rounded">
    {% with messages = get_flashed_messages() %}
    {% if messages %}
    {% for message in messages %}
    <div class="alert alert-danger" role="alert">
        {{ message }}
    </div>
    {% endfor %}
    {% endif %}
    {% endwith %}
    <h1>Create Contact</h1><hr>
    <form action="/user_create_contact" method="post" enctype="multipart/form-data">
    <div class="row">
        <div class="col">
            <label for="">Name</label>
            <input name="name" class="form-control" type="text">
        </div>
    </div>
    <div class="row">
        <div class="col">
            <label for="">Relation</label>
            <select name="relation" class="form-select">
                <option>Parents</option>
                <option>friends</option>
                <option>Teachers</option>
                <option>Cousines</option>
                <option>Others</option>
            </select>
        </div>
    </div>
</div>
```

```

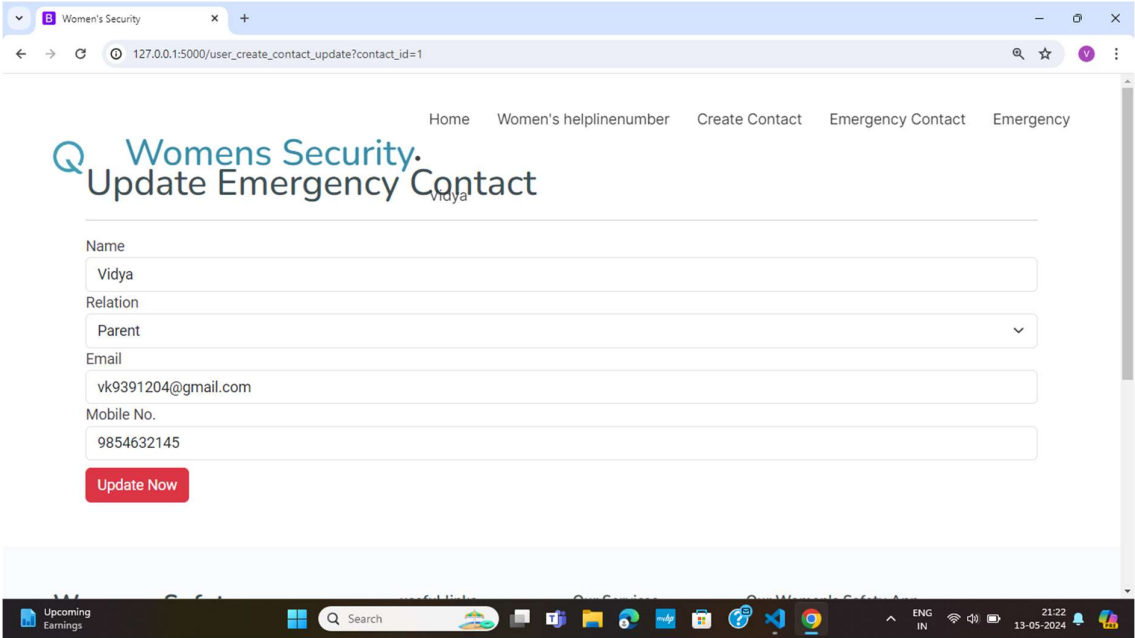
<div class="row">
  <div class="col">
    <label for="">Mobile No.</label>
    <textarea name="mobilenumber" class="form-control"></textarea>
  </div>
</div>

<div class="row">
  <div class="col">
    <label for="">Email</label>
    <textarea name="email" class="form-control"></textarea>
  </div>
</div>
<div class="row">
  <div class="col">
    <button type="submit" class="btn btn-danger mt-2">Add Contact Now</button>
  </div>
</div>
</form>

</div>
{% endblock %}

```

8. CREATE CONTACT SHOW




```

{% extends "user_layout.html" %}
{% block content %}
<div class="container mt-5 pt-5">
  <h1>EMERGENCY CONTACT</h1><hr>
  
  {% with messages = get_flashed_messages() %}
  {% if messages %}
  {% for message in messages %}
  <script>
    alert("{ {message} }")
  </script>
  {% endfor %}
  {% endif %}
  {% endwith %}

  <table class="table table-dark table-striped">
    <thead>
      <tr>
        <th scope="col">ID</th>
        <th scope="col">Name</th>
        <th scope="col">Relation</th>
        <th scope="col">Email</th>
        <th scope="col">Mobile No.</th>
        <th scope="col">Date Time</th>
        <th scope="col">Update</th>
        <th scope="col">Delete</th>
        <th scope="col">Comment</th>
      </tr>
    </thead>
    <tbody>
      {% for r in record %}
      <tr>
        <td>{{r[0]}}</td>

```

```

<td>{{r[1]}}</td>
<td>{{r[2]}}</td>
<td>{{r[3]}}</td>
<td>{{r[4]}}</td>
<td>{{r[5]}}</td>

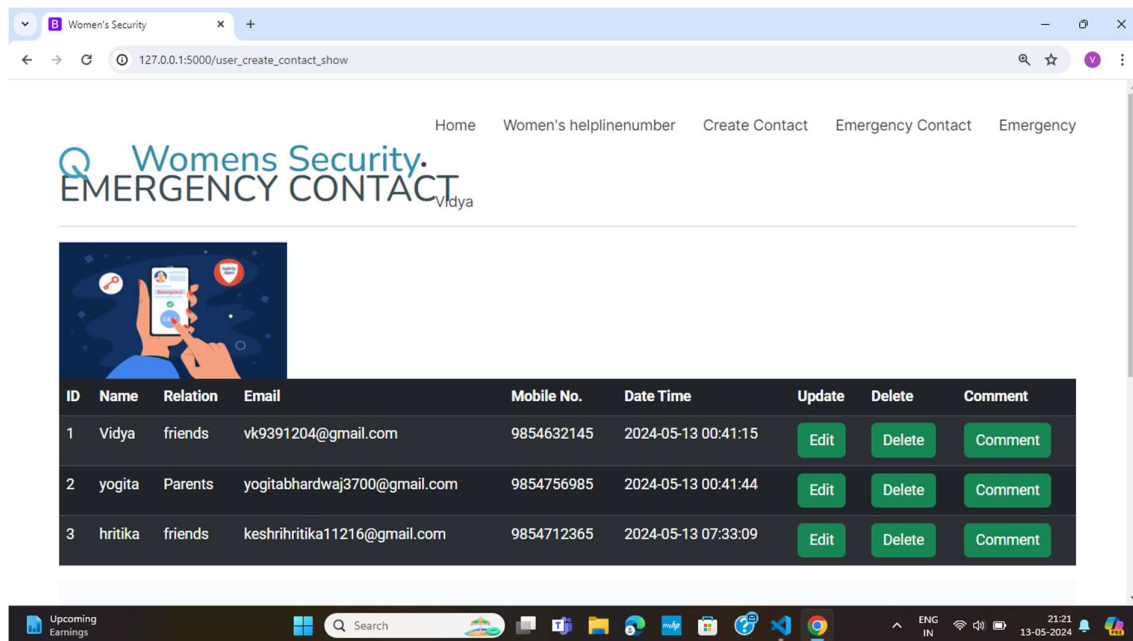
<td><a href="/user_create_contact_update?contact_id={{r[0]}}"><button class="btn
btn-success">Edit</button></a></td>
<td><a href="/user_create_contact_delete?contact_id={{r[0]}}"><button class="btn
btn-success">Delete</button></a></td>
<td><a href="/user_review?contact_id={{r[0]}}"><button class="btn btn-
success">Comment</button></a></td>
</tr>
{% endfor %}
</tbody>

</table>

{% endblock %}

```

9. CREATE CONTACT UPDATE



```
{% extends "user_layout.html" %}
```

```
{% block content %}
```

```
<div class="container mybg mt-5 p-5 rounded">
    {% with messages = get_flashed_messages() %}
    {% if messages %}
    {% for message in messages %}
    <div class="alert alert-danger" role="alert">
        {{message}}
    </div>
    {% endfor %}
    {% endif %}
    {% endwith %}
    <h1>Update Emergency Contact</h1><hr>
```

```

{% for r in record %}
<form action="/user_create_contact_update?contact_id={{r[0]}}" method="post">
<div class="row">
  <div class="col">
    <label for="">Name</label>
    <input name="name" class="form-control" value="{{r[1]}}" type="text">
  </div>
</div>
<div class="row">
  <div class="col">
    <label for="">Relation</label>
    <select name="relation" class="form-select">
      <option {% if(r[2]=='Parent') %} selected {% endif %}>Parent</option>
      <option {% if(r[2]=='Friends') %} selected {% endif %}>friends</option>
      <option {% if(r[2]=='Teachers') %} selected {% endif %}>Teachers</option>
      <option {% if(r[2]=='Cousin') %} selected {% endif %}>Cousin</option>
      <option {% if(r[2]=='Others') %} selected {% endif %}>Others</option>
    </select>
  </div>
</div>
<div class="row">
  <div class="col">
    <label for="">Email</label>
    <input email="email" class="form-control" value="{{r[3]}}" type="text">
  </div>
</div>
<div class="row">
  <div class="col">
    <label for="">Mobile No.</label>
    <input email="mobilenumber" class="form-control" value="{{r[4]}}"
type="text">
  </div>
</div>
<div class="row">

```

```
<div class="col">
  <button type="submit" class="btn btn-danger mt-2">Update Now</button>
</div>
</div>
</form>
{% endfor %}
</div>
{% endblock %}
```

10. EMERGENCY CONTACT

Women's Security

127.0.0.1:5000/user_helplinenumber

Home Women's helpline number Create Contact Emergency Contact Emergency

Q Womens Security.

Vidya

HELPLINE	PHONE NUMBER
Women Helpline-All India:	Call1091854685
Ranchi Commission for Women:	Call 011-96522145 Call 011-85462135
Outer Ranchi Helpline:	Call011-96584213
Student Helpline	Call011-95468542
National Commission for Women	Call011-23237166 Call011-23234918

127.0.0.1:5000/user_profile

AVL - LIV
Video highlight

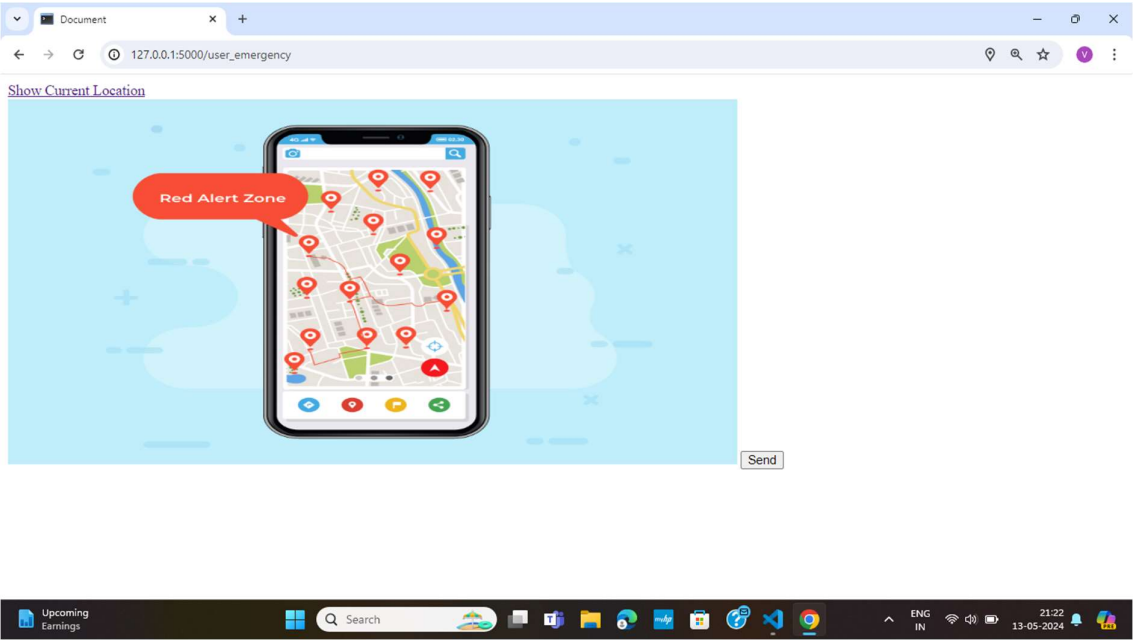
Search

ENG
IN

21:21
13-05-2024

```
{% extends "user_layout.html" %}
{% block content %}
<div class="container mt-5 pt-5">
  {% for r in record %}
    <h5 class="text-primary">Name: <span class="text-dark">{{r[0]}}</span></h5>
    <h6 class="text-danger">Relation: {{r[1]}}</h6>
    <div>{{r[3]}}</div>
  {% endfor %}
</div>
{% endblock %}
```

10. EMERGENCY

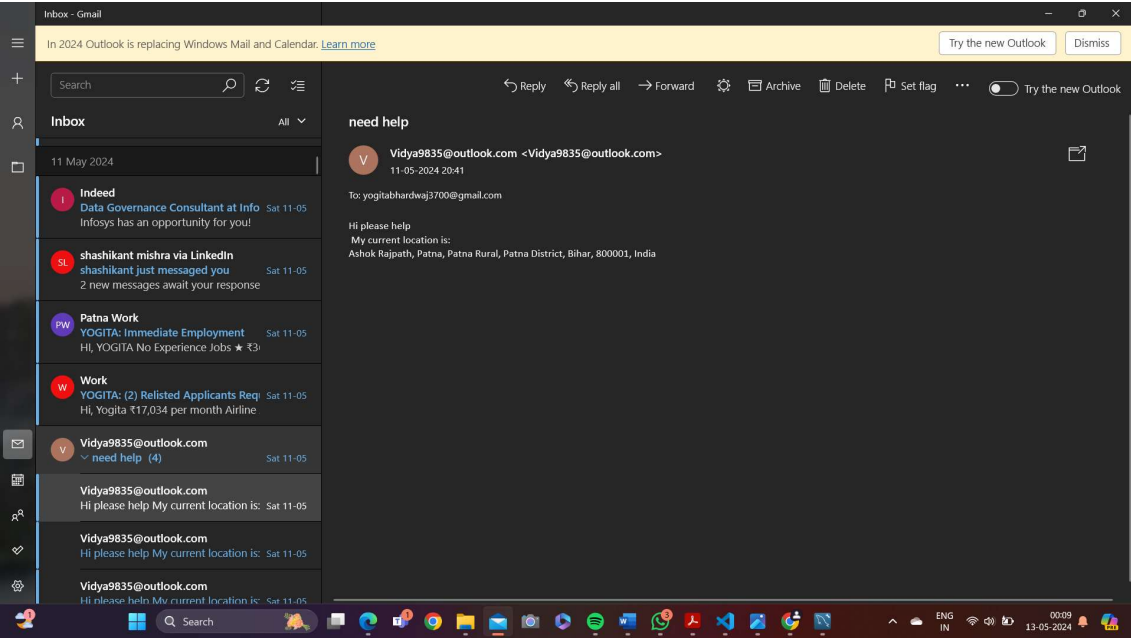



```
# importing modules
from geopy.geocoders import Nominatim

def loc(log,lat):
    # calling the nominatim tool
    geoLoc = Nominatim(user_agent="GetLoc")
    # l=str(lat)+str(log)
    # passing the coordinates
    locname = geoLoc.reverse(lat+","+log)

    # printing the address/location name
    return locname
```

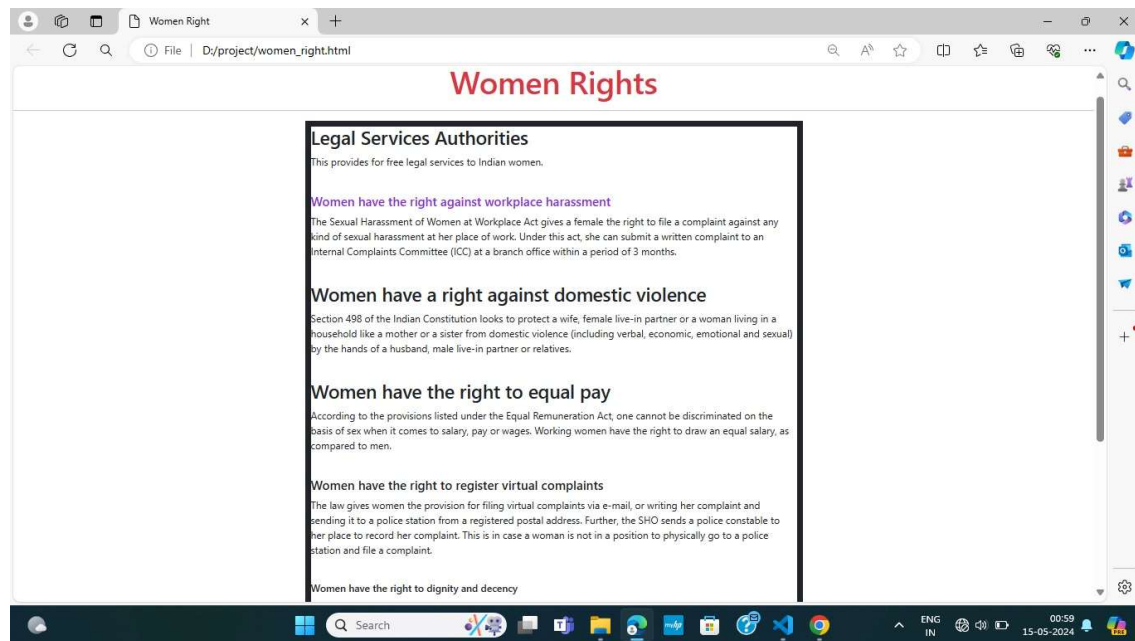
11. RECIVE MAIL



12. COME TO DASHBOARD



13. WOMENS RIGHT



```

<!DOCTYPE html>
<html lang="en">
{% extends "user_layout.html" %}

{% block content %}

<br>
<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>Women Right</title>
  <!-- CSS only -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">

    <style>
      div{
        border: solid;
        width: 800px;
        height: 1100px;
        border-width: 10px;
        margin: auto;
      }
      h1{
        color: blueviolet;
        font-size: 50px;
      }
      h2{
        font-size: 30px;
      }
      h3{
        color: blueviolet;
        font-size: 20px;
      }
      h4{
        font-size: 30px;
      }
      h5{
        font-size: 30px;
      }
      h6{
        font-size: 30px;
      }
    </style>
  </head>
  <body>
    <div>
      <h1>Women Right</h1>
      <h2>What is Women Right</h2>
      <h3>Why Women Right</h3>
      <h4>How to achieve Women Right</h4>
      <h5>What is the importance of Women Right</h5>
      <h6>What are the challenges of Women Right</h6>
    </div>
  </body>
</html>

```

```

        h7{
            font-size: 30px;
        }
    </style>

</head>
<body>
    <center>
        <h1 class="text-danger">Women Rights</h1><hr></center>
        <form action="/women_right" method="POST">
            <div class="box">
                <h2>Legal Services Authorities</h2>
                <p>This provides for free legal services to Indian women.</p>

                <br>
                <h3>Women have the right against workplace harassment</h3>
                <p>The Sexual Harassment of Women at Workplace Act gives a female
the right to file a complaint
                against any kind of sexual harassment at her place of work.
Under this act, she can submit a
                written complaint to an Internal Complaints Committee (ICC) at
a branch office within a period
                of 3 months.</p>
                <br>
                <h4>Women have a right against domestic violence</h4>
                <p>Section 498 of the Indian Constitution looks to protect a wife,
female live-in partner or a woman
                living in a household like a mother or a sister from domestic
violence (including verbal,
                economic, emotional and sexual) by the hands of a husband,
male live-in partner or relatives.</p>

                <br>
                <h4>Women have the right to equal pay</h4>
                <p>According to the provisions listed under the Equal Remuneration
Act, one cannot be discriminated
                on the basis of sex when it comes to salary, pay or wages.
Working women have the right to draw
                an equal salary, as compared to men.</p>

                <br>

                <h5>Women have the right to register virtual complaints</h5>
                <p>The law gives women the provision for filing virtual complaints
via e-mail, or writing her
                complaint and sending it to a police station from a registered
postal address. Further, the SHO

```

sends a police constable to her place to record her complaint.
This is in case a woman is not in
a position to physically go to a police station and file a
complaint.</p>

<h6>Women have the right to dignity and decency</h6>

<p>In an event that the accused is a woman, any medical
examination procedure on her must be
performed by -- or in the presence of -- another woman.</p>

<h7>Women have the right to register virtual complaints</h7>

<p>The law gives women the provision for filing virtual complaints
via e-mail, or writing her
complaint and sending it to a police station from a registered
postal address. Further, the SHO
sends a police constable to her place to record her complaint.
This is in case a woman is not in
a position to physically go to a police station and file a
complaint.</p>

</form>

</div>

</body>

</html>

{% endblock %}

APP.PY

```
from flask import Flask,render_template,request,flash,redirect,url_for,session
from user import User_Operation
from encryption import Encryption
from myemail import Email
from myrandom import randomnumber
from validate import myvalidate
from admin import adminOperation
from datetime import datetime
import find_location
```

```
app = Flask(__name__) #flask object
app.secret_key="jdkjhdkjdsf"
```

```
userobj =User_Operation() #user object
#-----email object-----
emailobj = Email(app)
#-----
validobj = myvalidate() #validation object
```

```
adminobj = adminOperation() # object of admin
```

```
@app.route('/')
def index():
    return render_template('index.html')
```

```
@app.route('/user_signup',methods=['POST','GET'])
def user_signup():
    if(request.method=="GET"):
        return render_template('user_signup.html')
    else:
        name= request.form['name']
        email= request.form['email']
        mobile= request.form['mobile']
        password= request.form['password']
        #-----validation-----
        frm = [name,email,mobile,password]
        if(not validobj.required(frm)):
```



```

        flash("field can't be empty!!")
        return redirect(url_for('user_signup'))

    if(not validobj.alphacheck(name)):
        flash("Name must be alphabet")
        return redirect(url_for('user_signup'))

    if(not validobj.digitcheck(mobile)):
        flash("mobile must be 10 digit")
        return redirect(url_for('user_signup'))
        #-----password encryption-----
    e = Encryption() #object of encryption class
    password = e.convert(password)
    #-----
    userobj.user_signup(name,email,mobile,password)
    #-----Email send-----
    global otp
    otp=randomnumber()
    subject="app mail verification code"
    message="hi"+name+"\n your OTP is;" +str(otp)+"\n thank you \n app"
    #emailobj.compose_mail('Dezinex Technology Letter',email,'WElcome to
oir company \n thank u\n dezinex')
    emailobj.compose_mail(subject,email,message)
    #-----
    flash("OTP send to your register email ID")
    return redirect(url_for('otpverify',email=email))

@app.route('/otpverify',methods=['POST','GET'])
def otpverify():
    if(request.method=='GET'):
        email = request.args.get('email')
        return render_template('otpverify.html',email=email)
    else:
        userotp = request.form['otp']
        useremail = request.form['email']
        if(str(otp)==userotp):
            flash("successfully regisered..Login Now")
            return redirect(url_for('user_login'))
        else:
            userobj.user_delete(useremail)
            flash("Email is not verified.. please register again")
            return redirect(url_for('user_signup'))

@app.route('/user_login',methods=['POST','GET'])
def user_login():
    if(request.method=='GET'):
        return render_template('user_login.html')

```

```

else:
    email=request.form['email']
    password=request.form['password']
    #-----password encryption-----
    e = Encryption() # objrct of Encryption class
    password = e.convert(password)
    #-----user login-----
    ob = User_Operation() #object of UserOP class in user module
    r = ob.user_login(email,password)
    if (r==0):
        flash("invalid email id and password")
        return redirect(url_for('user_login'))
    else:
        return redirect(url_for('user_dashboard'))

@app.route('/user_delete',methods=['POST','GET'])
def user_delete():
    if('user_email' in session):
        if(request.method=='GET'):
            userobj.user_delete(session['user_email'])
            flash("Your account deleted successfully...Thank You!!")
            return redirect(url_for('user_signup'))
    else:
        flash("you can't access this page...login now to continue..")
        return redirect(url_for('user_login'))

@app.route('/user_logout',methods=['POST','GET'])
def user_logout():
    if('user_email' in session):
        if(request.method=='GET'):
            session.clear() # it will destroy all user's session
            flash("successfully Logged out!!")
            return redirect(url_for('user_login'))
    else:
        flash("you can't access this page...login now to continue..")
        return redirect(url_for('user_login'))

@app.route('/user_dashboard',methods=['POST','GET'])
def user_dashboard():
    if('user_email' in session):
        if(request.method=='GET'):
            return render_template('user_dashboard.html')
    else:
        flash("you can't access this page...login now to continue..")
        return redirect(url_for('user_login'))

@app.route('/user_profile',methods=['POST','GET'])

```

```

def user_profile():
    if('user_email' in session):
        if(request.method=='GET'):
            record = userobj.user_profile()
            return render_template('user_profile.html',record=record)
        else:
            name = request.form['name']
            mobile = request.form['mobile']
            userobj.user_profile_update(name,mobile)
            flash('your profile is updated successfully!!')
            return redirect(url_for('user_profile'))
    else:
        flash("you can't access this page...login now to continue..")
        return redirect(url_for('user_login'))

#-----account delete-----

@app.route('/user_password',methods=['POST','GET'])
def user_password():
    if('user_email' in session):
        if(request.method=='GET'):
            return render_template('user_password.html')
        else:
            oldpassword = request.form['oldpassword']
            newpassword = request.form['newpassword']
            #-----encryption-----
            frm = [oldpassword,newpassword]
            if(not validobj.required(frm)):
                flash("field can't be empty!!")
                return redirect(url_for('user_password'))
            #-----encryprion-----
            e = Encryption()
            oldpassword = e.convert(oldpassword)
            newpassword = e.convert(newpassword)
            r= userobj.user_password(oldpassword,newpassword)
            if(r==0):
                flash("your curent password is invalid")
                return redirect(url_for('user_password'))
            else:
                flash("your password is changed successfully!!")
                return redirect(url_for('user_logout'))
    else:
        flash("you are not authorized to access this page.. please login
continue..")
        return redirect(url_for('user_login'))

@app.route('/user_helplinenumner',methods=['POST','GET'])

```

```

def user_helplinenumber():
    if('user_email' in session):
        if(request.method=='GET'):
            return render_template('user_helplinenumber.html')
        else:
            name = request.form['name']
            relation = request.form['relation']
            email = request.form['email']
            photo = request.form['photo']
            mobilenumber = request.form['mobilenumber']

            #-----validate-----
            frm = [name, relation,email,photo,mobilenumber]
            if(not validobj.required(frm)):
                flash("field can't be empty!!")
                return redirect(url_for('user_helplinenumber'))

            photoname =photo.filename
            if(photoname==''):
                flash("please upload your photo")
                return redirect(url_for('user_helplinenumber'))

            d = datetime.now() #current date time
            t=int(round(d.timestamp()))
            photoname = str(t)+'.'+photoname.split('.')[1]
            photo.save('static/user/'+photoname)

            #-----
            userobj.user_signup(name,relation,email,photo,mobilenumber)
            flash("Successfully Registered..Share Your information Now!!")
            return redirect(url_for('user_login'))

@app.route('/user_create_contact',methods=['POST','GET'])
def user_create_contact():
    if('user_email' in session):
        if(request.method=='GET'):
            return render_template('user_create_contact.html')
        else:
            name = request.form['name']
            relation = request.form['relation']
            email = request.form['email']
            mobilenumber = request.form['mobilenumber']

            #-----validate-----
            frm = [name,relation,email,mobilenumber]
            if( not validobj.required(frm)):
                flash("field can't be empty!!")

```

```

        return redirect(url_for('user_create_contact'))
#-----
        userobj.user_create_contact(name,relation,email,mobilenumber)
        flash("Successfully Registered..Share Your contact Now!!")
        return redirect(url_for('user_create_contact'))
    else:
        flash("You can't access this page...login now to continue")
        return redirect(url_for('user_login'))

@app.route('/user_create_contact_show',methods=['POST','GET'])
def user_create_contact_show():
    if('user_email' in session):
        if(request.method=='GET'):
            record=userobj.user_create_contact_show()
            return
    render_template('user_create_contact_show.html',record=record)

    else:
        flash("you can't access this page..login now to continue..!!")
        return redirect(url_for('user_login'))

@app.route('/user_create_contact_update',methods=['POST','GET'])
def user_create_contact_update():
    if('user_email' in session):
        if(request.method=='GET'):
            contact_id=request.args.get('contact_id')
            record = userobj.user_create_contact_update_show(contact_id)
            return
    render_template('user_create_contact_update.html',record=record)

    else:
        contact_id=request.args.get('contact_id')
        name = request.form['name']
        relation = request.form['relation']
        email = request.form['email']
        mobilenumber = request.form['mobilenumber']

        #-----validate-----
        frm = [name,relation,email,mobilenumber]
        if(not validobj.required(frm)):
            flash("field can't be empty!!")
            return
    redirect(url_for('user_create_contact_update.html',contact_id=contact_id))

#-----

```

```

        userobj.user_create_contact_update(name,relation,email,mobilenumbe
r,contact_id)
        flash("emergency contact is updated successfully!!")
        return
    redirect(url_for('user_create_contact_update',contact_id=contact_id))
    else:
        flash("you can't access this page....login now to continue....")
        return redirect(url_for('user_login'))

@app.route('/user_create_contact_delete',methods=['POST','GET'])
def user_create_contact_delete():
    if('user_email' in session):
        if(request.method=='GET'):
            contact_id=request.args.get('contact_id')
            userobj.user_create_contact_delete(contact_id)
            flash("Your add contact is deleted successfully!!")
            return redirect(url_for('user_create_contact_show'))
        else:
            flash("you can't access this page....login now to continue....")
            return redirect(url_for('user_login'))

@app.route('/user_emergency',methods=['GET','POST'])
def user_emergency():
    if(request.method=='GET'):
        return render_template('user_emergency.html')
    if(request.method=='POST'):
        log = request.form['log']
        lat = request.form['lat']
        address =str(find_location.loc(log,lat))
        email= userobj.user_emergency()
        for e in email:
            email = e[0]
            subject="need help"
            message="Hi please help \n My current location is: \n"+address
            emailobj.compose_mail(subject,email,message)
        flash("Emergency email send successfully!!")
        return redirect(url_for('user_dashboard'))
    else:
        flash("you can't access this page....login now to continue....")
        return redirect(url_for('user_login'))

@app.route('/user_emergency_view',methods=['POST','GET'])
def user_emergency_view():
    if('user_email' in session):
        if(request.method=='GET'):

```

```
        record = userobj.user_emergency_view()
        return render_template('user_emergency_view.html',record=record)

    else:
        flash("you can't access this page...login now to continue..")
        return redirect(url_for('user_login'))


@app.errorhandler(404)
def error404(e):
    return "Page Not Found"

if __name__=='__main__':
    app.run(debug=True)
```

2.USER.PY

```
import mysql.connector

from flask import session
from datetime import datetime

class User_Operation:
    def connection(self):
        con=mysql.connector.connect(host="localhost",port="3306",user="root",password="root",database="project")
        return con

    #-----user signup-----
    def user_signup(self,name,email,mobile,password):
        db=self.connection()
        cur = db.cursor()
        sq="insert into user(name,email,mobile,password) values (%s,%s,%s,%s)"
        record=[name,email,mobile,password]
        cur.execute(sq,record)
        db.commit() #to save perammenrin db
        cur.close()
        db.close()
        return

    #-----data delete-----
    def user_delete(self,email): #delete data
        db = self.connection()
        cur = db.cursor()
        sq = "delete from user where email=%s"
        record=[email]
        cur.execute(sq,record)
        db.commit() #to save permanent in db
        cur.close() #to save permanent in db
        db.close()
```



```

    return

#-----data retrieve-----
def user_login(self,email,password):
    db=self.connection()
    cur=db.cursor()
    sq="select name,email from user where email=%s and password=%s"
    record=[email,password]
    cur.execute(sq,record)
    row=cur.fetchall()
    rc=cur.rowcount #count no.of record
    cur.close()
    db.close()
    if(rc==0):
        return 0
    else:
        session['user_name']=row[0][0]
        session['user_email']=row[0][1]
        return 1

#-----user profile-----
def user_profile(self):
    db = self.connection()
    cur =db.cursor()
    sq = "select name,email,mobile from user where email=%s"
    record=[session['user_email']]
    cur.execute(sq,record)
    row = cur.fetchall()
    cur.close()
    db.close()
    return row

#-----update profile-----
def user_profile_update(self,name,mobile):
    db = self.connection()

```

```

cur=db.cursor()
sq = "update user set name=%s,mobile=%s=where email=%s"
record= [name,mobile,session['user_email']]
cur.execute(sq,record)
db.commit()
session['user_name']=name
cur.close()
db.close()
return

```

```

def user_password(self,oldpassword,newpassword):
    db = self.connection()
    cur=db.cursor()
    sq = "select * from user where email=%s and password=%s"
    record=[session['user_email'],oldpassword]
    cur.execute(sq,record)
    row = cur.fetchall()
    rc = cur.rowcount
    if(rc==0):
        return 0

```

```

sq = "update user set password=%s where email=%s"
record=[newpassword,session['user_email']]
cur.execute(sq,record)
db.commit()
cur.close()
db.close()
return row

```

```

def user_helplinenumber(self):
    db=self.connection()
    mycursor=db.cursor()
    sq="select user_email,email,name,city,photo from user where user_email=%s"
    record=[session['user_email']]

```

```

mycursor.execute(sq,record)
row = mycursor.fetchall()
mycursor.close()
db.close()
return row

def user_create_contact(self,name,relation,email,mobilenumber):
    db=self.connection()
    mycursor=db.cursor()
    sq="insert into contact (user_email,name,relation,email,mobilenumber,creator_at)
values(%s,%s,%s,%s,%s,%s)"
    creator_at=datetime.now()
    record=[session['user_email'],name,relation,email,mobilenumber,creator_at]
    mycursor.execute(sq,record)
    db.commit()
    mycursor.close()
    db.close()
    return

def user_create_contact_show(self):
    db=self.connection()
    mycursor=db.cursor()
    sq="select contact_id,name,relation,email,mobilenumber,creator_at from contact where
user_email=%s"
    record=[session['user_email']]
    mycursor.execute(sq,record)
    row = mycursor.fetchall()
    return row

def user_create_contact_delete(self,name_id):
    db=self.connection()
    mycursor=db.cursor()
    sq="delete from contact where contact_id=%s"
    record=[name_id]

```

```
mycursor.execute(sq,record)
```

```
db.commit()
```

```
mycursor.close()
```

```
db.close()
```

```
return
```

```
def user_create_contact_update_show(self,contact_id):
```

```
    db=self.connection()
```

```
    mycursor=db.cursor()
```

```
        sq="select  contact_id,name,relation,email,mobilenumber  from  contact  where  
contact_id=%s"
```

```
    record=[contact_id]
```

```
    mycursor.execute(sq,record)
```

```
    row = mycursor.fetchall()
```

```
    mycursor.close()
```

```
    db.close()
```

```
    return row
```

```
def user_create_contact_update(self,name,relation,email,mobilenumber,contact_id):
```

```
    db=self.connection()
```

```
    mycursor=db.cursor()
```

```
        sq="update          contact          set  
name=%s,relation=%s,email=%s,mobilenumber=%s,contact_id=%s"
```

```
    record=[name,relation,email,mobilenumber,contact_id]
```

```
    mycursor.execute(sq,record)
```

```
    db.commit()
```

```
    mycursor.close()
```

```
    db.close()
```

```
    return
```

```
def user_emergency(self):
```

```
    db=self.connection()
```

```

mycursor=db.cursor()
sq ="select email from contact where user_email=%s"
record=[session['user_email']]
mycursor.execute(sq,record)
row =mycursor.fetchall()
mycursor.close()
db.close()
return row

def user_emergency_view(self):
    db=self.connection()
    mycursor=db.cursor()
    sq ="select contact_id,name,relation,email,mobilenumber from contact order by
admin_at desc"
    mycursor.execute(sq)
    row = mycursor.fetchall()
    mycursor.close()
    db.close()
    return row

def user_emergency_detail(self,contact_id):
    db=self.connection()
    mycursor=db.cursor()
    sq ="select name,relation,email,mobilenumber from contact where contact_id=%s"
    record=[contact_id]
    mycursor.execute(sq,record)
    row = mycursor.fetchall()
    mycursor.close()
    db.close()
    return row

def user_review_insert(self,contact_id,comment,star):
    db=self.connection()

```

```
mycursor=db.cursor()
sq="insert into review (user_email,contact_id,comment,star) values(%s,%s,%s,%s)"
record=[session['user_email'],contact_id,comment,star]
mycursor.execute(sq,record)
db.commit()
mycursor.close()
db.close()
return
```

```
def user_review(self,contact_id):
    db=self.connection()
    mycursor=db.cursor()
    sq ="select star,name,comment from review r,user u where r.user_email=u.email and
contact_id=%s"
    record=[contact_id]
    mycursor.execute(sq,record)
    row = mycursor.fetchall()
    mycursor.close()
    db.close()
    return row
```

3. ENCRYPTION.PY

```
import hashlib
```

```
class Encryption:
```

```
    def convert(self,data):
```

```
        data=hashlib.md5(data.encode())
```

```
        data=data.hexdigest()
```

```
    return data
```

4.MYRANDON.PY

```
import random
```

```
def randomnumber():
```

```
    num=random.randrange(1000,9999)
```

```
    return num
```

```
def randomalphanum():
```

```
    charvalue="ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
```

```
    s=""
```

```
    for i in range(1,5):
```

```
        index=random.randrange(0,36)
```

```
        s=s+charvalue[index]
```

```
    return s
```


5. MYEMAIL.PY

```
from flask_mail import *
```

```
class Email:
```

```
    def __init__(self,app): #constructor or intialized
```

```
        #-----mail configuration-----
```

```
        app.config["MAIL_SERVER"]='smtp.office365.com'
```

```
        app.config["MAIL_PORT"]='587'
```

```
        app.config["MAIL_USERNAME"]='Vidya9835@outlook.com' #put your outlook email
```

```
id
```

```
        app.config["MAIL_PASSWORD"]='@Vidya6201'
```

```
        app.config["MAIL_USE_TLS"]=True
```

```
        app.config["MAIL_USE_SSL"]=False
```

```
        self.mail=Mail(app) #mail object
```

```
        #-----
```

```
    def compose_mail(self,subject,email,message):
```

```
        msg=Message(subject,sender ='Vidya9835@outlook.com',recipients =[email])
```

```
        msg.body=message
```

```
        self.mail.send(msg)
```

6. VALIDATE.PY

```
class myvalidate:
    def required(self,frm):
        for f in frm:
            if f=="":
                return False
            return True
    def alphacheck(self,data):
        if(data.isalpha()):
            return True
        return False
    def digitcheck(self,data):
        if(data.isdigit() and len(data)==10):
            return True
        return False
```

8.DATA DICTIONARY

The data dictionary is useful in case of development.

Table name: USER

Primary Key: Name ,User_id

Description: User can add our data

S.No.	Field Name	Size	Type	Description
1.	name	255	VARCHAR	Primary key
2.	email	225	VARCHAR	
3.	mobile	225	VARCHAR	UQ
4.	password	225	VARCHAR	

Table name: CONTACT

Primary Key: Contact_id

Description: user add there contact details

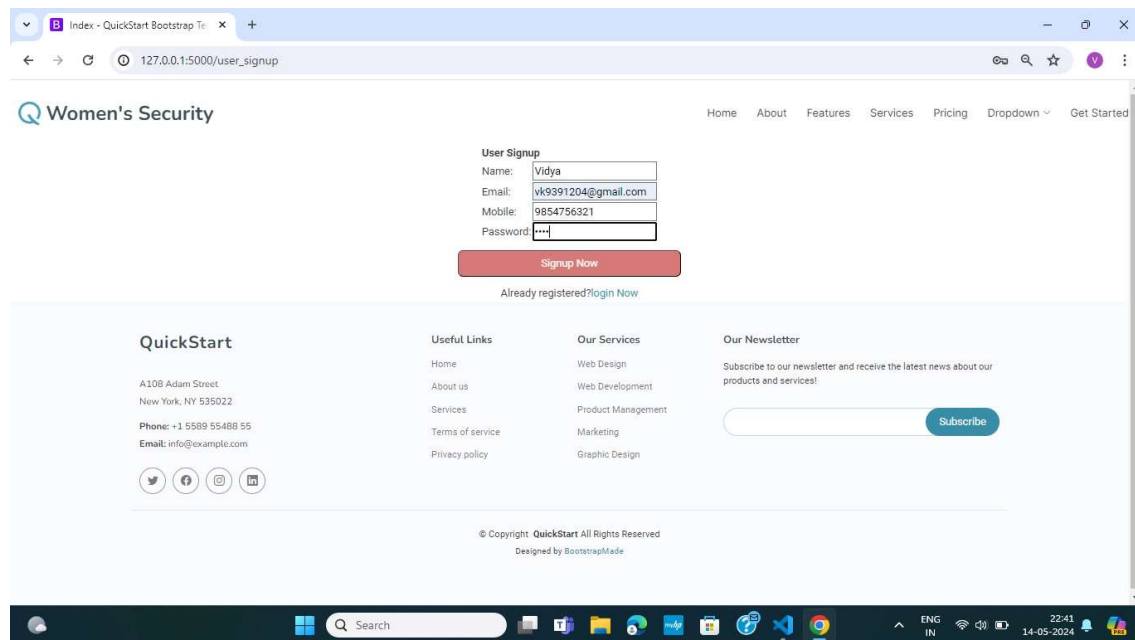
S.No.	Field Name	Size	Type	Description
1.	Contact_id		INT	Primary Key
2.	User_email	225	VARCHAR	
3.	name	225	VARCHAR	
4.	relation	225	VARCHAR	
5.	email		MEDIUMTEXT	
6.	mobilenumber		MEDIUMTEXT	
7.	Creator_at		DATETIME	
8.	city	225	VARCHAR	

9. OUTPUT

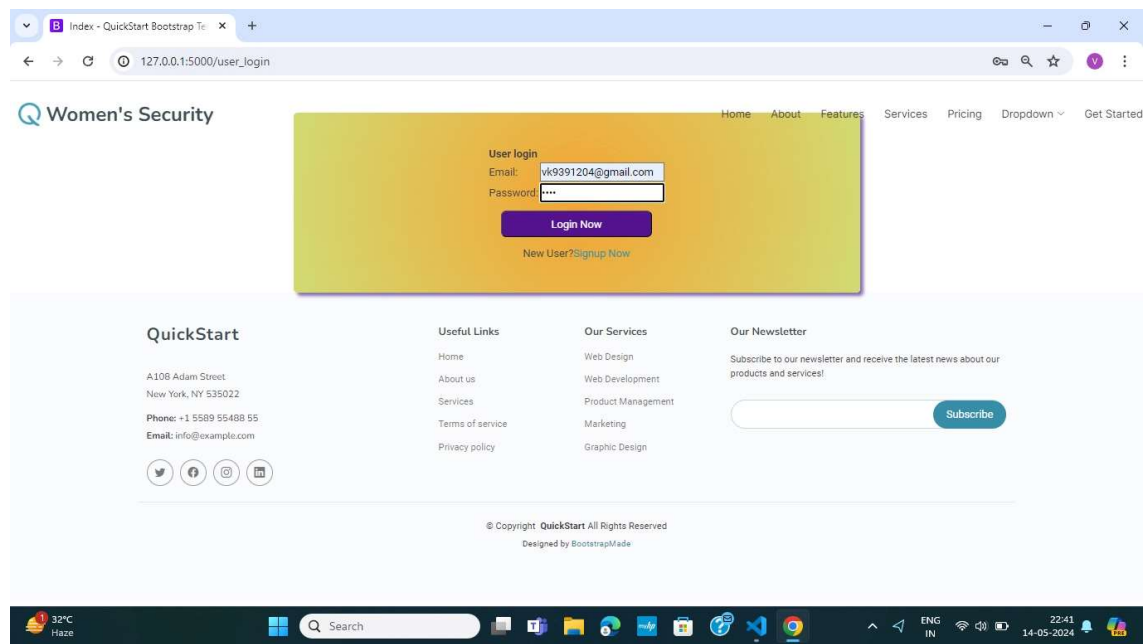
1.



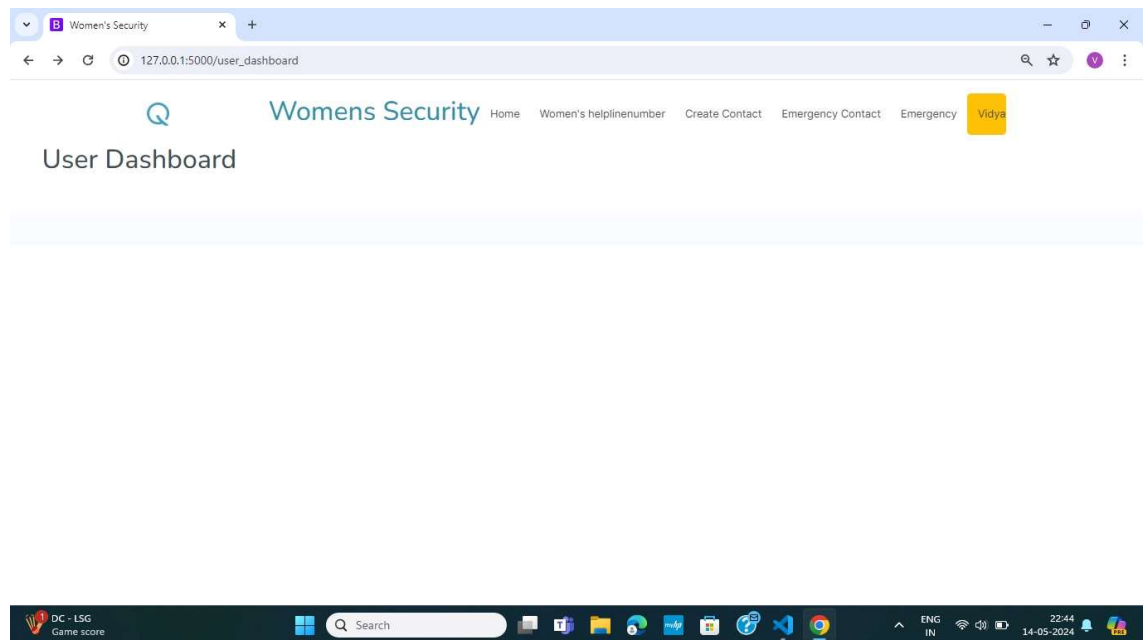
2.



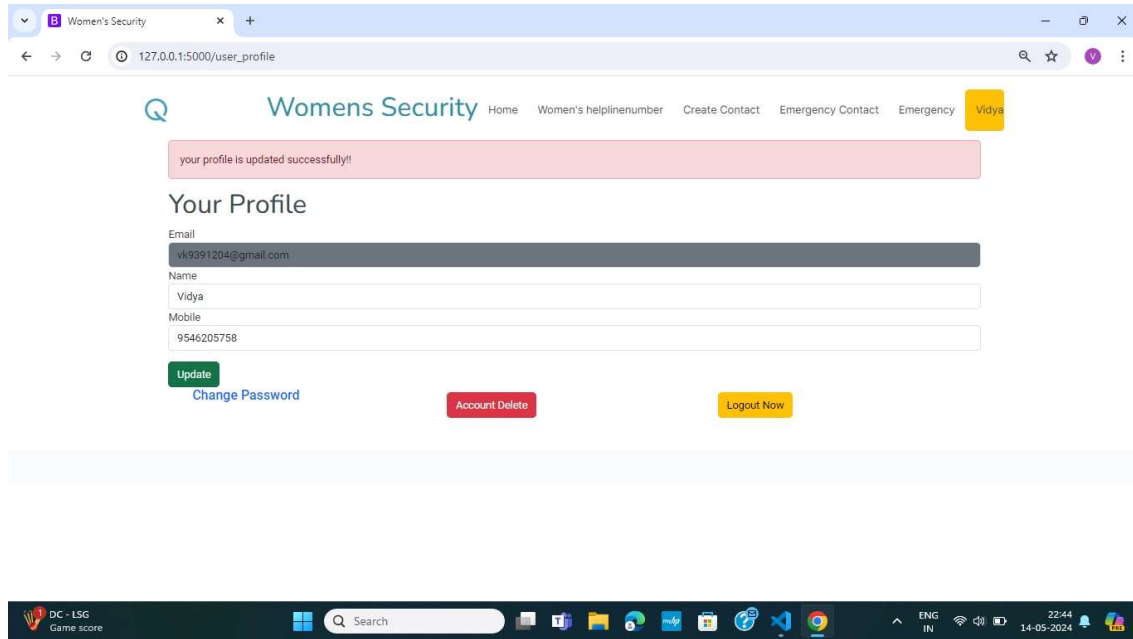
3.



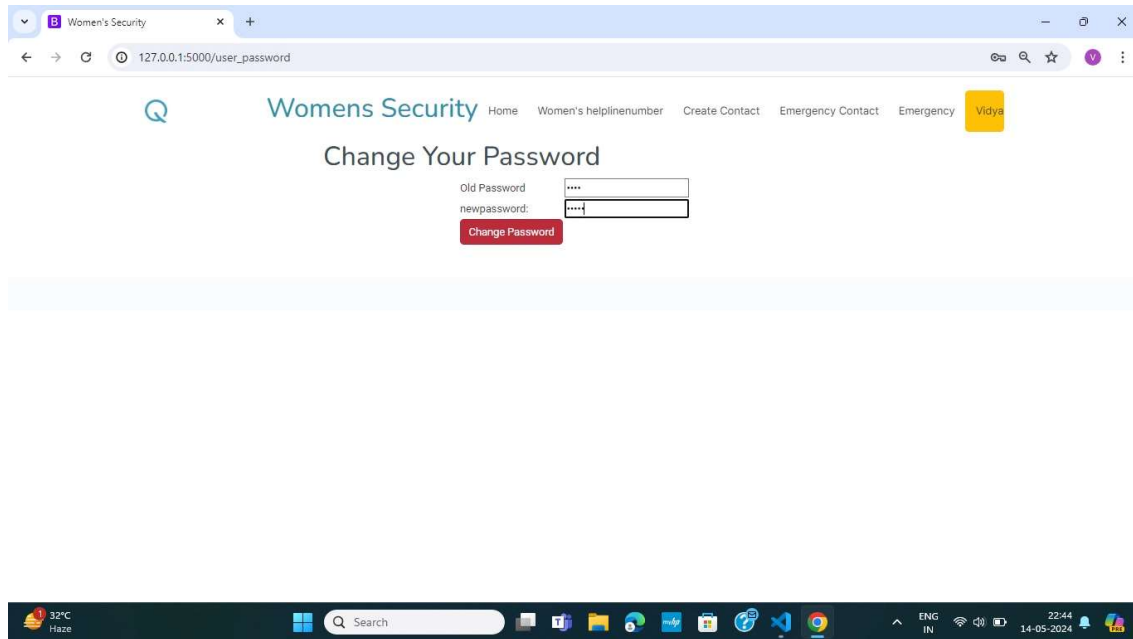
4.



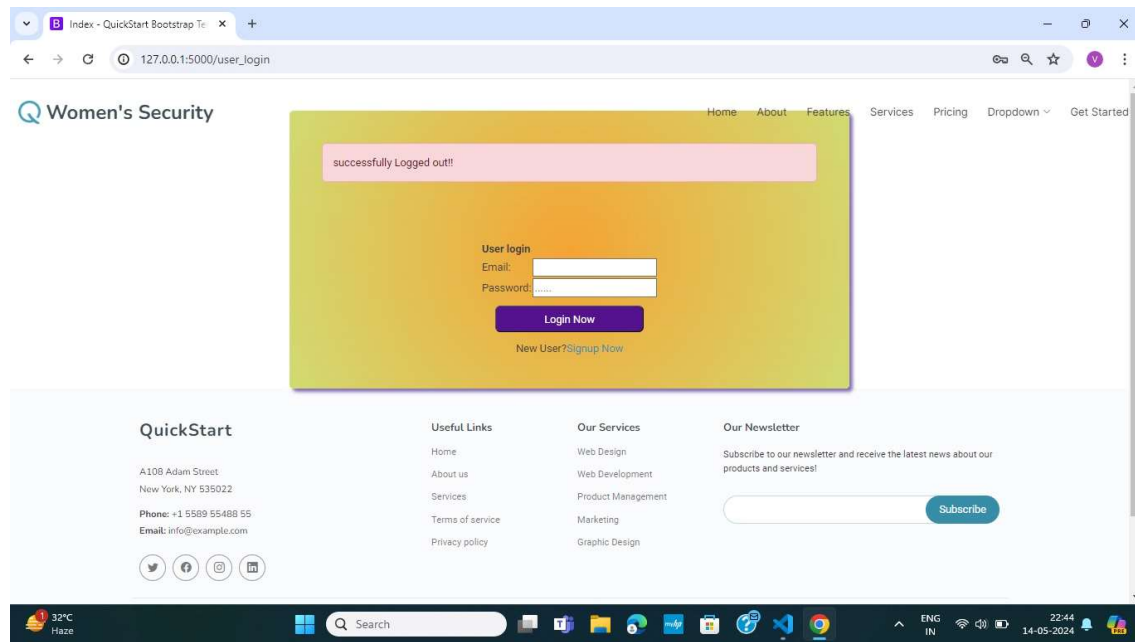
5.



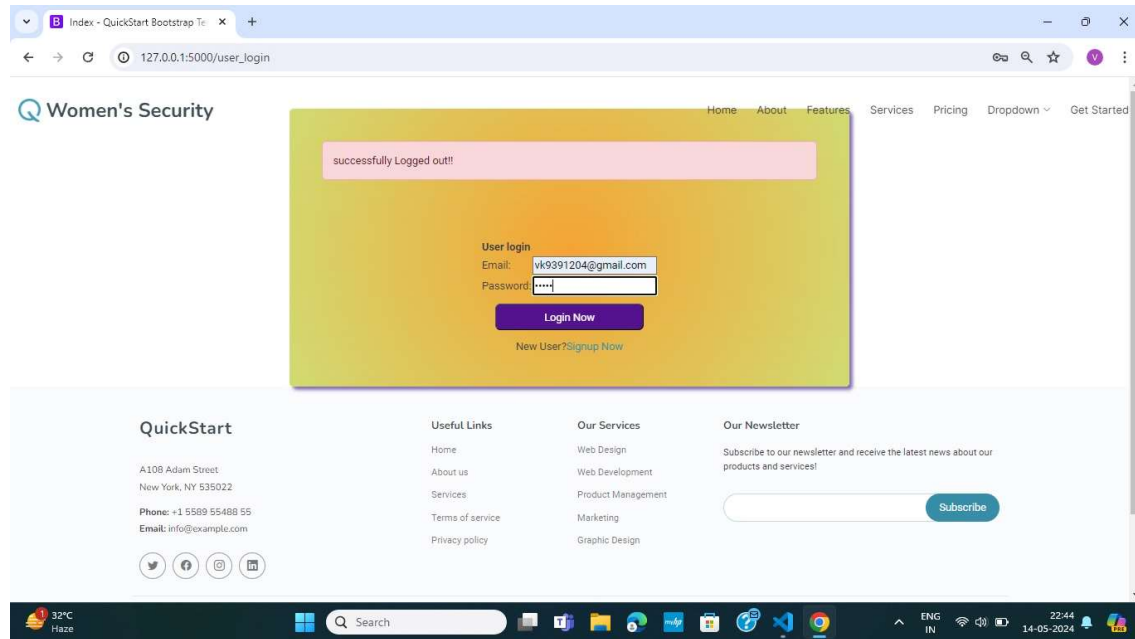
6.



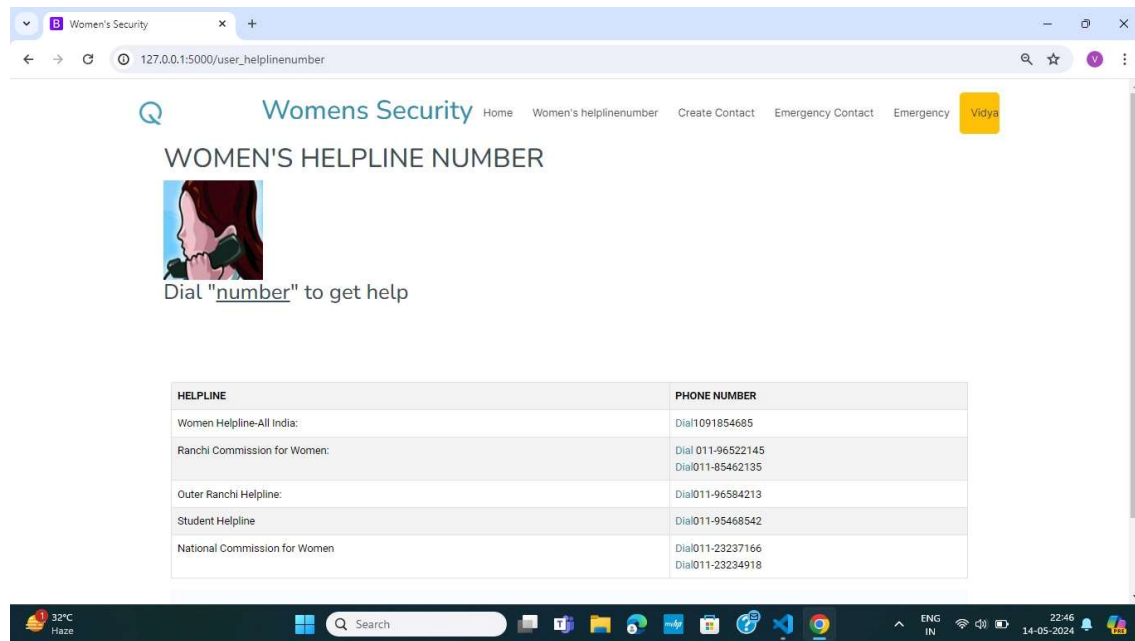
7.



8.



9.



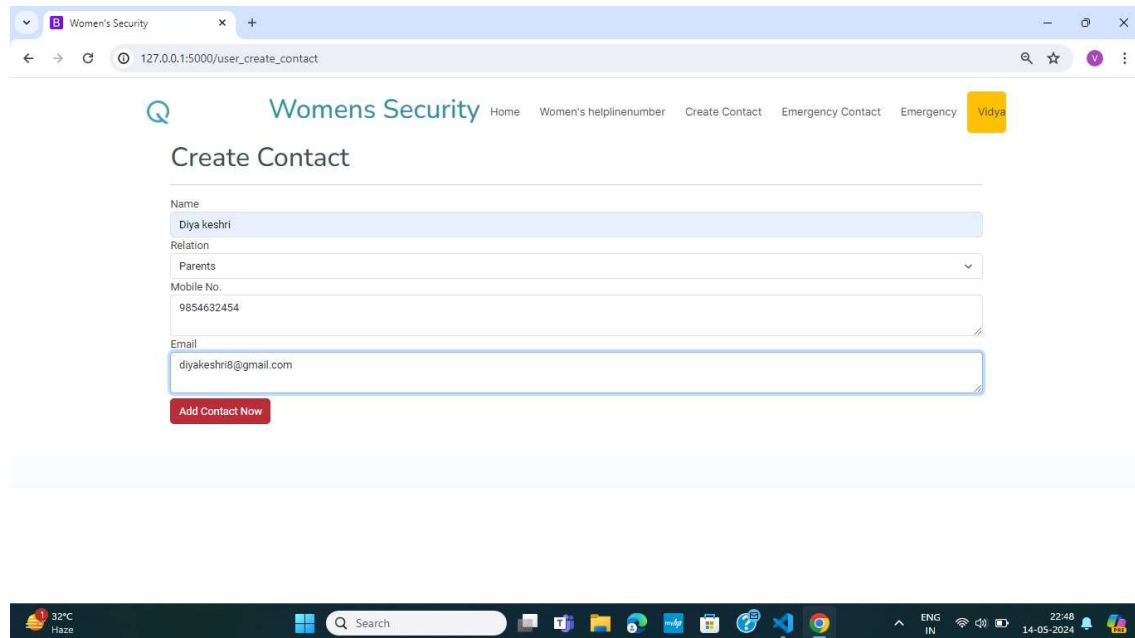
Womens Security Home Women's helpline number Create Contact Emergency Contact Emergency Vidya

WOMEN'S HELPLINE NUMBER

Dial "number" to get help

HELPLINE	PHONE NUMBER
Women Helpline-All India:	Dial 1091854685
Ranchi Commission for Women:	Dial 011-96522145 Dial 011-85462135
Outer Ranchi Helpline:	Dial 011-96584213
Student Helpline	Dial 011-95468542
National Commission for Women	Dial 011-23237166 Dial 011-23234918

10.



Womens Security Home Women's helpline number Create Contact Emergency Contact Emergency Vidya

Create Contact

Name
Diya keshri

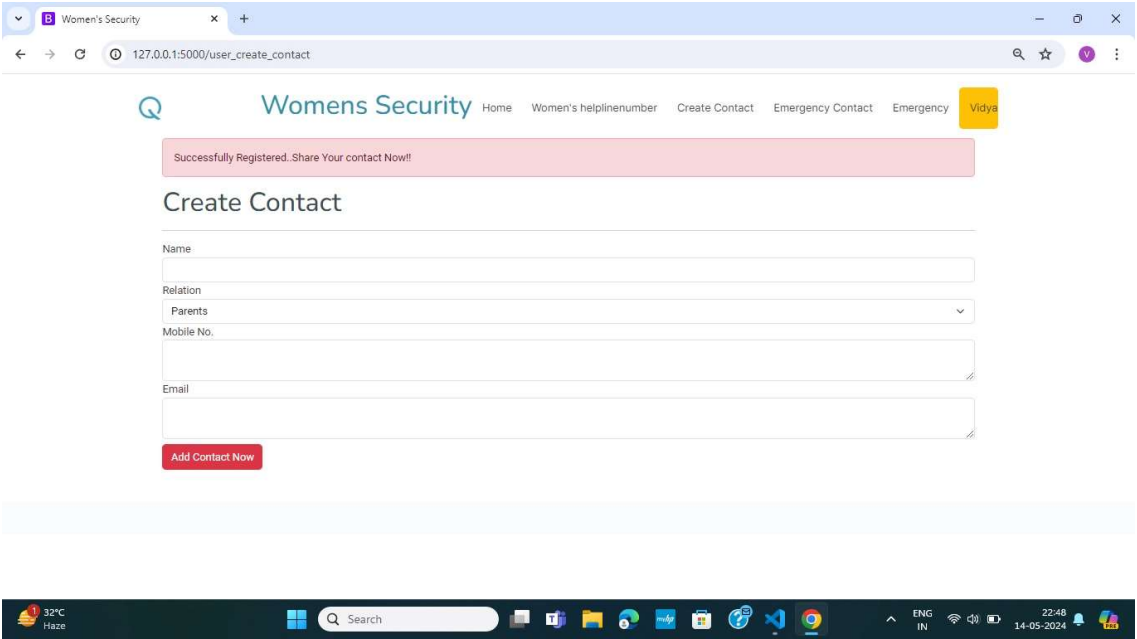
Relation
Parents

Mobile No.
9854632454

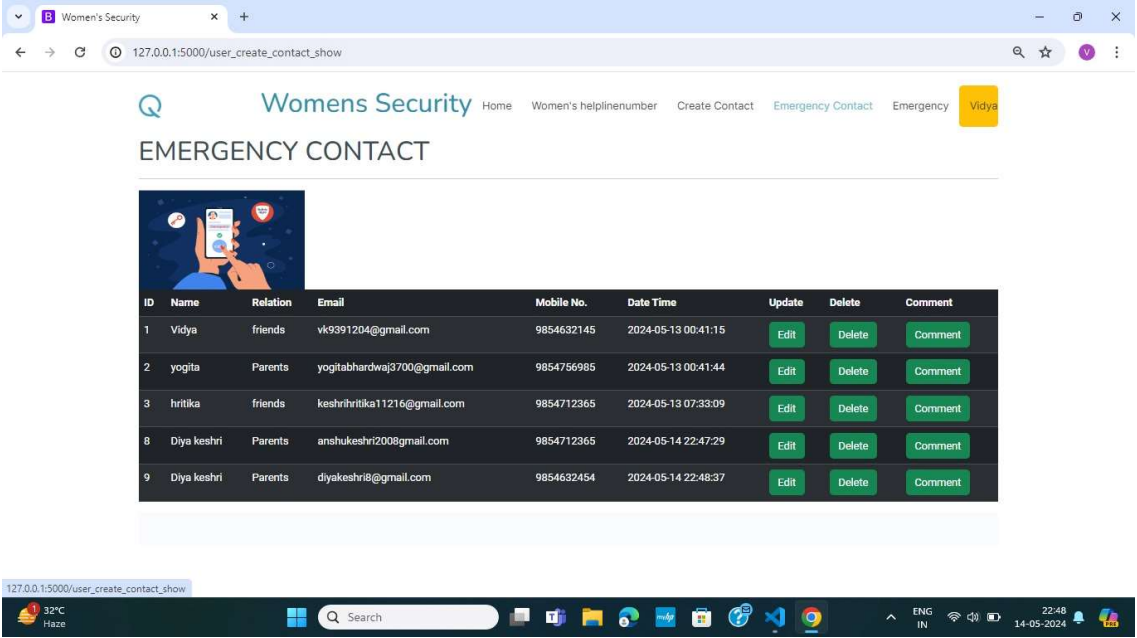
Email
diyakeshri8@gmail.com

Add Contact Now

11.



12.



13.

Women's Security

127.0.0.1:5000/user_create_contact_update?contact_id=1

Womens Security

HomeWomen's helpline numberCreate ContactEmergency ContactEmergency

Vidya

Update Emergency Contact

Name

Vidya

Relation

Parent

Email

vk9391204@gmail.com

Mobile No.

9854632147

Update Now

32°C
Haze

Search

ENG
IN

22:49
14-05-2024

14.

Women's Security

127.0.0.1:5000/user_create_contact_update?contact_id=1

Womens Security

HomeWomen's helpline numberCreate ContactEmergency ContactEmergency

Vidya

emergency contact is updated successfully!!

Update Emergency Contact

Name

Vidya

Relation

Parent

Email

vk9391204@gmail.com

Mobile No.

9854632147

Update Now

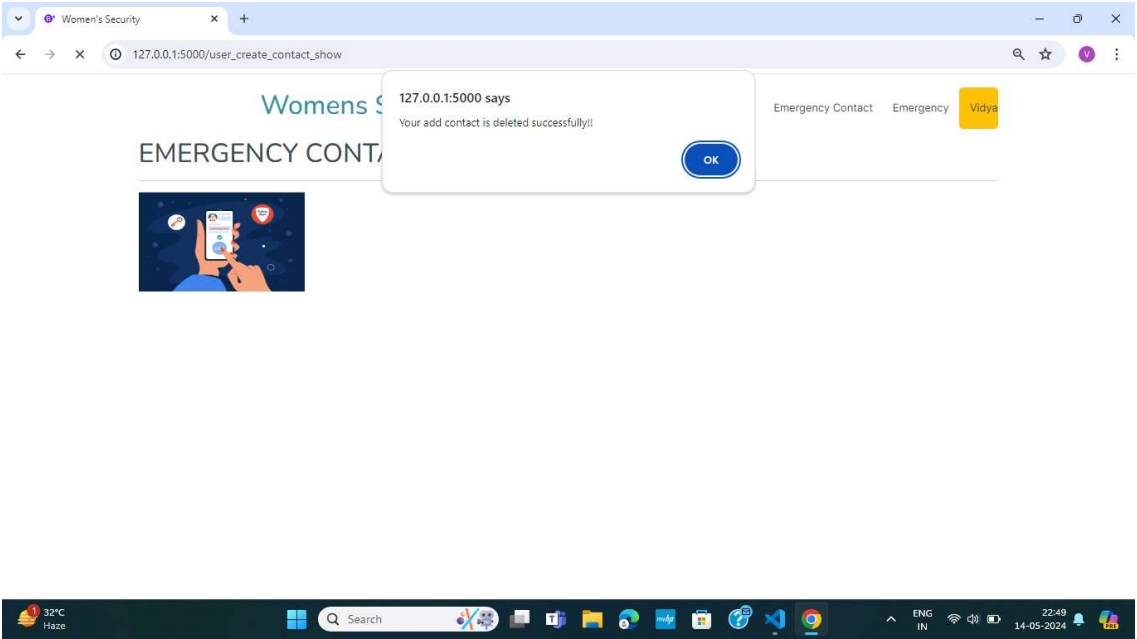
32°C
Haze

Search

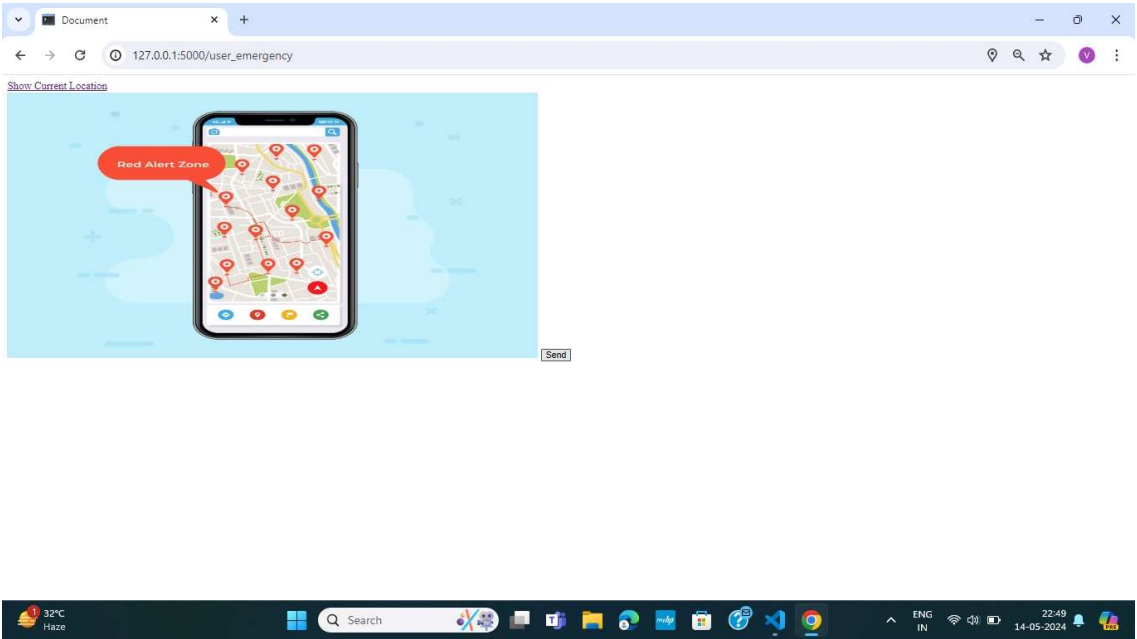
ENG
IN

22:49
14-05-2024

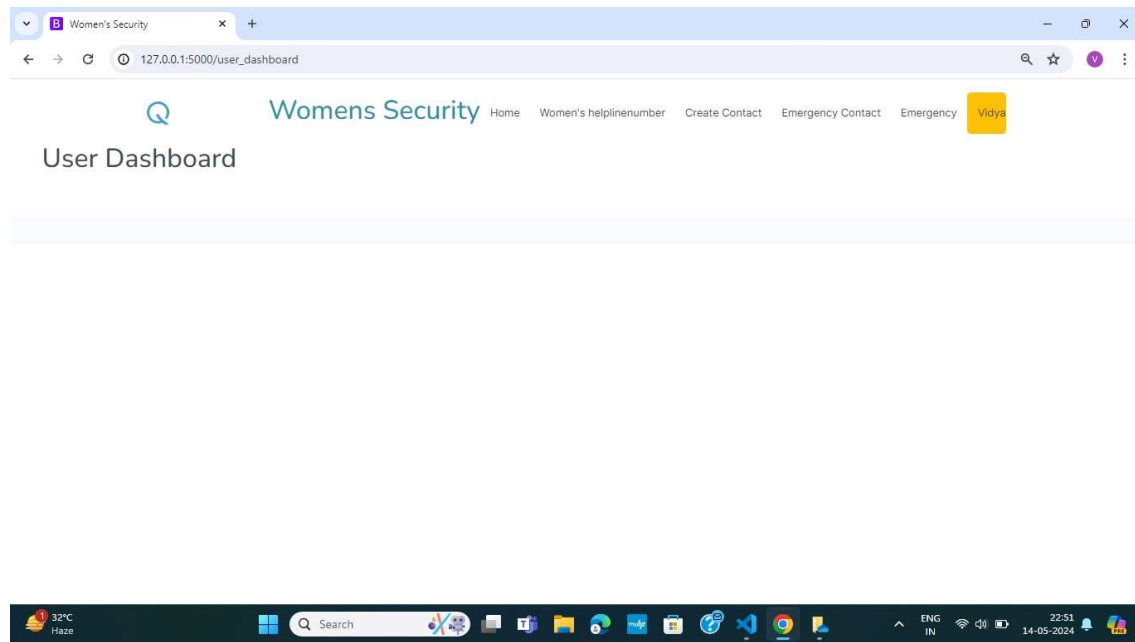
15.



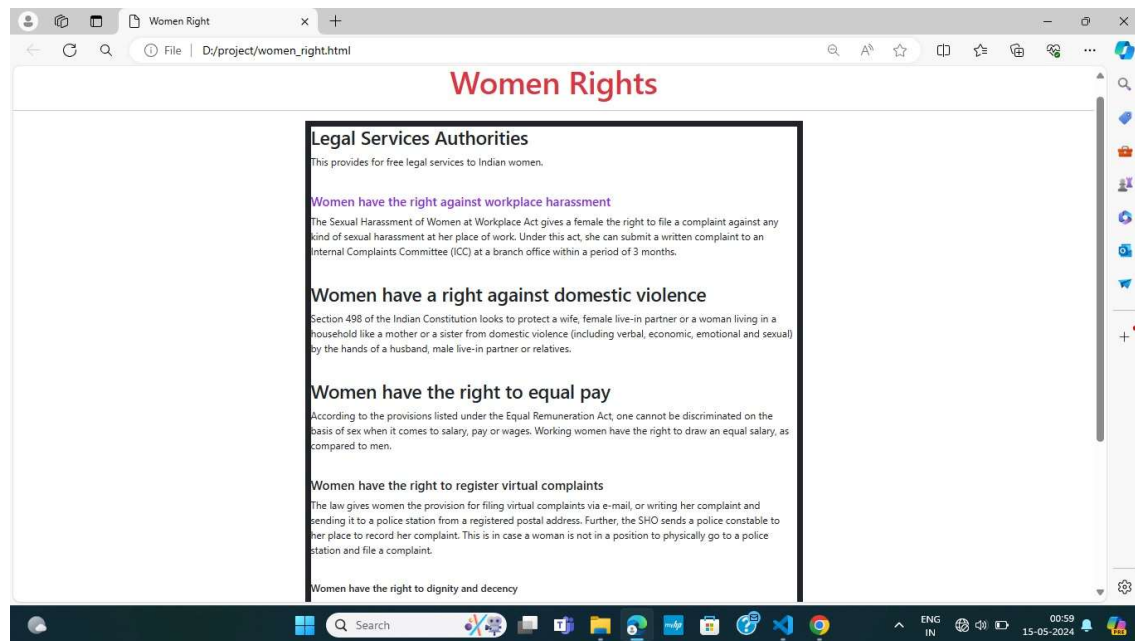
16.



17.



18.



9. FUTURE SCOPE

The future scope of women's safety apps is promising, with advancements in technology offering more sophisticated features. Here are some potential aspects:

1.Real-time Location Tracking:- Enhanced GPS technology for accurate location tracking can help in providing immediate assistance in case of emergencies.

2.AI-Powered Threat Detection:- Integration of artificial intelligence algorithms to analyze patterns and detect potential threats, triggering alerts or providing safety tips.

3.Emergency Alert Systems:- Improvements in emergency alert systems to notify trusted contacts, authorities, or nearby users in case of danger.

4.Crowdsourced Safety Networks:- Building communities within the app where users can support each other by reporting incidents, sharing safety tips, and offering help in emergencies.

5. Self-Defense Training:- Incorporating virtual self-defense training modules within the app to empower women with essential skills to protect themselves.

6. Integration with Wearable Devices:-Compatibility with wearable devices like smartwatches or smart jewelry for discreet distress signaling and tracking.

7. Anonymous Reporting:- Providing options for anonymous reporting of harassment or abuse, ensuring confidentiality and encouraging more women to speak up.

8. Legal Assistance and Resources:- Offering access to legal assistance, resources, and information on women's rights to help victims navigate the legal system.

9. Customizable Safety Plans:- Allowing users to create personalized safety plans based on their routines, preferences, and specific concerns.

10. Partnerships with Law Enforcement:- Collaborating with law enforcement agencies to streamline response procedures and ensure faster assistance during emergencies.

11. Continuous Improvement:-Regular updates and improvements based on user feedback and technological advancements to enhance effectiveness .

10. CONCLUSION

In conclusion, the future of women's safety apps holds significant promise in addressing the pressing issue of gender-based violence and ensuring the well-being of women globally. By incorporating advanced technologies, such as real-time location tracking, AI-powered threat detection, and emergency alert systems, these apps can offer immediate assistance and support in times of need.

Moreover, the integration of crowdsourced safety networks, self-defense training modules, and wearable device compatibility further enhances the effectiveness and accessibility of these apps. Additionally, features like anonymous reporting, legal assistance, and customizable safety plans contribute to creating a comprehensive support system for women facing threats to their safety.

Furthermore, partnerships with law enforcement agencies and continuous improvements based on user feedback ensure that these apps remain relevant and responsive to the evolving safety needs of women. Ultimately, women's safety apps have the potential to empower women, raise awareness about gender-based violence, and foster safer communities for all.

11. BIBLIOGRAPHY

WEBSITES:

1. www.w3school.com
2. www.google.co.in
3. javascriptkit.com
4. www.wikipedia.com
5. www.javatpoint.com
6. www.bootstrap.com

BOOKS:

- HTML Black Book
- HTML 5 for web Designers-Jeremy Kith
- Pro CSS & HTML Design patterns-Sergy Mavrody

