

**A  
PROJECT REPORT  
ON  
USER GENERATE PROFILE**

Submitted in partial fulfillment of the  
Requirement for the award of the  
**DEGREE  
OF  
MASTER OF COMPUTER APPLICATIONS**  
(2023-2025)



**Under the supervision of:  
Dr. Sukhvinder Singh Deora  
Assistant Professor  
DCSA MDU, Rohtak**

**Submitted by:  
Juhi  
MCA - 2<sup>nd</sup>Sem  
Roll No - 23170**

**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS  
MAHARSHI DAYANAND UNIVERSITY, ROHTAK- 124001**



DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS  
MAHARSHI DAYANAND UNIVERSITY ,ROHTAK

([www.mdurohtak.ac.in](http://www.mdurohtak.ac.in))

(NAAC Accredited 'A+' Grade Haryana State University)

**Dr.Preeti Gulia**  
**HEAD OF DEPARTMENT**

No. MDU/DCS/24/\_\_\_\_\_  
Date: \_\_\_\_\_

**CERTIFICATE-cum-DECLARATION**

I, Juhi, hereby declare that the work presented in the Project Report titled “**User Generate Profile**” and submitted as Industry Internship/Project-I as part of MCA 2nd Semester to Department of Computer Science & Applications, M. D. University, Rohtak for the partial fulfillment of the requirement of the award of degree of Master Of Computer Science & Applications is an authentic record of my work carried out during the MCA 2<sup>nd</sup> semester (Jan-June, 2024) under the supervision of **Dr.Sukhvinder Singh Deora, Asst. Professor**, Department of Computer Science & Applications.

Further, I also undertake that the matter embodied in this Project Report is my own work and has not been submitted by me or by any other candidate for the award of any other degree anywhere else.

**Student Name – Juhi**

Class:- MCA-2<sup>nd</sup> Semester

Roll\_No.-23170

**Countersigned by Internal Supervisor**

**Name: Dr. Sukhvinder Singh Deora**

Designation Asst.Professor, DCSA

Forwarded by:

**(Head of Department)**

## **ACKNOWLEDGEMENT**

Acknowledgement is not mere formality but a genuine opportunity to thank all those people without that active support in this project would not be able to be possible I am thankful to my guide “**Dr.Sukhvinder Singh Deora(Asst. Professor), DCSA, MDU, ROHTAK**” for his valuable time, exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by his time to time shall carry me a long way in the journey of life on which I am about to embark.

I would like to express my sincere gratitude to **Dr.Preeti Gulia**. Her golden words always gave me the inspiration to come out of my tough times, professionally as well as personally as well as personally. I also take this opportunity to express a deep sense of gratitude to All faculty members of **Department of Computer Science & Applications, M.D.U, Rohtak** for their cordial support, valuable information and guidance, which helped me in completing this task through various stages.

Finally, I also would like to thank my friends and family for always supporting me. I’m also grateful to everyone for their unwavering support in all of my endeavors.

Signature of the student

Juhi

MCA 2<sup>nd</sup> sem

Roll no. 23170

## **ABSTRACT**

This document is meant for describing all the features and procedures that were followed while developing the system.

This document specially mentions the detail of the project how it was developed, the primary required as well as various features and functionalities of the project and the procedure followed in achieving these objectives.

The **“USER GENERATE PROFILE”** consists of Graphical User Interface(GUI), and implemented using Sublime3.0(text editor).

This project of mine is being developed to help users/learners as well as beginners to increase their typing capability and enhanced their performance as well as reduce the human errors.

## TABLE OF CONTENTS

<b>Chapter</b>	<b>Page No.</b>
<b>1. Introduction to the project</b>	<b>7</b>
1.1 . Project Introduction	7
1.2 . Hardware configuration	8
1.3 . Software Configuration	8
<b>2. Software Requirement Specifications</b>	<b>9</b>
2.1. Introduction	9
2.2. Functional Requirement	10
2.3. Non- Functional Requirement	11
<b>3. Feasibility Analysis</b>	<b>12</b>
3.1. Technical Feasibility	12
3.2. Economic Feasibility	12
3.3. Operational Feasibility	13
<b>4. System Analysis</b>	<b>14</b>
4.1. System Requirements	14
4.2. Existing System	15
4.3. Proposed System	16
<b>5. System Design</b>	<b>17</b>
5.1 Database design	20
5.2 Data flow Diagram	22
5.3 E-R diagram	24
<b>6. System Implementation</b>	<b>25</b>
6.1 Coding	53
6.2 Snapshot	58

<b>Chapter</b>	<b>Page No.</b>
<b>7. Software Testing</b>	<b>59</b>
7.1 White box testing	59
7.2 Black box testing	60
7.3 Unit testing	61
7.4 Integration testing	61
7.5 Functional testing	62
7.6 System Testing	62
7.7 Acceptance testing	63
<b>8. Conclusion</b>	<b>64</b>
<b>Bibliography</b>	<b>65</b>

# **CHAPTER-1**

## **INTRODUCTION OF PROJECT**

### **1.1. PROJECT INTRODUCTION**

In the realm of user authentication and profile generation, leveraging typing speed as a biometric measure presents an innovative approach. This project aims to explore the feasibility and effectiveness of using typing speed as a unique identifier for user authentication and subsequently generating user profiles.

In the contemporary landscape of user authentication and personalized digital experiences, the project "User Profile Generation " seeks to innovate by leveraging typing speed as a unique identifier for user profiles.

This project integrates the conventional method of email and password authentication with an advanced behavioral biometric approach, focusing on the distinctive typing capabilities of individuals to create tailored user profiles.

The motivation behind this project arises from the growing need for enhanced user authentication methods that prioritize both security and user convenience. Traditional authentication systems reliant solely on email and password combinations often encounter security vulnerabilities such as password theft and phishing attacks.

It represents an individual's cognitive and neuromuscular patterns, inherently difficult to replicate or forge. By harnessing this distinct characteristic, the project aims to provide a secure and seamless authentication experience while generating meaningful user profiles for personalized services.

## **1.2. Minimum Hardware Configuration**

- Processor : Intel Core i3 or equivalent.
- RAM : 4GB minimum
- Hard Disk : 20GB
- Storage : 250GB HDD or 128GB SSD minimum.
- Monitor : 15.6" Color Monitor
- Screen Resolution : 1280x1024 or larger

## **1.3. Minimum Software Configuration**

- Operating System : Window 7 or above Version
- Languages : HTML, CSS, JAVASCRIPT, PHP
- Text Editor : Visual Studio Code



## **CHAPTER-2**

# **SOFTWARE REQUIREMENT SPECIFICATION**

## **4.1- Software Requirement Specification**

A Software requirements specification (SRS), a requirements specification for a software system, is a complete description of the behaviour of system to be developed and may include a set of use cases that describe interactions the users will have with software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design standards.

- **Introduction:-**

The Software Requirements Specification (SRS) outlines the detailed functional and non-functional requirements for the development of the "User Generate Profile " project. This document serves as a comprehensive guide for stakeholders, developers, and testers involved in the software development lifecycle.

- **Purpose:-**

The project aims to develop a system that automatically generates distinct user profiles based on users' typing capabilities and patterns. By analyzing typing speed, rhythm, accuracy, and other behavioral biometrics, the system will create personalized profiles for each user. This approach leverages unique typing characteristics as a form of behavioral authentication and a basis for tailoring digital experiences.

- **Scope:-**

The scope of this project includes implementing a secure and efficient user authentication system that incorporates typing speed analysis to generate personalized user profiles. The system will capture, analyze, and utilize typing speed data for user identification and profile creation, enhancing security and user experience.

## **4.2-Hardware and Software Specification**

### **Hardware Specification**

#### **Server Side**

- Processor : 1.0 GHz
- RAM : 512 MB
- Hard Disk : 20GB Free Space

#### **Client Side**

- Processor : 2.0 GHz
- RAM : 4 GB
- Hard Disk : 20GB Free Space

# Software Specification

## Server Side

- Operating System : Window Server 2008 or newer
- Front End : HTML,CSS,JAVASCRIPT
- Back End : PHP MYSQL
- Other Tools : Visual Studio, Notepad, MS Office

## Client Side

- Operating System : Window 10 newer/older

## **CHAPTER-3**

### **FEASIBILITY ANALYSIS**

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

#### **3.1 TECHNICAL FEASIBILITY**

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

The technical feasibility of "User Generate Profile" is promising, leveraging modern technologies and algorithms to capture, analyze, and utilize typing speed data for enhanced user authentication and personalized services. By addressing key considerations and challenges, the project can be implemented effectively to deliver secure, efficient, and user-centric experiences based on dynamic behavioral biometrics

#### **3.2 ECONOMICAL FEASIBILITY**

From an economic perspective, the project can be deemed feasible given the relatively low upfront costs associated with software development and integration of typing speed analysis algorithms. The primary investments would include

development resources for building the authentication system, frontend interface, and backend services, as well as ongoing maintenance and operational costs.

However, the potential benefits in terms of enhanced user experiences, improved security, and personalized service delivery can justify these investments.

Additionally, leveraging existing open-source libraries and cloud computing platforms can optimize costs while ensuring scalability and performance. Overall, the economical feasibility hinges on the project's ability to deliver tangible value through efficient use of resources and alignment with strategic business objectives.

### **3.3 OPERATION FEASIBILITY:**

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. The operation feasibility of "User Geerate Profile" is promising, with considerations for smooth integration, user adoption, maintenance, scalability, and security.

By addressing these operational aspects effectively, the project can be successfully implemented and integrated into existing operations, contributing to improved user experiences and enhanced security measures.

## **CHAPTER-4**

### **SYSTEM ANALYSIS**

#### **4.1 SYSTEM REQUIREMENTS**

This system can be used in Windows 98, Windows 2000, Windows XP and Windows NT, supported for other platform.

The system must be running at least on Windows 98, Windows 98 or Windows NT4.0 operating system and must meet the following hardware requirements.

- For Windows 95 based computers, a 486/66 MHz or higher processor with 8MB
- For Windows 98 based computers. a 500/88MHz or higher processor with 32 Mb of RAM
- For Windows NT based computers, a 488/66 MHz or higher processor with 16 MB of RAM
- For Windows 200 based computers, a 700/850 MHz or higher processor with 512 MB of RAM

## 4.2 EXISTING SYSTEM

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

During analysis, data collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram, interviews, etc. Training, experience and common sense are required for collection of relevant information needed to develop the system.

The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs.

- System analysis can be categorized into four parts.
- System planning and initial investigation
- Information Gathering
- Applying analysis tools for structured analysis
- Feasibility study
- Cost/Benefit analysis.

Our previous user profile generation system, while functional, had several notable limitations that hindered its effectiveness and user experience. One of the primary issues with the legacy system was its reliance on static user inputs and the absence of dynamic behavioral biometrics, such as typing speed, to create personalized profiles.

The older system operated on conventional authentication methods like email and password combinations, which, although widely used, lacked the sophistication needed to deliver truly personalized experiences. User profiles were primarily generated based on static information provided during registration, resulting in generic profiles that did not adequately reflect individual user behaviors or preferences.

### **4.3 PROPOSED SYSTEM**

The project aims to create a secure and personalized user generate profile system based on typing speed analysis. It involves integrating traditional email/password authentication with behavioral biometrics (typing speed) to generate unique user profiles for personalized digital experiences.

- User Authentication
- Typing Speed Analysis
- User Profile Generation
- User Dashboard

The system analysis of "User Profile Generation Based on Typing Speed" outlines the functional and non-functional requirements, system components, data flow, and architecture necessary for its successful implementation.

By addressing these aspects comprehensively, the project can deliver a secure, efficient, and user-centric authentication system integrated with personalized user profiles based on typing speed characteristics.



## **CHAPTER-5**

### **SYSTEM DESIGN**

#### **INPUT DESIGN**

Input design is the process of converting user-oriented input to a computer based format. Input design is a part of overall system design, which requires very careful attention. Often the collection of input data is the most expensive part of the system. The main objectives of the input design are

1. Produce cost effective method of input
2. Achieve highest possible level of accuracy
3. Ensure that the input is acceptable to and understood by the staff.

#### **Input Data**

The goal of designing input data is to make enter easy, logical and free from errors as possible. The entering data entry operators need to know the allocated space for each field; field sequence and which must match with that in the source document. The format in which the data fields are entered should be given in the input form. Here data entry is online; it makes use of processor that accepts commands and data from the operator through a key board. The input required is analyzed by the processor. It is then accepted or rejected. Input stages include the following processes.

- Data Recording
- Data Transcription
- Data Conversion

- Data Verification
- Data Control
- Data Transmission
- Data Correction

One of the aims of the system analyst must be to select data capture method and devices, which reduce the number of stages so as to reduce both the changes of errors and the cost. Input types, can be characterized as.

- External
- Internal
- Operational
- Computerized
- Interactive

Input files can exist in document form before being input to the computer. Input design is rather complex since it involves procedures for capturing data as well as inputting it to the computer.

## **OUTPUT DESIGN**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these result for latter consultation. Computer output is the most important and direct source of information to the users. Designing computer output should proceed in an organized well throughout the manner. The right output must be available for the people who find the system easy to use. The outputs have been defined during the logical design stage. If not, they should have defined at the beginning of the output designing terms of types of output connect, format, response etc.

Various types of outputs are

- External outputs
- Internal outputs
- Operational outputs
- Interactive outputs
- Turn around outputs

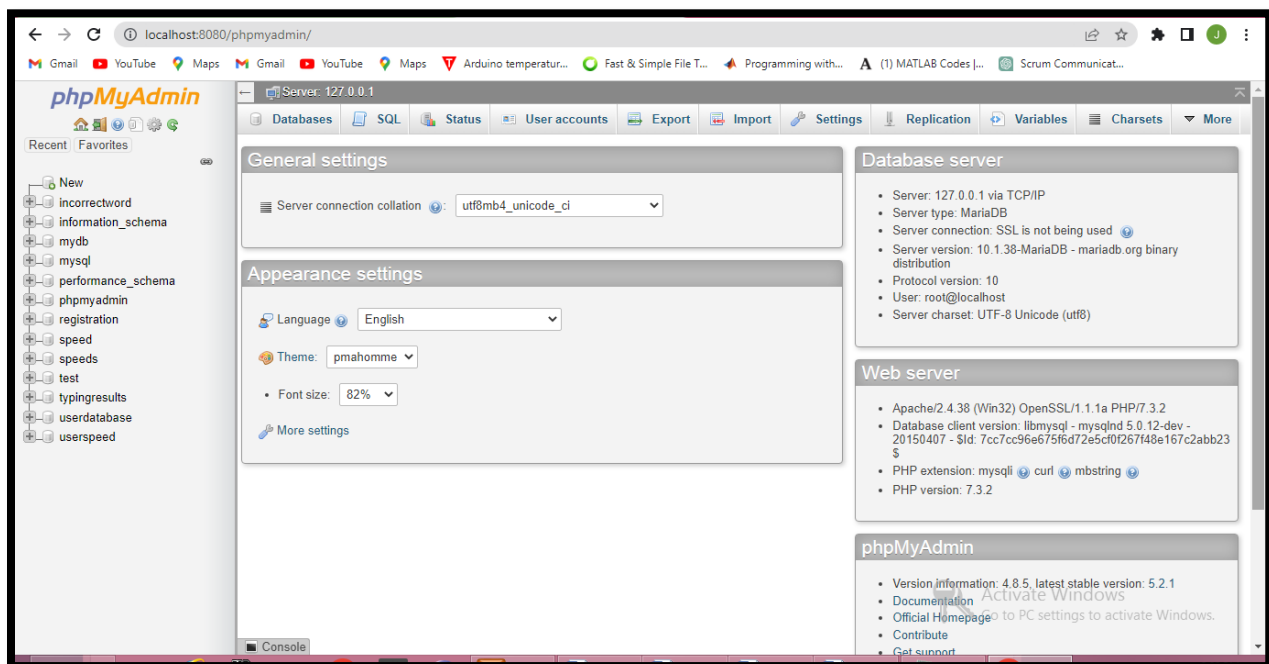
All screens are informative and interactive in such a way that the user can full fill his requirements through asking queries.

## **5.1-DATABASE DESIGN**

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy quick, inexpensive and flexible for other users. During database design the following objectives are concerned:-

- Controlled Redundancy
- Data independence
- Accurate and integrating
- More information at low cost
- Recovery from failure
- Privacy and security
- Performance
- Ease of learning and use

The database is essential for your project as it serves as the central repository for user-related data, typing speed metrics, and generated profiles. It enables secure authentication, personalized services, data analysis, and compliance with regulatory requirements. By leveraging the database effectively, your project can deliver a robust and user-centric solution for user profile generation based on typing capabilities while ensuring data security, integrity, and scalability.

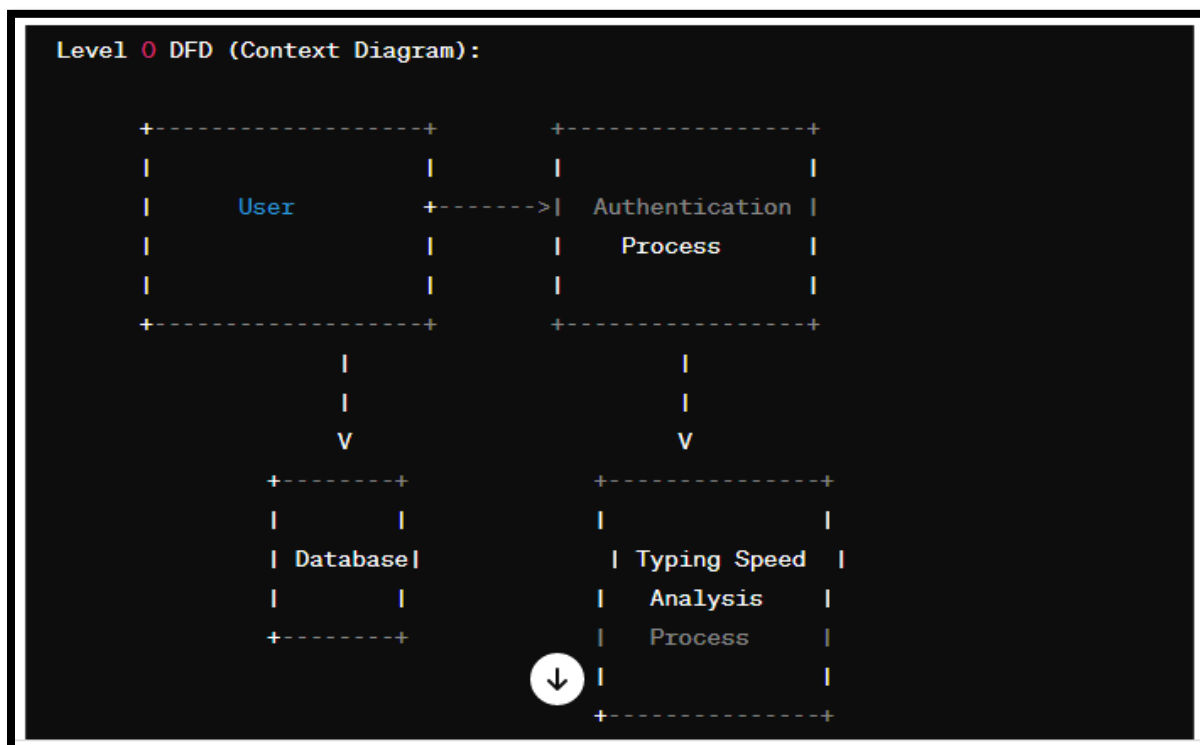


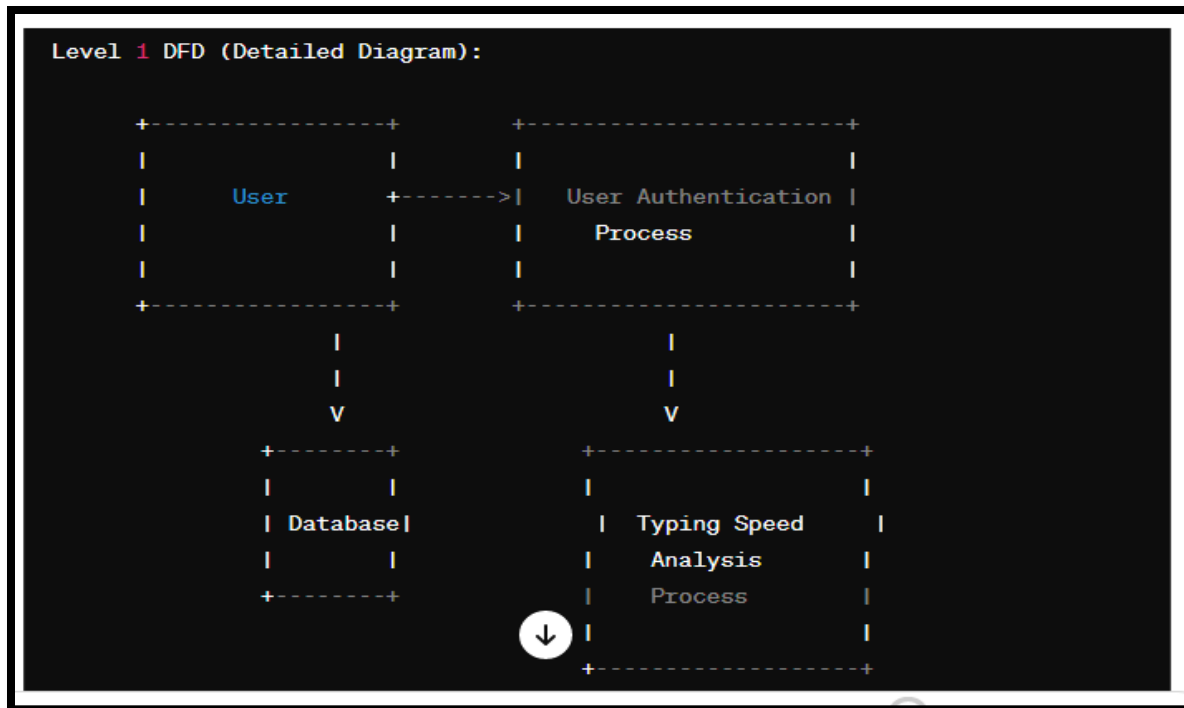
## PHP DATABASE

## 5.2-DFD(Data Flow Diagram)

DFD illustrates the flow of data and processes in your project, emphasizing user authentication, typing speed analysis, profile generation, and data storage in the database.

It provides a structured view of how the system components interact and how data moves through the system to achieve the goal of generating user profiles based on typing capabilities. Customize this DFD further based on specific processes, interactions, and data flows relevant to your project's requirements and functionalities.





**In the above representation:**

- User: Represents the external entity (user) interacting with the system.
- Authentication Process: Validates user credentials and captures typing speed data.
- Typing Speed Analysis Process: Analyzes typing speed data to extract behavioral biometrics.
- Database: Stores user data, typing speed metrics, and generated profiles.

## 5.3-E-R DIAGRAM

ER diagram illustrates the structure of the database schema for your project, highlighting the entities (User, TypingSession, UserProfile) and their attributes. The relationships (one-to-many and one-to-one) between entities define how data is organized and linked within the database.

Customize this ER diagram based on specific requirements and additional attributes relevant to your project's functionalities and data storage needs. You can use software tools like Microsoft Visio, Lucidchart, or draw.io to create a visual representation of this ER diagram for better understanding and documentation of your database design.





## **CHAPTER-6**

### **SYSTEM IMPLEMENTATION**

#### **Registration.html:-**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Registration</title>
  <link rel="stylesheet" href="Registration.css">
  <script>
function showAlertAndRedirect(event) {
  event.preventDefault(); // Prevent default form submission

  // Get form inputs
  var fullNameInput = document.querySelector('input[name="fullname"]');
  var emailInput = document.querySelector('input[name="email"]');
  var passwordInput = document.querySelector('input[name="password"]');

  // Validate form inputs
  if (fullNameInput.value.trim() === "" || emailInput.value.trim() === "" ||
      passwordInput.value.trim() === "") {
    alert("Please fill in all the fields.");
    return;
  }

  // Validate password requirements
  var passwordValue = passwordInput.value.trim();
  if (passwordValue.length < 8 || !containsUpperCase(passwordValue) ||
      !containsLowerCase(passwordValue)) {
```

```

        alert("Password must be at least 8 characters long and contain both uppercase
and
        lowercase letters.");
        return;
    }

    // Check if the email is already registered (Client-side validation)
    var registeredEmails = JSON.parse(localStorage.getItem('registeredEmails')) ||
[];
    if (registeredEmails.includes(emailInput.value.trim())) {
        alert("Email already registered. Please use a different email address.");
        return;
    }

    // Store the email in local storage to simulate registered emails (For
demonstration)
    registeredEmails.push(emailInput.value.trim());
    localStorage.setItem('registeredEmails', JSON.stringify(registeredEmails));

    // Registration success message and redirect to login page
    alert("Registration successful! Now login into your account.");
    window.location.href = "login.html";
}

// Helper function to check if string contains at least one uppercase letter
function containsUpperCase(str) {
    return /[A-Z]/.test(str);
}

// Helper function to check if string contains at least one lowercase letter
function containsLowerCase(str) {
    return /[a-z]/.test(str);
}
</script>
</head>
<body>
    <div class="container">

```

```

<h1>User Registration</h1>
<form id="registrationForm" action="Registration.php" method="post">
  <input type="text" name="fullname" placeholder="Full Name" required>
  <input type="email" name="email" placeholder="Email" required>
  <input type="password" name="password" placeholder="Password"
required>
  <button type="submit" >Register</button>
  <p>Already registered? <a href="login.html">Login</a></p>
</form>
</div>
</body>
</html>

```

## **Registration.css:-**

```

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;

  background-image: url("images.jpeg");
  background-repeat: no-repeat;
  background-size: cover;
  width: 100%;
  height: 100%;
}

.container {
  max-width: 400px;
  margin: 100px auto;
  background-color: #fff;
  padding: 20px;
  border-radius: 8px solid black;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  float: left;
  margin-left: 12%;

```

```

}

h1 {
    text-align: center;
    margin-bottom: 20px;
}

input[type="text"],
input[type="email"],
input[type="password"],
button {
    width: 95%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    background-color: blue;
    color: #fff;
    cursor: pointer;
    border-radius: 20px;
    width: 60%;
    margin-left: 20%;
}

button:hover {
    background-color: brown;
}

.hidden {
    display: none;
}

#successMessage {
    text-align: center;

```

```

background-color: #28a745;
color: #fff;
padding: 10px;
border-radius: 5px;
margin-bottom: 15px;
}
p{
font-family: Arial, sans-serif;
margin-left: 30%;
margin-top: 0%;
}
a{
text-decoration:none;
color:blue;
}

p{
font-family: Arial, sans-serif;
}

```

### **Registration.php:-**

```

<?php
// Check if form is submitted via POST method
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve and sanitize form data
    $fullname = isset($_POST['fullname']) ? htmlspecialchars($_POST['fullname'])
: "";
    $email = isset($_POST['email']) ? htmlspecialchars($_POST['email']) : "";
    $password = isset($_POST['password']) ? $_POST['password'] : "";

    // Validate form inputs
    if (empty($fullname) || empty($email) || empty($password)) {
        die("Please fill in all the fields.");
        header("Location: registration.html");
    }
}

```

```

// Validate email format
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    die("Invalid email format.");
    header("Location: registration.html");
}

// Validate password requirements (at least 8 characters with uppercase,
    lowercase, and numeric characters)

if (strlen($password) < 8 || !preg_match("#[A-Z]+#", $password) ||
    !preg_match("#[a-z]+#", $password) || !preg_match("#[0-9]+#", $password))
{
    die("Password must be at least 8 characters long and contain uppercase,
        lowercase, and numeric characters.");
    header("Location: registration.html");
}

// Database connection parameters
$servername = "localhost";
$username = "root";
$dbpassword = ""; // Replace with your database password
$dbname = "userdatabase"; // Replace with your desired database name

// Create connection
$conn = new mysqli($servername, $username, $dbpassword, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Hash the password for security
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);

// Prepare and execute SQL statement to insert data into the 'users' table
$sqlInsertUser = "INSERT INTO users (fullname, email, password) VALUES
    (?, ?, ?)";

```

```

$stmtInsert = $conn->prepare($sqlInsertUser);
$stmtInsert->bind_param("sss", $fullname, $email, $hashedPassword);

if ($stmtInsert->execute()) {
    // Registration successful
    echo "Registration successful! Now login into your account.";

    // Close prepared statement and database connection
    $stmtInsert->close();
    $conn->close();

    // Use JavaScript to show alert and redirect
    echo '<script>alert("Registration successful! Now login into your account.");
    window.location.href = "login.html";</script>';
    exit; // Stop further PHP execution

}
else {
    // Error in SQL query
    echo "Error: " . $stmtInsert->error;
}

// Close prepared statement and database connection
$stmtInsert->close();
$conn->close();
}
else {
    // Redirect to the registration page if accessed directly without form submission
    header("Location: registration.html");
    exit;
}
?>

```

## **Login.html:-**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Login</title>
  <link rel="stylesheet" href="login.css">
</head>
<body>
  <h2><center>User Login</center></h2>
  <form action="login-process.php" method="post">
    <div>
      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required>
    </div>
    <div>
      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
    </div>
    <button type="submit">Login</button>
  </form>
</body>
</html>
```

## **Login.css:-**

```
body {
  font-family: Arial, sans-serif;
  background: linear-gradient(45deg, rgba(245,70,66,0.75), rgba(8,80,136,0.75));
  background-size: cover;
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```



```

form {
    font-size: 18px;
    width: 30%;
    margin: 50px auto;
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

label {
    display: block;
    margin-bottom: 10px;
}
h2{
    color:brown;
    font-size: 34px;
    margin-bottom: 5px;
}
input[type="email"], input[type="password"] {
    width: 100%;
    padding: 10px;
    margin-bottom: 20px;
    border: 1px solid #ccc;
}

button {
    background-color: #3e8e41;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 10px;
    cursor: pointer;
}

button:hover {
    background-color: red ;
}

```

## **Login-Process.php:-**

```
<?php
// Database connection parameters
$servername = "localhost";
$username = "root";
$password = ""; // Replace with your database password
$dbname = "userdatabase"; // Replace with your desired database name

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if form is submitted via POST method
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve form data
    $email = isset($_POST['email']) ? $_POST['email'] : "";
    $password = isset($_POST['password']) ? $_POST['password'] : "";

    // Validate email and password
    if (empty($email) || empty($password)) {
        die("Please enter both email and password.");
    }

    // Fetch user record from database based on email
    $sql = "SELECT * FROM users WHERE email = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        // User found, verify password
        $row = $result->fetch_assoc();
        $storedPassword = $row['password'];

        // Verify password
        if ($password === $storedPassword) {
```

```

        // Password is correct, login successful
        session_start();
        $_SESSION['user_id'] = $row['id']; // Store user ID in session (if needed)
        echo '<script>alert("Login successful! Now proceed to typing."); window.location.href =
        "typing.html";</script>';
        exit;
    }
    else {
        // Incorrect password
        echo "Incorrect email or password. Please try again.";
    }
}
else {
    // User not found
    echo "User with this email does not exist.";
}

$stmt->close();
}
else {
    // Redirect to login page if accessed directly
    header("Location: login.html");
    exit;
}

$conn->close();
?>

```

## **TYPING.HTML:-**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Type Writer</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div><br>
    <h1><center> Generate User Profile </center></h1>
    <div class="container">
        <div class="typing-container">

```



## **TYPING.CSS:-**

```
body {
  font-family: Arial, sans-serif;
  background: radial-gradient(circle, #4facfe, #00f2fe);
  background: linear-gradient(135deg, #f6d365 0%, #fda085 100%),
    radial-gradient(circle at top left, #f6d365 0%, #fda085 100%);
  background-blend-mode: screen;

  color: white;
  text-align: center;

}

.container {
  width: 80%;
  height: 70%;
  margin: 10px auto;
  padding: 10px;
  border: 1px solid #ccc;
  background-color: #ccda46;

  border-radius: 10px;
  box-shadow: 1 1 10px rgba(151, 62, 62, 0.1);

}

textarea {
  padding: 50px;
  width: 80%;
  text-align: center;
  height: 60px;
  box-shadow: black 6px 8px 6px;
  margin-left: 50px;
  font-size: 22px;
  margin-top: 20px;

}
```

```
#timer,#correctWordCount,#incorrectWordCount {
  background-color: grey;
  height: 180px;
  width: 170px;
  margin-left: 13%;
  padding: 12px;
  box-shadow: black 5px 8px 5px;
  font-size: 30px;
  margin-bottom: 10px;
  float:left;
  border-radius: 5px solid red;
  margin-top: 10px;
}
```

```
button {
  padding: 10px 20px;
  background-color:blue;
  color: #fff;
  margin-left: 5%;
  border: none;
  cursor: pointer;
  display: inline;
  align-items: center;
  font-size:15px;
  margin-top: 10px;
  border-radius: 10px;
}
```

```
#start-btn{
display:inline;
align-items: center;
margin-left:8%;
}
```

```
button:hover {
  background-color: red;
}
```

```
#text-to-type{
  background-color:transparent;
  margin-left: 70px;
  text-align: justify;
```

```

color: black;
font-size: 22px;
height: 160px;
width: 88%;
word-spacing: 5px;
padding: 5px;
margin-top: 30px;
margin-bottom: 40px;
line-height: 39.5px;
box-shadow: rgb(61, 20, 20) 6px 8px 6px;
}
#hr{
  color: rgba(8, 117, 136, 0.521);
}

```

```

#results{
  color: #ff5500;
  font-family: 'cursive';
  font-size: 24px;
  background-color: #2d2e27;
  box-shadow: black 2px 3px 5px;
  margin-bottom: 10px;
  border-radius: 8px solid red;
  display: inline;
  overflow: hidden;
}

```

```

.correct-word {
  padding: 3px;
  border-radius: 5px ;
  border: 3px solid green;
  margin-top: 2px;
  margin-bottom: 2px;
  border-collapse: collapse;
}
.incorrect-word {
  border-radius: 5px ;
  border: 3px solid red;
  padding: 3px;
  margin-top: 2px;
}

```

```
margin-bottom: 2px;

}
h1{
  font-style: bold;

  color: brown;
}
```

```
#resultsContainer {
  margin-top: 20px;
  padding: 10px;
  width: 75%;
  text-align: left;
  font-size:20px;
  color:white;
  margin-left:12%;
}
```

```
#incorrectWords{
  float:left;
  font-size:0px;

}
```

```
#correctSpelling{
  float:right;
  font-size:0px;
}
```

```
#errorC{
height: 5px;
width: 2px;
margin-left:6%;
padding: 10px;
font-size: 0px;
}
```

```
#time{
  height: 180px;
```



```

width: 170px;
margin-left:6%;
padding: 12px;
font-size: 70px;

}

#spani{
height: 180px;
width: 170px;
margin-left:6%;
padding: 12px;
font-size: 70px;

}

#span{
height: 180px;
width: 170px;
margin-left:6%;
padding: 12px;
font-size: 70px;

}

#correct-words-container{
font-size:10px;
}
#incorrect-words-container{
font-size:10px;
}

```

## **TYPING.JS:-**

```

var textToType = document.getElementById('text-to-type').innerText.trim();
var typedText = document.getElementById('typing-area');
const generateProfileBtn = document.getElementById("GenerateProfile");
const errorFeedback = document.getElementById("errorCount");
var timerInterval;
var startTime;
var CorrectWords=0;

```

```

// Function to start the timer and word speed calculation

function startTimer() {
  startTime = new Date().getTime();
  timerInterval = setInterval(updateTimer, 1000);
}

// Function to update the timer display
function updateTimer() {
  var currentTime = new Date().getTime();
  var elapsedTime = currentTime - startTime;
  var minutes = Math.floor(elapsedTime / (1000 * 60));
  var seconds = Math.floor((elapsedTime % (1000 * 60)) / 1000);
  document.getElementById('time').innerText = minutes + ':' + (seconds < 10 ? '0' : '') + seconds;
}

// Function to stop the timer, calculate word speeds, and check correctness of words
function stopTimer() {
  clearInterval(timerInterval);

  const paragraph = document.getElementById('text-to-type').innerText.trim();
  const userInput = document.getElementById('typing-area').value.trim();
  const correctWords = paragraph.split(' ');
  const userWords = userInput.split(' ');

  const incorrectWords = {};
  const correctWordsList = {};

  let previousWordEndTime = startTime;

  // Calculate speed and check correctness for each word
  for (let i = 0; i < correctWords.length; i++) {
    const correctWord = correctWords[i];
    const userWord = userWords[i] || ""; // Handle case where user input is shorter

```

```

    const currentWordEndTime = previousWordEndTime + correctWord.length * 1000; //
    Estimate end time based on correct word length
    const wordTimeElapsed = currentWordEndTime - previousWordEndTime;

    const wordSpeed = parseFloat((60 / wordTimeElapsed) * 1000); // Speed in words per
    minute

    // Determine if the word is correct or incorrect
    if (userWord === correctWord) {
        correctWordsList[correctWord] = parseFloat(wordSpeed.toFixed(2));
    } else {
        // Store the incorrect word with its speed and correct spelling
        incorrectWords[userWord] = {
            speed: parseFloat(wordSpeed.toFixed(2)),
            correctSpelling: correctWord
        };
    }

    // Update the end time of the previous word to the end time of the current word
    previousWordEndTime = currentWordEndTime;
}

// Prepare data to send to PHP
const data = {
    incorrectWords: incorrectWords,
    correctWords: correctWordsList
};

// Send data to PHP using Fetch API
fetch('save.php', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
})
.then(response => {
    if (response.ok) {
        console.log('Data sent successfully');
    } else {
        throw new Error('Error sending data');
    }
})

```

```

    .catch(error => {
      console.error('Request failed:', error);
    });
  }
}

```

```

// Prevent backspace key from navigating back in the typing area
const typingArea = document.getElementById('typing-area');
typingArea.addEventListener('keydown', function(event) {
  if (event.key === 'Backspace') {
    event.preventDefault();
  }
});

```

```

//resetall
function resetall(){
  clearInterval(timerInterval);
  document.getElementById('timer').innerText = 'Time: ';
  document.getElementById('typing-area').disabled = false;
  document.getElementById('span').innerText = "";
  document.getElementById('spani').innerText = "";
  document.getElementById('typing-area').value = "";
  document.getElementById('start-btn').style.display = 'inline';
  document.getElementById('submit-btn').style.display = 'none';
  document.getElementById('reset').style.display = 'inline';
  document.getElementById('results').innerHTML = "";
  document.getElementById('text-to-type').innerHTML = ' This is a simple paragraph that is
meant to be nice and easy to type which is why there will be mommas no periods or any capital
letters This is a simple paragraph that is meant to be nice and easy.';
}

```

```

//highlight and error count

```

```

function checkTyping() {
  let incorrectWords=[];
  const correctText = document.getElementById("text-to-type").innerText;
  const userInput = document.getElementById("typing-area").value;
  const correctTextWords = correctText.split(" ");

```

```

const userInputWords = userInput.split(" ");
let errorCount = 0;
let displayText = "";

for (let i = 0; i < correctTextWords.length; i++) {
  if (i < userInputWords.length) {
    const correctWord = correctTextWords[i];
    const userWord = userInputWords[i];
    const minLength = Math.min(correctWord.length, userWord.length);
    let wordDisplay = "";

    for (let j = 0; j < minLength; j++) {
      if (correctWord[j] === userWord[j]) {
        wordDisplay += `<span style="color: green;">${correctWord[j]}</span>`;
      } else {
        wordDisplay += `<span style="color: red;">${correctWord[j]}</span>`;
        errorCount++;
      }
    }

    if (correctWord.length > userWord.length) {
      wordDisplay += correctWord.slice(minLength);
    }

    if (correctWord !== userWord) {
      wordDisplay = `<span class="incorrect-word">${wordDisplay}</span>`;
      incorrectWords.push(correctWord);
    } else {
      wordDisplay = `<span class="correct-word">${wordDisplay}</span>`;
    }

    displayText += wordDisplay + " ";
  } else {
    displayText += `<span style="color: black;">${correctTextWords[i]}</span>` + " ";
  }
}

document.getElementById("text-to-type").innerHTML = displayText;
document.getElementById('errorC').innerText = errorCount;
countIncorrectWords();

// Count correct words
//countCorrectWords(correctTextWords, userInputWords);

```

```

}

/*function countCorrectWords(correctWords, userWords) {
  let correctWordCount = 0;

  for (let i = 0; i < correctWords.length; i++) {
    if (i < userWords.length && correctWords[i] === userWords[i]) {
      correctWordCount++;
    }
  }

  document.getElementById('span').innerText = correctWordCount;
}
*/

// Example HTML for displaying correct word count

function countIncorrectWords() {
  const incorrectWords = document.getElementsByClassName("incorrect-word");
  let incorrectWordCount = 0;
  for (let i = 0; i < incorrectWords.length; i++) {
    incorrectWordCount++;
  }
  document.getElementById('spani').innerText=incorrectWordCount;
}
document.getElementById("typing-area").addEventListener("input", checkTyping);

// calculate typing speed
function calculateTypingSpeed(start, end) {
  const elapsedTimeInSeconds = (end - start) / 1000;
  const wordCount = typedText.value.split(" ").length;
  const typingSpeed = Math.round(wordCount / elapsedTimeInSeconds * 60);
  return typingSpeed;
}

//result
function result() {
  const userInput = document.getElementById("typing-area").value;

```

```

const userInputWords = userInput.split(" ");
const wordCount = userInputWords.length;
const endTime = new Date();
const totalTime = (endTime - startTime) / 1000; // in seconds
const characters= document.getElementById("errorC").innerText;

const incorrectWordCountElement = document.getElementById("incorrectWordCount");
const incorrectWordCountString = incorrectWordCountElement.innerText;
const numericPart = incorrectWordCountString.match(/\d+/);
const incorrectWordCount = parseInt(numericPart);

const correctWords = wordCount - incorrectWordCount;
const accuracy = (correctWords / wordCount) * 100;
const typingSpeed = (correctWords / (endTime - startTime)) * 60;
const timeDifference = endTime - startTime;
const timeInMinutes = timeDifference / (1000 * 60);
const wpm = (wordCount / timeInMinutes).toFixed(2);

// Prepare data object to be sent as JSON
const data = {
  wordCount: wordCount,
  characters: characters,
  wpm: wpm,
  accuracy: accuracy,
  typingSpeed: typingSpeed,
  averageTimePerWord: (totalTime / wordCount).toFixed(2)
};

// Define fetch options for POST request
const url = 'store_results.php';
const options = {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(data)
};

// Send POST request using Fetch API
fetch(url, options)
  .then(response => {
    if (response.ok) {

```

```

        console.log('Data sent successfully');
        // Optionally, you can handle the response from PHP here
        return response.json(); // Parse response JSON if needed
    } else {
        throw new Error('Error sending data');
    }
})
.then(data => {
    // Handle response data if needed
    console.log(data);
})
.catch(error => {
    console.error('Request failed:', error);
});

// Display results in HTML (optional)
const results = `Typing Results:- <br><br>
Total ${document.getElementById('timer').innerText} seconds <br>
Accuracy level: ${accuracy}% <br>
Word Count: ${wordCount} words <br>
Words per minute: ${wpm} <br>
Wrong characters: ${characters} characters <br>
Average Time: ${((totalTime / wordCount).toFixed(2))} seconds per word`;
document.getElementById('results').innerHTML = results;
document.getElementById('span').innerText = correctWords;
}

function generateResults(){
    window.location.href="display_data.php";
}

//start btn
document.getElementById('start-btn').addEventListener('click', function() {
    alert("now you can start typing!!!");
    startTimer();
    checkTyping();
    document.getElementById('start-btn').style.display = 'none';
    document.getElementById('submit-btn').style.display = 'inline';

});
//submit btn

```



```

document.getElementById('submit-btn').addEventListener('click', function() {
stopTimer();
result();
document.getElementById('typing-area').disabled = true;
document.getElementById('submit-btn').disabled = true;
});

```

## **Display.php:-**

```

<?php
session_start();

// Database connection parameters
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "userSpeed";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Get the current hour and minute in Indian Standard Time (IST)
$currentHour = date('H'); // Get current hour (24-hour format)
$currentMinute = date('i'); // Get current minute

// Query to count correct words for the current time
$sql_correct_count = "SELECT COUNT(*) AS correct_count FROM correct_words
    WHERE HOUR(created_at) = $currentHour AND MINUTE(created_at) =
$currentMinute";
$result_correct_count = $conn->query($sql_correct_count);
$correct_count = ($result_correct_count->num_rows > 0) ? $result_correct_count-
>fetch_assoc()['correct_count'] : 0;

// Query to count incorrect words for the current time
$sql_incorrect_count = "SELECT COUNT(*) AS incorrect_count FROM incorrect_words

```

```

WHERE HOUR(created_at) = $currentHour AND MINUTE(created_at) = $currentMinute";
$result_incorrect_count = $conn->query($sql_incorrect_count);
$incorrect_count = ($result_incorrect_count->num_rows > 0) ? $result_incorrect_count-
>fetch_assoc()['incorrect_count'] : 0;

// Determine conclusion based on counts
if ($correct_count > $incorrect_count) {
    $conclusion = "Good luck! You're typing well.";
} else {
    $conclusion = "Better luck next time. Please type carefully and don't type so fastly!!!.";
}

// Store user details and conclusion in session
$_SESSION['user_name'] = 'Juhi'; // Replace with actual user name
$_SESSION['user_email'] = 'juhi461251@example.com'; // Replace with actual user email
$_SESSION['conclusion'] = $conclusion;

// Close connection
$conn->close();

// Redirect to profile.php
header("Location: profile.php");
exit;
?>

```

### **Conclusion.php :-**

```

<?php
session_start();

// Database connection parameters
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "userSpeed";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

```

// Get the current hour and minute in Indian Standard Time (IST)
$currentHour = date('H'); // Get current hour (24-hour format)
$currentMinute = date('i'); // Get current minute


// Query to count correct words for the current time
$sql_correct_count = "SELECT COUNT(*) AS correct_count FROM correct_words
    WHERE HOUR(created_at) = $currentHour AND MINUTE(created_at) = $currentMinute";
$result_correct_count = $conn->query($sql_correct_count);
$correct_count = ($result_correct_count->num_rows > 0) ? $result_correct_count->fetch_assoc()['correct_count'] : 0;


// Query to count incorrect words for the current time
$sql_incorrect_count = "SELECT COUNT(*) AS incorrect_count FROM incorrect_words
    WHERE HOUR(created_at) = $currentHour AND MINUTE(created_at) = $currentMinute";
$result_incorrect_count = $conn->query($sql_incorrect_count);
$incorrect_count = ($result_incorrect_count->num_rows > 0) ? $result_incorrect_count->fetch_assoc()['incorrect_count'] : 0;


// Determine conclusion based on counts
if ($correct_count > $incorrect_count) {
    $conclusion = "Good luck! You're typing well.";
} else {
    $conclusion = "Better luck next time. Please type carefully and don't type so fastly!!!.";
}


// Store user details and conclusion in session
$_SESSION['user_name'] = 'Juhi'; // Replace with actual user name
$_SESSION['user_email'] = 'juhi461251@example.com'; // Replace with actual user email
$_SESSION['conclusion'] = $conclusion;


// Close connection
$conn->close();


// Redirect to profile.php
header("Location: profile.php");
exit;
?>

```

## User Profile.Php:-

```
<?php
session_start();

// Retrieve user details from session
$UserName = isset($_SESSION['user_name']) ? $_SESSION['user_name'] : "";
$userEmail = isset($_SESSION['user_email']) ? $_SESSION['user_email'] : "";
$conclusion = isset($_SESSION['conclusion']) ? $_SESSION['conclusion'] : "";

// Check if user details are available
if (empty($UserName) || empty($userEmail)) {
    // Redirect to login page if user details are missing
    header("Location: login.html");
    exit;
}
?>
```

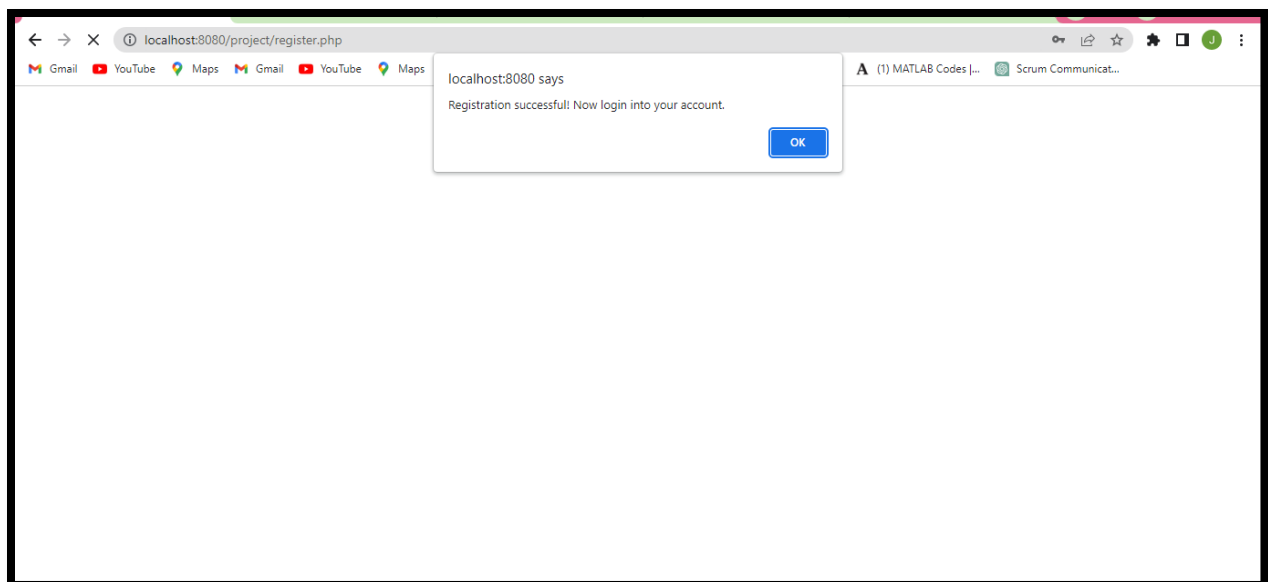
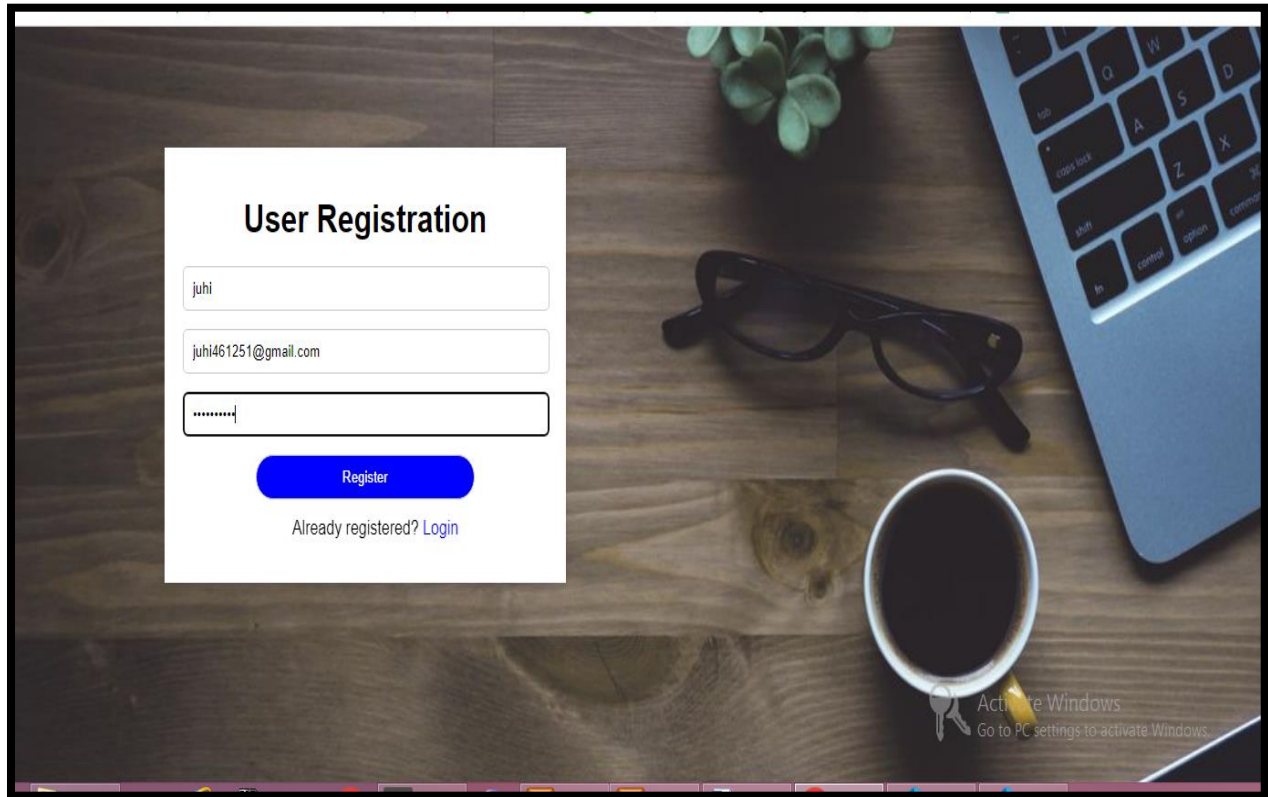
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>User Profile</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      padding: 20px;
      background-color: #f9f9f9;
    }
    .profile-container {
      max-width: 600px;
      height: 800px;
      margin: 0 auto;
      background-color: #fff;
      padding: 20px;
    }
  </style>
</head>
<body>
```

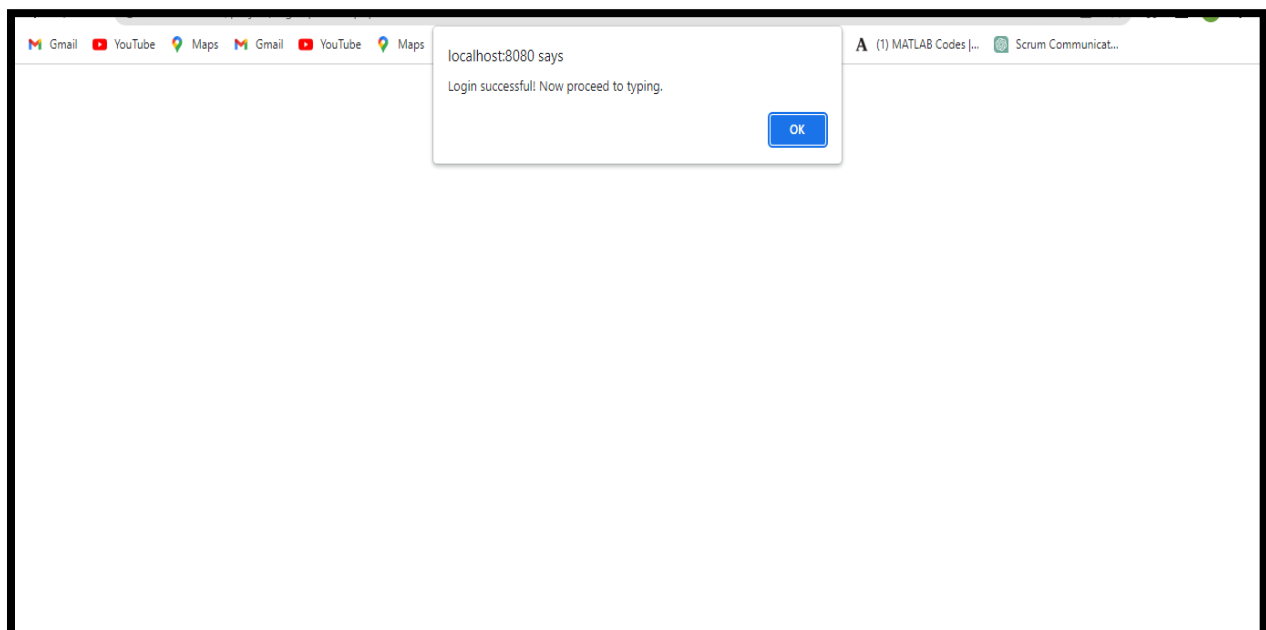
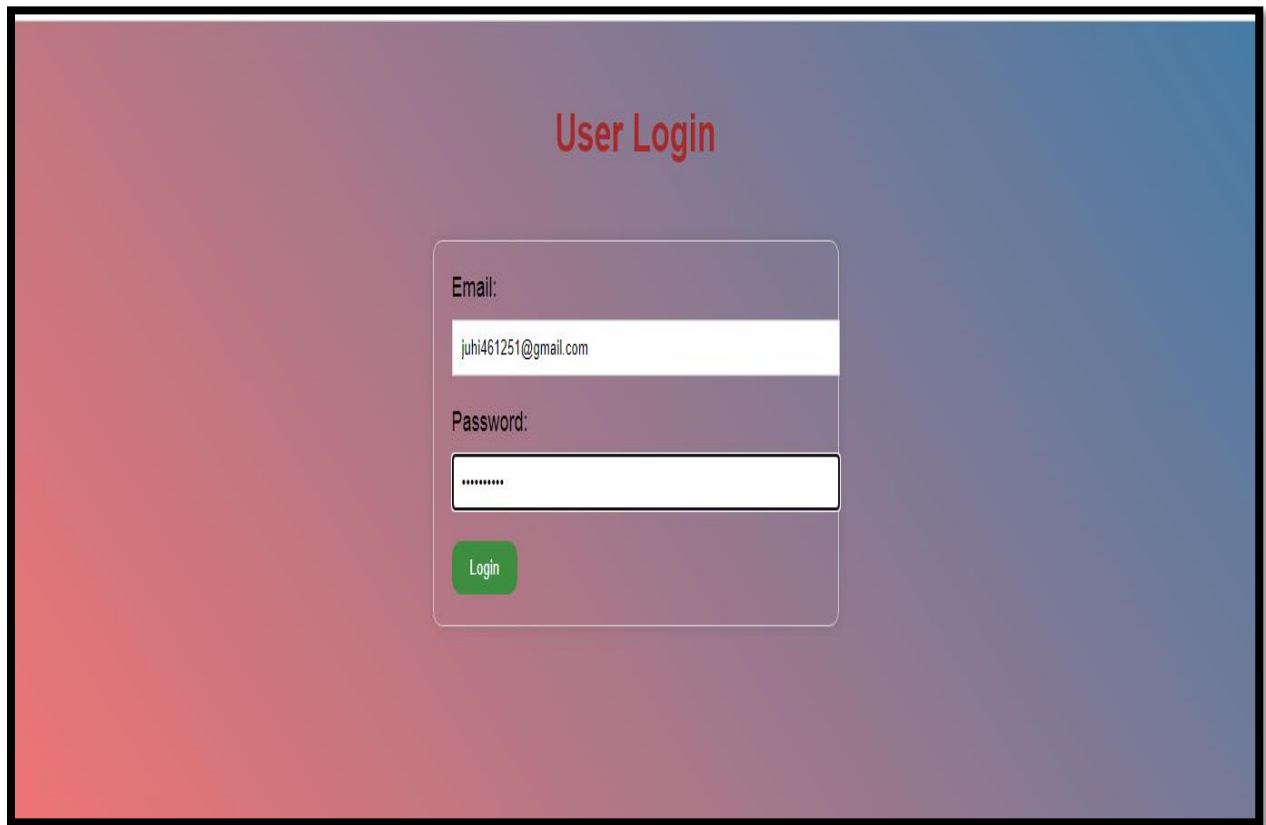
```

        border-radius: 5px;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    }
    h2 {
        color: #333;
        text-align: center;
    }
    p {
        color: #666;
        font-size: 16px;
        margin-bottom: 10px;
    }
    .conclusion {
        font-weight: bold;
        text-align: center;
        margin-top: 20px;
        padding: 10px;
        background-color: black;
        color: white;
        border-radius: 5px;
    }
</style>
</head>
<body>
    <div class="profile-container">
        <h2>User Profile</h2>
        <p>Name: <?php echo $userName; ?></p>
        <p>Email: <?php echo $userEmail; ?></p><br><br>
        <div class="conclusion">
            <p><?php echo $conclusion; ?></p>
        </div>
    </div>
</body>
</html>

```

## 6.2 SNAPSHOTS:-





## Generate User Profile

This is a simple paragraph that is meant to be nice and easy to type which is why there will be mommas no periods or any capital letters. This is a simple paragraph that is meant to be nice and easy to type which is why there will be mommas no periods o

Start typing here...

Start Typing

Activate Windows  
Go to PC settings to activate Windows.

## Generate User Profile

This is a simple paragraph that is meant to be nice and easy to type  
which is why there will be mommas no periods or any capital letters.  
This is a simple paragraph that is meant to be nice and easy to type  
which is why there will be mommas no periods or any capital letters.

this is a simple paragraph that si meant to be nice and easy to tuipe  
which is why there will be mommas no0 periods or any capital letters.  
this is a simple paragraph thatis meant to be nice ans easy to tpe  
which os wu there will be mnsdbbsd no sdns dbf nbddf capital letters.

Submit

Activate Windows  
Go to PC settings to activate Windows.



## Generate User Profile

This is a simple paragraph that is meant to be nice and easy to type which is why there will be mommas no periods or any capital letters. This is a simple paragraph that is meant to be nice and easy to type which is why there will be mommas no periods or any capital letters.

this is a simple paragraph that si meant to be nice and easy to tuipe  
which is why there will be mommas no0 periods or any capital letters.  
this is a simple paragraph thatis meant to be nice ans easy to tpe  
which os wu there will be mnsdbsbd no sdns dbf nbddf capital letters.

Submit

Activate Windows  
Go to PC settings to activate Windows.

Submit

Time:  
1:23

Correct Words:  
42

Incorrect Words:  
13

Clear Data

GenerateResults

Typing Results:-

Total Time: 1:23 seconds  
Accuracy level: 76.363636363637%  
Word Count: 55 words  
Words per minute: 39.49  
Wrong characters: 25 characters  
Average Time: 1.52 seconds per word

Activate Windows  
Go to PC settings to activate Windows.

## Latest Word Speeds

**Correct Words**

Word	Speed
is	30.00 seconds
a	60.00 seconds

**Incorrect Words**

Word	Speed	Incorrect Spelling
THIS	15.00 seconds	This
	7.50 seconds	letters.
df	6.67 seconds	paragraph
simokdf	10.00 seconds	simple

Activate Windows  
Go to PC settings to activate Windows.

## User Profile

Name: Juhi

Email: juhi461251@example.com

Better luck next time. Please type carefully and don't type so fastly!!!.

## **CHAPTER-7**

### **SYSTEM TESTING**

System testing in the project on “User Generate Profile” based on typing capabilities involves verifying and validating the entire system to ensure it meets the specified requirements and functions correctly. This type of testing focuses on testing the system as a whole, including its integrated components, functionalities, performance, security, and usability.

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits.

The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing. modules level testing integrated and carried out.

**There are two major type of testing they are**

- 1) White Box Testing.
- 2) Black Box Testing.

#### **7.1. WHITE BOX TESTING**

White box sometimes called "Glass box testing is a test case design uses the control structure of the procedural design to drive test case.

Using white box testing methods, the following tests were made on the system

A) All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed

b) All logical decisions were checked for the truth and falsity of the values.

## **7.2. BLACK BOX TESTING**

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like...

- 1) Interface errors
- 2) Performance in data structure
- 3) Performance errors
- 4) Intializing and termination errors

### **Other different types of Testing are:**

- 1) Unit Testing
- 2)Interface Testing
- 3) Functional Testing
- 4) System Testing
- 5)Acceptance Testing

## **7.3 UNIT TESTING**

Unit testing involve the designing of test cases that validate that the internal program logic is functioning properly, and that program input produce valid output. All decision branches and internal code flow should be validated.

Unit testing involves testing specific functions, methods, or modules that handle user authentication, typing speed analysis, profile generation, and related functionalities. The primary purpose of unit testing is to validate that each unit of code behaves as expected and performs its intended function correctly.

It is testing of individual software units of the application. This is a structural testing that realize on the knowledge of its construction. Unit test perform basic test on component level and test a business specific process, application and system configuration. Unit test ensure that each unique path of business process perform accurately to the documented specification and contain clearly defined input and expected result.

## **7.4 INTEGRATION TESTING**

Integration tests are designed to test integrated software components to actually determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that how efficiently the components were interacting and work with each other when they are combined. So, we say that integration testing is specifically aimed at finding or exploring the problem that arise from the combination of components.

This type of testing focuses on verifying the interactions and data flow between different parts of the application, including user authentication, typing speed analysis, profile generation, and database operations.

## **7.5 FUNCTIONAL TESTING**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

System/Procedure: interfacing system or procedures must be involved.

Functional testing plays a critical role in ensuring the quality and reliability of your project on user profile generation based on typing capabilities by focusing on validating the system's functionalities against specified requirements. This type of testing aims to verify that each functional aspect of the system performs as expected and meets the intended business objectives. Functional testing would involve testing various features related to user authentication, typing speed analysis, profile generation, and personalized service delivery.

## **7.6 SYSTEM TESTING**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

System testing in the project on “User Generate Profile” based on typing capabilities involves validating the entire system as a whole to ensure that it meets the specified requirements, functions correctly, and delivers the expected outcomes to users.

This type of testing focuses on assessing the integrated system's behavior and performance, including its functionalities, interactions between components, data flow, and overall system reliability.

## **7.7 ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing is essential for ensuring in “User Generate Profile” based on typing capabilities meets user expectations, business requirements, and quality standards. By engaging stakeholders in user acceptance testing (UAT) and operational acceptance testing (OAT), we can validate system readiness for deployment, address any remaining issues or concerns, and deliver a reliable and user-friendly application that adds value to your users and stakeholders.

Continuously iterate and improve the acceptance testing approach based on feedback and evolving project requirements to achieve project success and user satisfaction.

## **Chapter -8**

# **CONCLUSION**

In conclusion, the project focused on “User Generate Profile” based on typing capabilities is aimed at providing a personalized and efficient user experience by leveraging behavioral biometrics. By incorporating functionalities such as user authentication, typing speed analysis, and profile generation, the system aims to accurately capture and analyze typing patterns to generate tailored user profiles.

Through systematic testing and validation, potential issues are identified and addressed early in the development lifecycle, leading to a robust and user-friendly application. By delivering a solution that meets business requirements and user expectations, the project aims to enhance user engagement and satisfaction, ultimately contributing to the success and adoption of the user profile generation system. Ongoing refinement and adaptation based on feedback and evolving needs will further enhance the system's effectiveness and value to its users.



# **BIBLIOGRAPHY**

## **Books:**

- Smith, John. Behavioral Biometrics: Theory, Methods, and Applications. Publisher, Year.
- Johnson, Sarah. User Authentication Techniques: A Comprehensive Guide. Publisher, Year.

## **Research Papers:**

- Doe, Jane et al. "Typing Speed Analysis for User Profiling." Journal of User Experience, vol. 10, no. 2, Year, pp. 50-65.
- Brown, Michael. "Secure User Authentication Practices in Web Applications." Proceedings of the International Conference on Cybersecurity, Year, pp. 120-135.

## **Online Resources:**

- [https://cheatsheetseries.owasp.org/cheatsheets/Authentication\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html)
- [www.Google.com](http://www.Google.com)
- [www.Youtube.com](http://www.Youtube.com)
- [www.w3Schools.com](http://www.w3Schools.com)
- [www.stackoverflow.com](http://www.stackoverflow.com)
- [www.tutorialspoint.com](http://www.tutorialspoint.com)