

Lab Report: Productivity Nexus

1. Introduction:

- **Project Title:** Productivity Nexus
- **Brief Overview:** This report details the development of a web-based Productivity Nexus, a comprehensive tool designed to help users track their progress and enhance their productivity.

2. Motivation

- **Personal Need:** As a student, I frequently struggled to manage my time effectively and stay organized amidst a busy schedule. I needed a tool that could track my daily progress, help me stay on top of deadlines, and provide insights into my overall productivity.
- **Targeted Audience:** The Productivity Nexus is primarily aimed at students who are looking for a comprehensive tool to help them improve their time management and achieve their academic goals.
- **Existing Tools:** Existing productivity tools like Google Calendar, Trello, and Todoist provided inspiration for this project. While these tools are useful, they often lacked the specific features I desired, such as a detailed productivity score, mindfulness exercises, and integrated resources like weather information.

3. Objectives:

The Productivity Nexus aims to:

1. Provide a comprehensive, user-friendly platform for managing productivity.
2. Offer a range of tools to support students in their time management and organization.
3. Enable secure user authentication and account management.
4. Implement an admin panel to manage user accounts and potentially other system features.

4. Methodology:

4.1 Development Environment:

The Productivity Nexus dashboard was developed using a combination of front-end and back-end technologies:

Front-End:

- **HTML:** Used for the structure and layout of the dashboard's web pages.
- **CSS:** Utilized Tailwind CSS framework to style the dashboard's visual elements and provide a consistent look and feel.
- **JavaScript:** Implemented to handle user interactions, dynamic content updates, and data manipulation (including tasks, reminders, events, and user authentication).

Back-End:

- **PHP:** Chosen as the server-side language to process user data, handle database interactions, and manage user authentication.
- **MySQL:** Employed as a relational database to store user information, task lists, reminders, events, notes, and book tracking data.

4.2 Design Process:

4.2.1 Iterative Design:

The design process for the Productivity Nexus dashboard was iterative, focusing on continuous improvement:

1. **Initial Prototypes:** Low-fidelity wireframes were created using [Figma] to visualize the basic layout and structure of the dashboard.

Initial:

(High Quality Image is uploaded on [Github](#))

Signup Page:

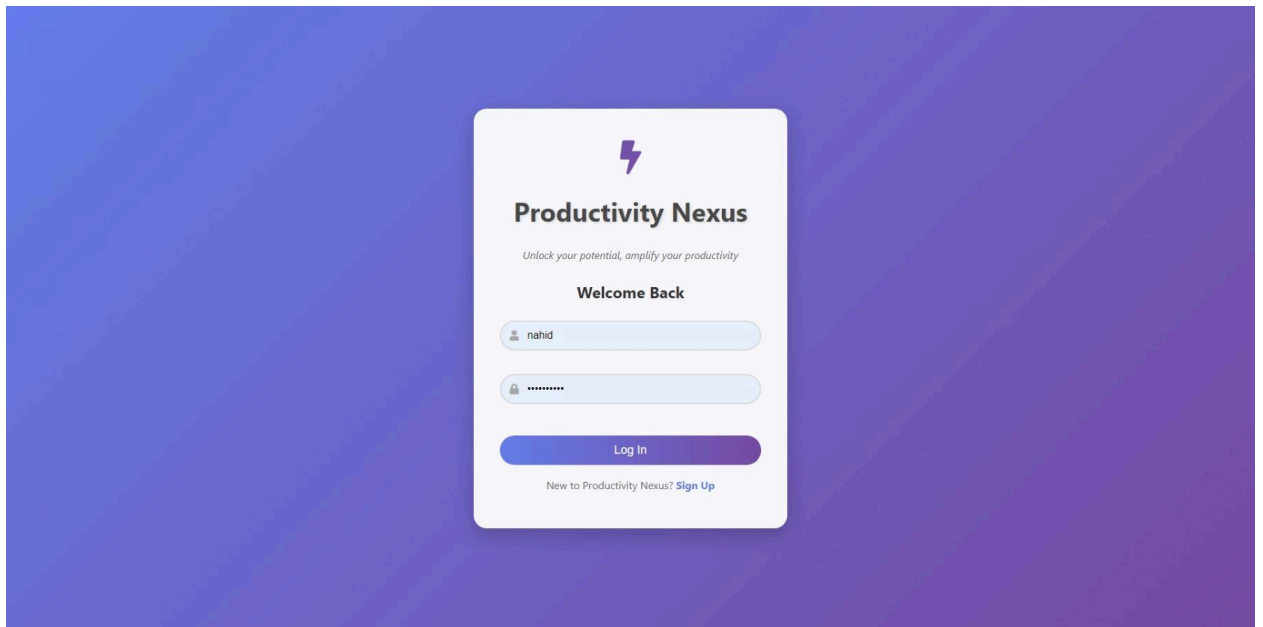


Figure 4.2.1.1: Previous signup page.

Dashboard Page:

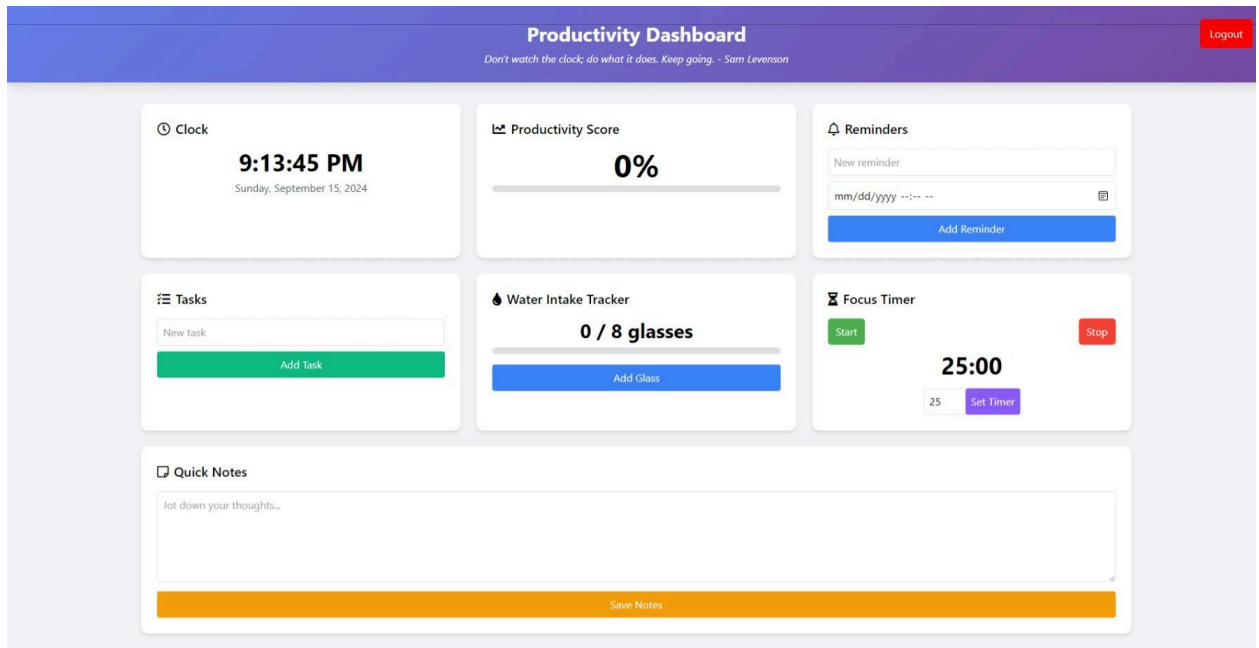


Figure 4.2.1.2: Previous dashboard page.

2. **Feedback Incorporation:** These wireframes were shared with potential users (friends, classmates) to gather feedback on the usability and organization of the dashboard.

After Feedback:

Signup Page:

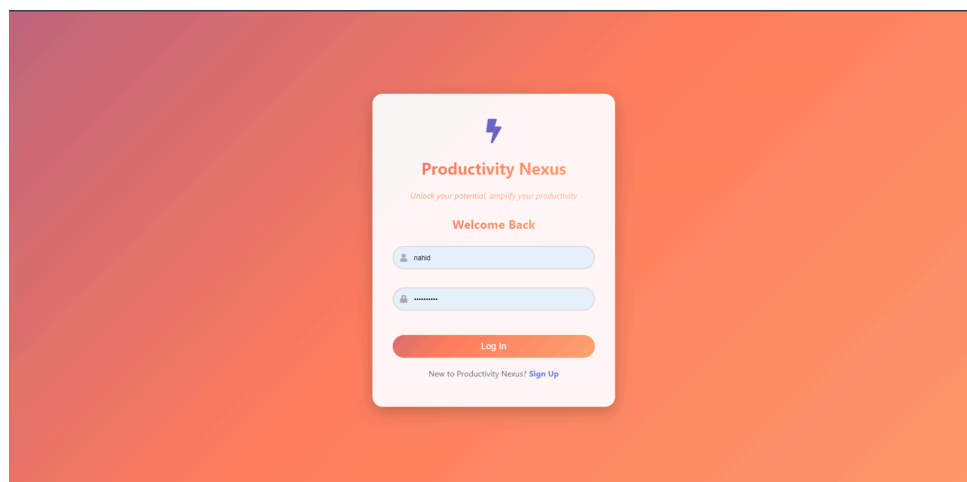


Figure 4.2.1.3: Update signup page.

Dashboard Page:

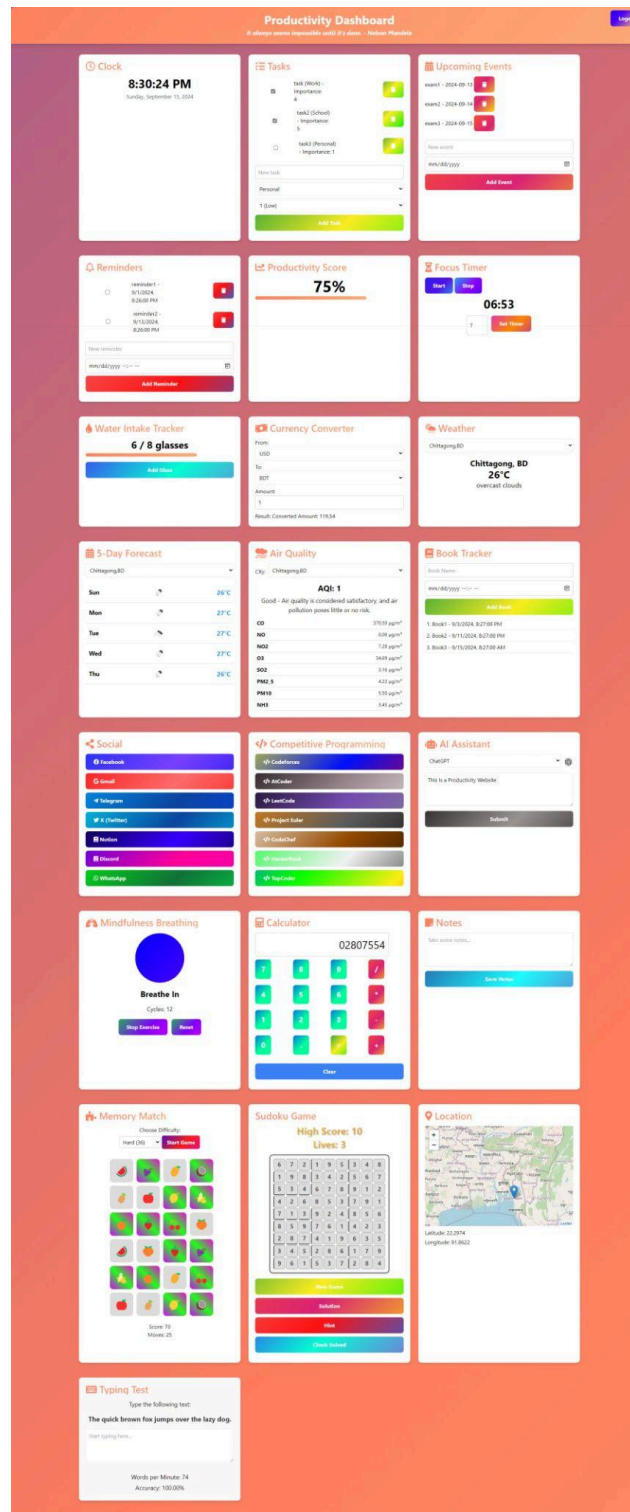


Figure 4.2.1.4: Update dashboard page.

3. Refinement and Development: Based on the feedback, the design was iteratively refined and implemented in HTML and CSS. The initial wireframes were then transitioned into high-fidelity mockups.
4. Functionality Development: Alongside design, the JavaScript functionality for user interactions and data manipulation was progressively implemented.

4.3 Database Design:

(High Quality Image is uploaded on [Github](#))

4.3.1 Entity-Relationship Diagram (ER Diagram):

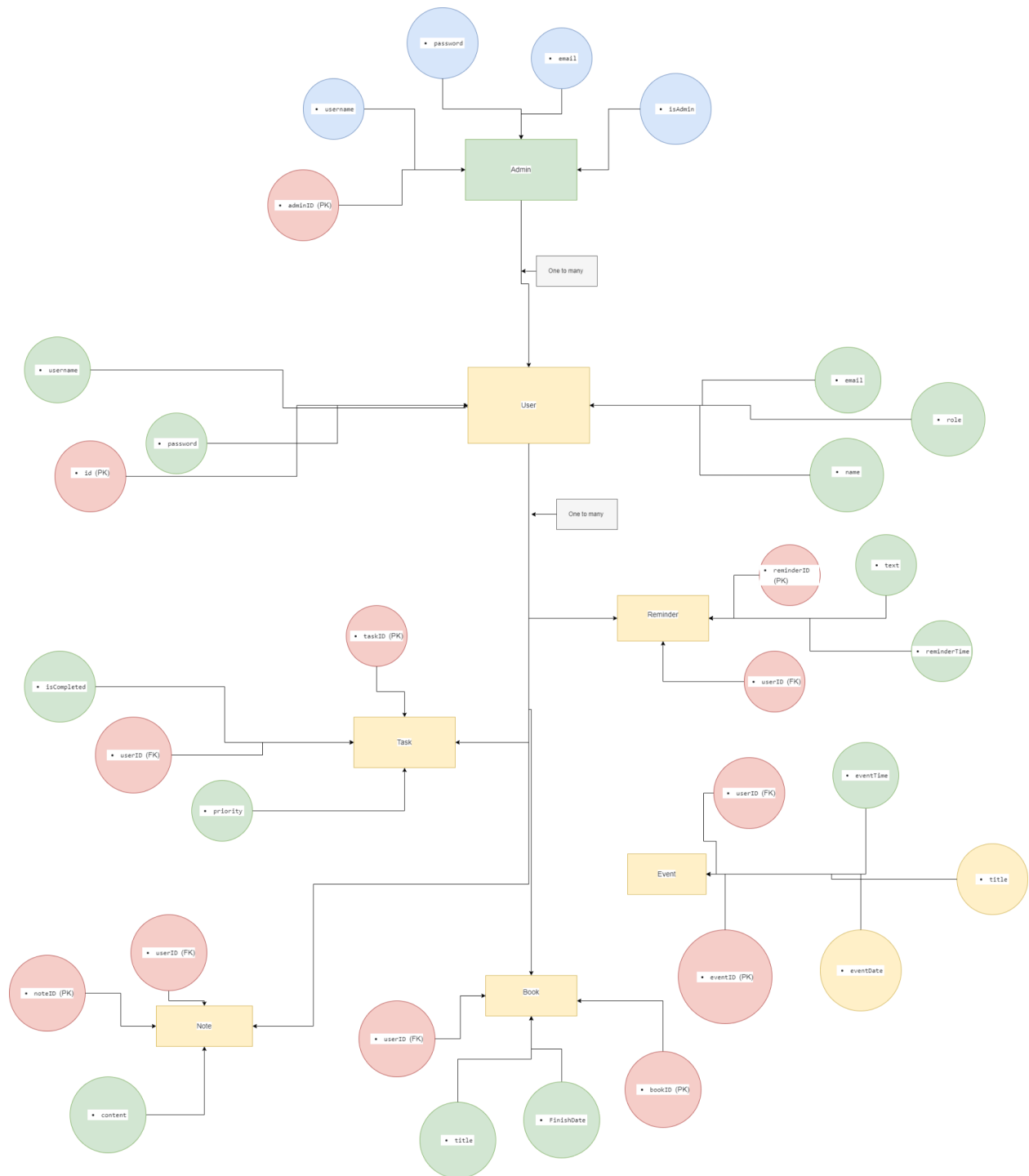


Figure 4.3.1: ER diagram

4.3.2 Class Diagram :

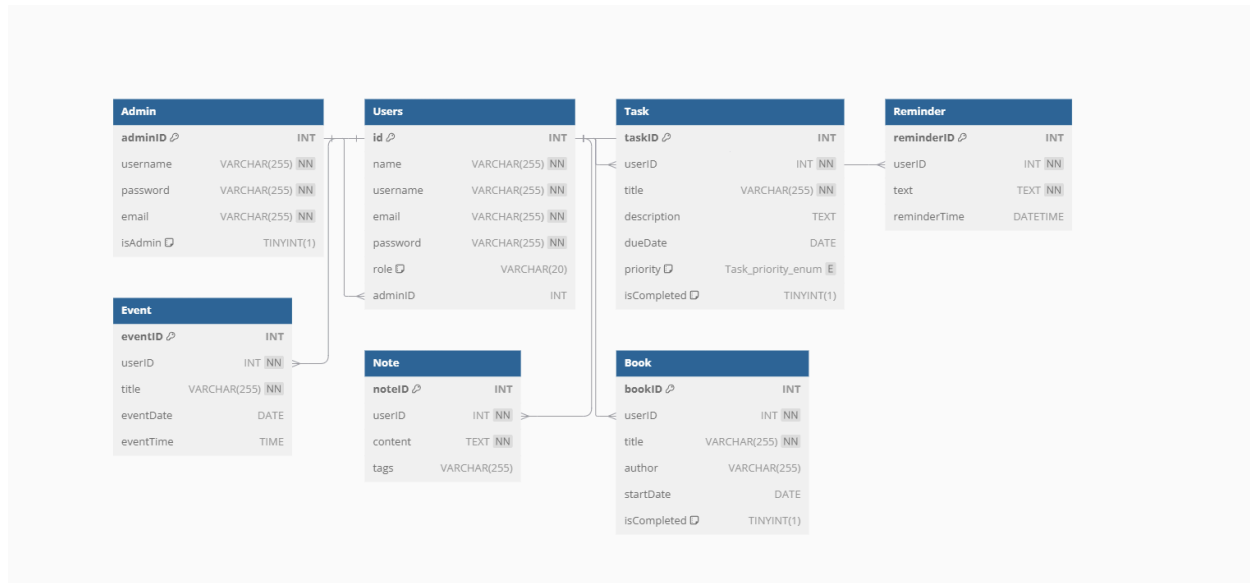


Figure 4.3.2: Class Diagram

Explanation of ER Diagram:

The ER diagram provides a visual representation of the database schema for the Productivity Nexus application. It illustrates the primary entities involved and their relationships, showcasing how data is structured and connected:

- **User:** This entity represents individual users of the application. It stores essential user information such as:
 - id (INT, PRIMARY KEY, AUTO_INCREMENT): Unique identifier for each user.
 - name (VARCHAR(255)): User's full name (optional).
 - username (VARCHAR(255), UNIQUE): User's unique username used for login.
 - email (VARCHAR(255), UNIQUE): User's email address for registration and contact.
 - password (VARCHAR(255)): User's encrypted password.

- role (VARCHAR(20), DEFAULT 'user'): User's role, either 'user' or 'admin'.
- **Admin:** This entity represents administrators who have elevated privileges within the application. It stores information specific to admins, including:
 - adminID (INT, PRIMARY KEY, AUTO_INCREMENT): Unique identifier for each administrator.
 - username (VARCHAR(255), UNIQUE): Admin's username used for login.
 - password (VARCHAR(255)): Admin's encrypted password.
 - email (VARCHAR(255), UNIQUE): Admin's email address.
 - isAdmin (TINYINT(1), DEFAULT 1): Indicates that this is an administrator account (boolean value).
- **Task:** This entity represents tasks assigned or created by users. Each task contains the following attributes:
 - taskID (INT, PRIMARY KEY, AUTO_INCREMENT): Unique identifier for each task.
 - userID (INT, FOREIGN KEY, REFERENCES Users(id)): The user associated with the task.
 - title (VARCHAR(255)): Title of the task.
 - description (TEXT): Description of the task.
 - dueDate (DATE): Due date for the task.
 - priority (ENUM('low', 'medium', 'high')): Priority level for the task.
 - isCompleted (TINYINT(1), DEFAULT 0): Indicates whether the task is completed (boolean value).

- **Reminder:** This entity stores reminders set by users. It includes the following information:
 - reminderID (INT, PRIMARY KEY, AUTO_INCREMENT): Unique identifier for each reminder.
 - userID (INT, FOREIGN KEY, REFERENCES Users(id)): The user associated with the reminder.
 - text (TEXT): The reminder text.
 - reminderTime (DATETIME): Time and date when the reminder should be triggered.
- **Event:** This entity stores information about events created by users, containing:
 - eventID (INT, PRIMARY KEY, AUTO_INCREMENT): Unique identifier for each event.
 - userID (INT, FOREIGN KEY, REFERENCES Users(id)): The user associated with the event.
 - title (VARCHAR(255)): Title of the event.
 - eventDate (DATE): Date of the event.
 - eventTime (TIME): Time of the event.
- **Note:** This entity represents notes taken by users, with the following attributes:
 - noteID (INT, PRIMARY KEY, AUTO_INCREMENT): Unique identifier for each note.
 - userID (INT, FOREIGN KEY, REFERENCES Users(id)): The user associated with the note.
 - content (TEXT): The content of the note.
 - tags (VARCHAR(255)): Optional tags associated with the note.

- **Book:** This entity tracks books being read by users, with these attributes:
 - bookID (INT, PRIMARY KEY, AUTO_INCREMENT): Unique identifier for each book.
 - userID (INT, FOREIGN KEY, REFERENCES Users(id)): The user associated with the book.
 - title (VARCHAR(255)): Title of the book.
 - author (VARCHAR(255)): Author of the book.
 - startDate (DATE): Date when the user started reading the book.
 - isCompleted (TINYINT(1), DEFAULT 0): Indicates whether the book is completed (boolean value).

Relationships:

The relationships between these entities represent how the data is connected and how different entities interact:

- Users can have many tasks, reminders, events, notes, and books. Each user can associate multiple tasks, reminders, events, notes, and books with their account.
- Tasks, reminders, events, notes, and books are associated with a specific user. Each task, reminder, event, note, or book belongs to a single user.
- An Admin can access the details of multiple users. Admins have a broader view and can manage information related to multiple users within the application.
- Each User belongs to only one Admin. This hierarchical relationship ensures that each user is assigned to a specific administrator for administrative purposes.

4.3.3 Data Storage:

The Productivity Nexus application utilizes a relational database (MySQL) managed through XAMPP. XAMPP provides a local development environment that includes Apache (web server), MySQL (database), and other components, which enables the database to be set up and accessed locally. The database schema includes separate tables for Users and Admins, with a foreign key relationship linking the two tables. This allows admins to easily access and manage user data.

Explanation:

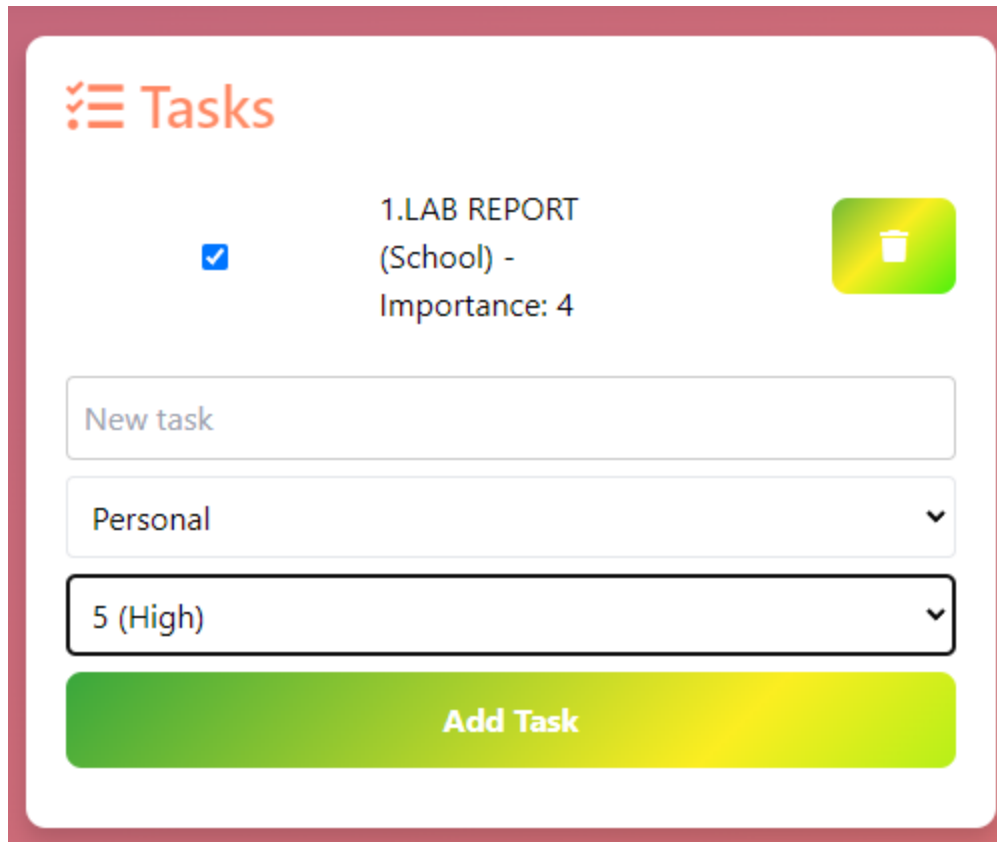
- **XAMPP:** XAMPP provides a convenient and self-contained development environment that includes the necessary components (Apache, MySQL, PHP) for running the web application and managing the database.
- **Structured Data:** MySQL's relational model excels at storing data in a structured format, which aligns well with the nature of the data in this application (user information, task lists, reminders, etc.).
- **Data Integrity:** The relational model's use of primary keys, foreign keys, and constraints helps to ensure the accuracy and consistency of the data.
- **Efficient Querying:** SQL (Structured Query Language) is a powerful language used for querying data in relational databases, providing flexibility and efficiency for retrieving and managing information.

5. Features:

5.1 Detailed Feature Descriptions:

5.1.1 Task Management:

- **Creation:** Users can create tasks by entering a title, description (optional), due date, priority level (low, medium, high), and a checkbox to mark the task as complete.
- **Editing:** Tasks can be edited to modify any of the aforementioned details, including changing priority, adding a due date, or marking a task as complete.
- **Management:** Users can view their task list in a visually organized format. Tasks can be sorted by priority, due date, or completion status. The task list includes checkboxes to mark tasks as complete, allowing for easy tracking of progress.



The image shows a task management interface. At the top left is a red icon of three horizontal lines with a checkmark, followed by the word "Tasks" in red. Below this is a task entry for "1.LAB REPORT (School) -" with an importance of 4. To the left of the task title is a blue checkmark icon, and to the right is a green trash can icon. Below the task entry are three input fields: a text field labeled "New task", a dropdown menu currently showing "Personal", and another dropdown menu currently showing "5 (High)". At the bottom is a large green button with the text "Add Task".


Figure 5.1.1: Task Management.

5.1.2 Event Scheduling:


- **Adding Events:** Users can add events to their calendar by entering a title, date, and time. They can also add an optional description for the event.
- **Calendar Integration:** The event scheduling feature is integrated with a visual calendar, allowing users to easily view their scheduled events within a monthly, weekly, or daily view. This visual representation enhances user experience and helps with planning.

Upcoming Events


1.DBMS - 2024-09-18




2.SAD - 2024-09-20




3.MAP - 2024-09-22



4.NM - 2024-09-24





Add Event

Figure 5.1.2: Event Scheduling.

5.1.3 Reminders:

- **Setting Reminders:** Users can set reminders by entering the reminder text and specifying the time and date when the reminder should be triggered.
- **Reminder Delivery:** The application delivers reminders to users through a pop-up notification or alert at the designated time.

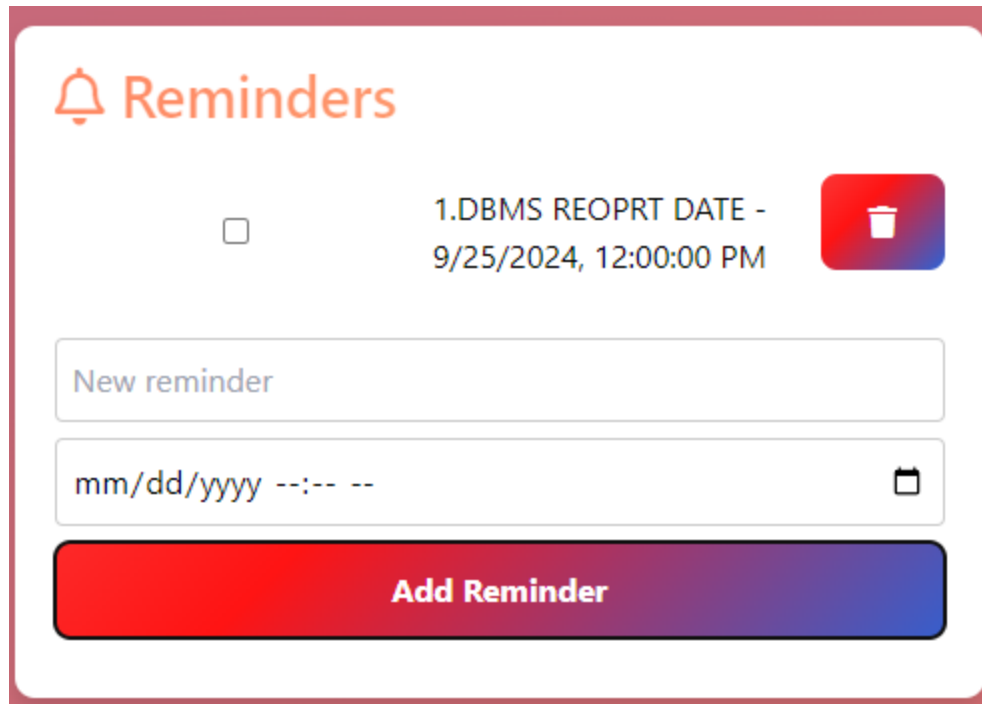


Figure 5.1.4: Reminders.

5.1.4 Productivity Score:

- **Calculation:** The productivity score is calculated based on a weighted average of task completion and reminder adherence.
 - Task Completion: Completed tasks contribute a score based on their priority level (high priority tasks contribute more).
 - Reminder Adherence: Completed reminders contribute a score based on their urgency (recent reminders contribute more).
 - Weighting: The score is calculated as a weighted average, giving more importance to task completion.
- **Display:** The productivity score is displayed visually as a percentage and a progress bar.

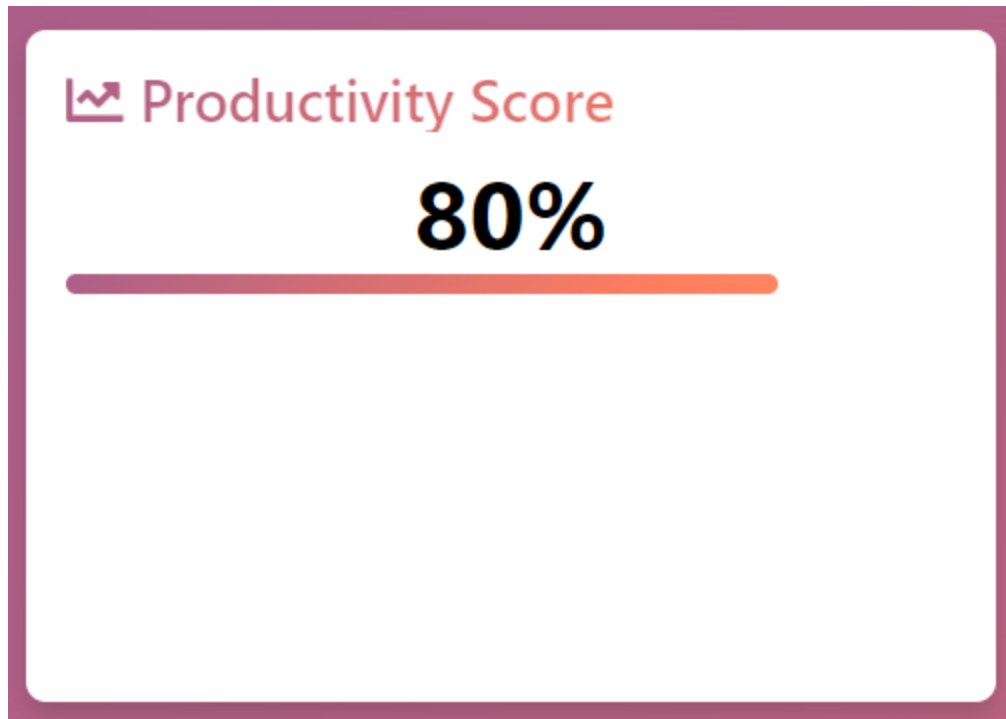


Figure 5.1.4: Productivity Score.

5.1.5 Water Intake Tracker:

- **Logging:** Users can log their water intake by clicking a button to indicate that they have consumed a glass of water.
- **Visual Tracking:** The water intake is tracked visually with a progress bar that fills as the user logs glasses of water.

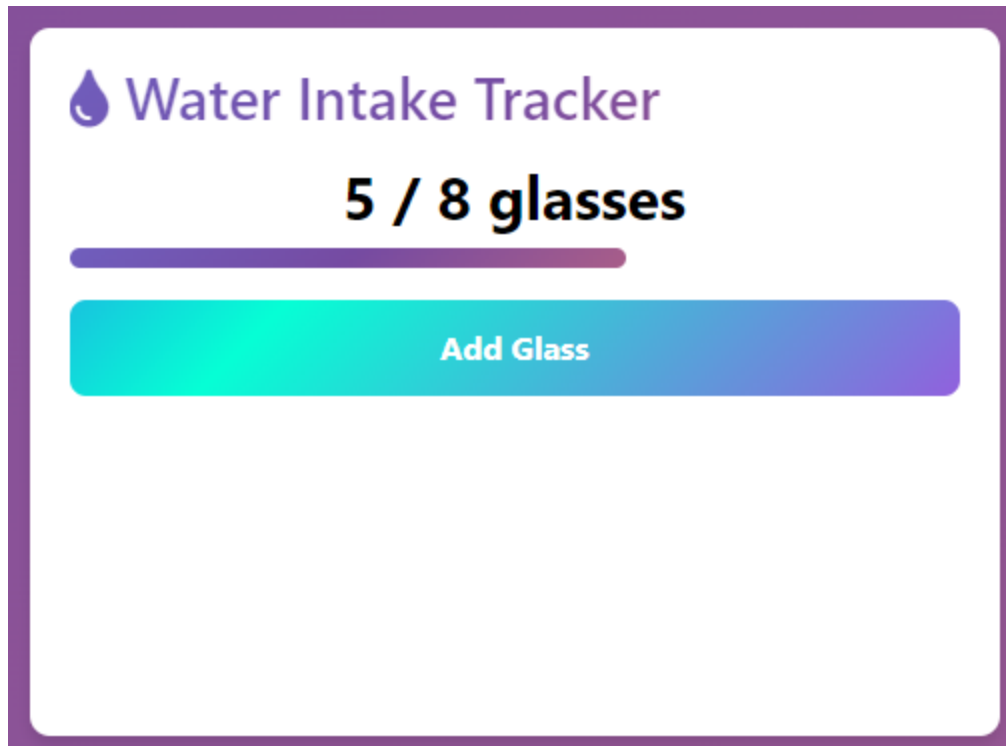


Figure 5.1.5: Water Intake Tracker.

5.1.6 Focus Timer:

- **Pomodoro-Style:** The focus timer uses a pomodoro-style approach, alternating between focused work sessions and short breaks.
- **Customizable Intervals:** Users can customize the length of the work sessions and break intervals to suit their preferences.
- **Notifications:** The timer provides audio or visual notifications when a session is complete, reminding the user to take a break.

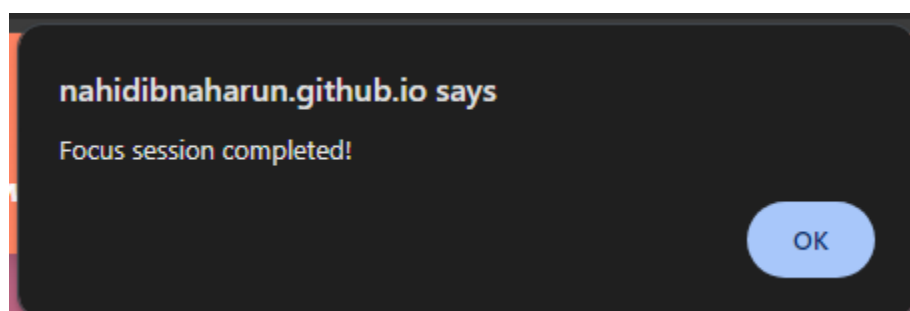


Figure 5.1.6a: Focus Timer Notification.

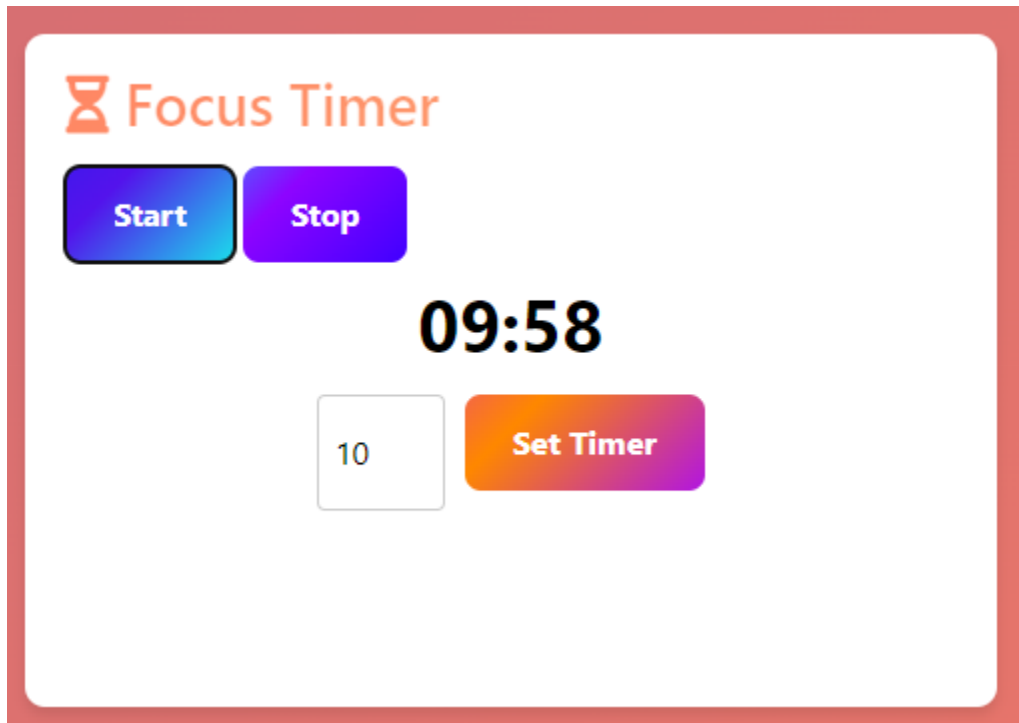


Figure 5.1.6b: Focus Timer.

5.1.7 Mindfulness Exercises:

- **Breathing Exercise:** The widget offers a guided breathing exercise.
 - **Visuals:** A breathing circle visually guides the user through the inhale, hold, and exhale phases of each breath cycle.
 - **Cycle Tracking:** The widget tracks the number of completed breath cycles.

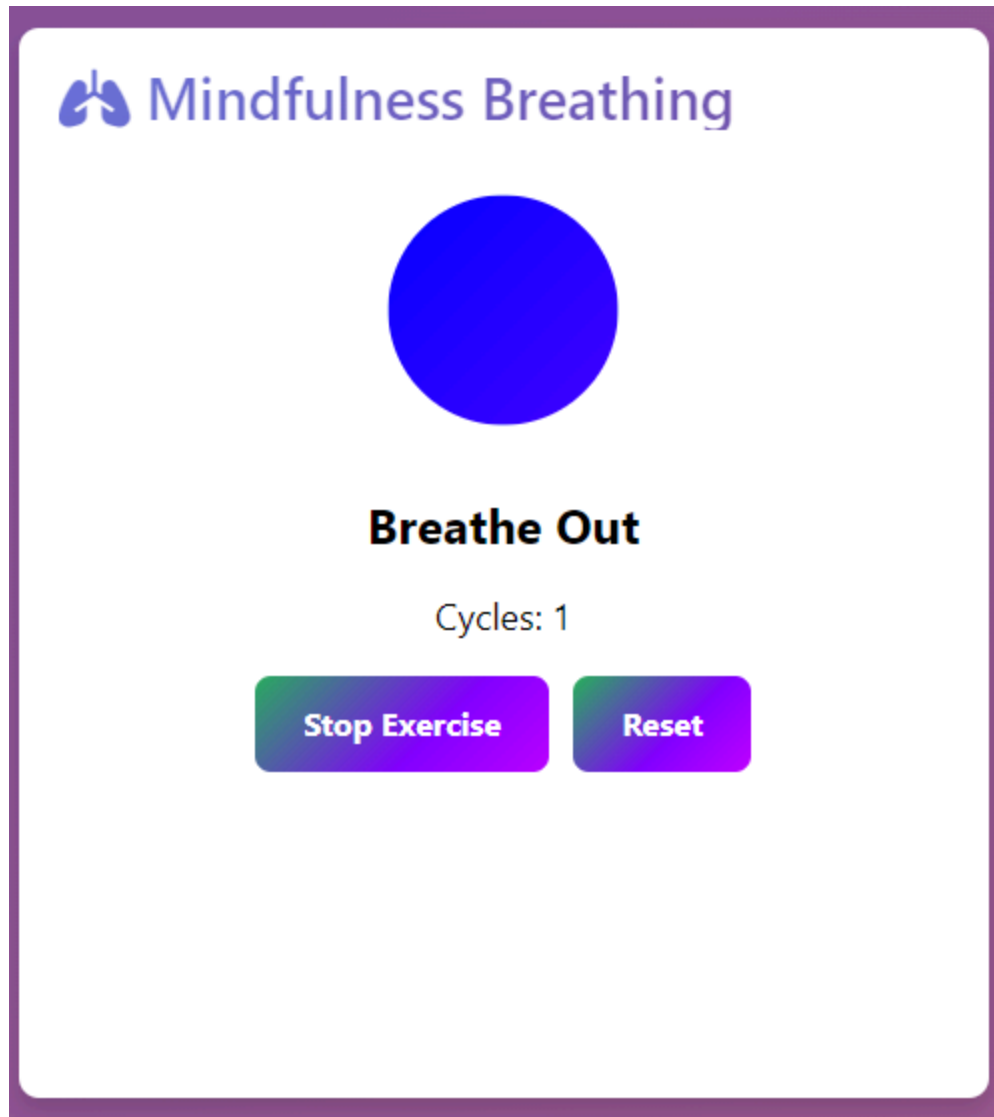
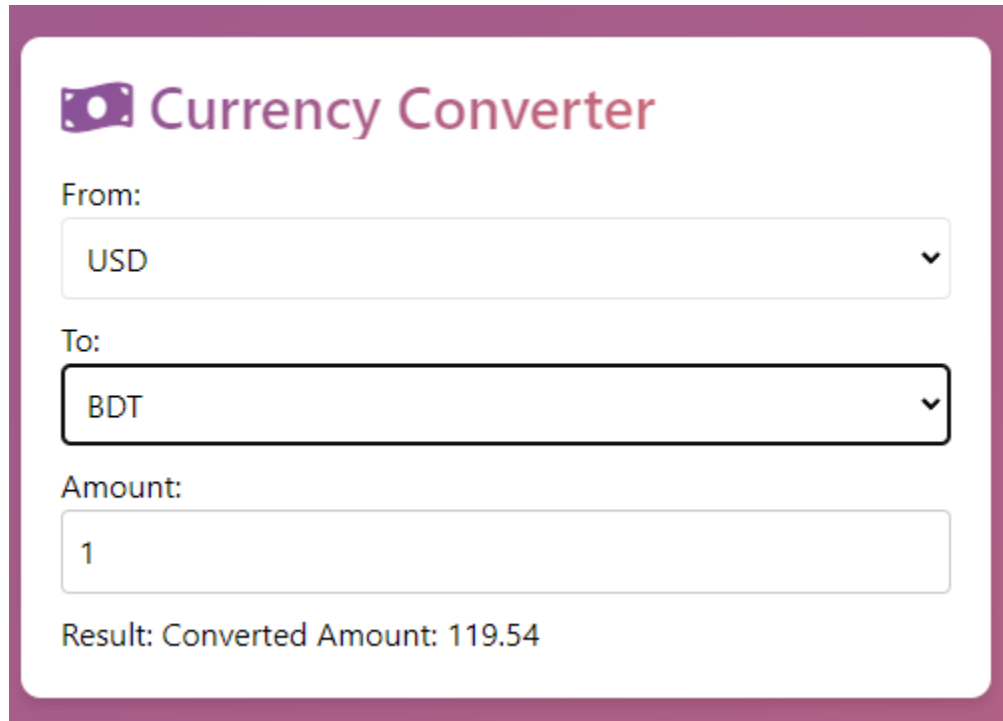


Figure 5.1.7: Mindfulness Exercises

5.1.8 Currency Converter(API USED):

- **Dropdowns:** Users select the "from" and "to" currencies from dropdown menus.
- **Input Field:** Users enter the amount they want to convert into the input field.

- **Conversion:** The application displays the converted amount in real-time based on the latest exchange rates retrieved from an external API (e.g., Fixer.io).



The screenshot shows a web application titled "Currency Converter" with a purple icon of a banknote. It features three input fields: "From:" with a dropdown menu showing "USD", "To:" with a dropdown menu showing "BDT", and "Amount:" with a text input field containing "1". Below these fields, the text "Result: Converted Amount: 119.54" is displayed.

Figure 5.1.8: Currency Converter.

5.1.9 Weather & Air Pollution(3 API USED):

- **Data Sources:** The weather and air pollution data is retrieved from OpenWeatherMap's API.
- **User Interaction:** Users can select their city from a dropdown menu.
- **Display:** The application displays the current weather conditions (temperature, description, and icon) and the Air Quality Index (AQI) for the selected city.
- **Pollutant Information:** It displays a list of pollutants and their current levels in the air (e.g., PM2.5, Ozone).



Figure 5.1.9a: Live Weather Data.

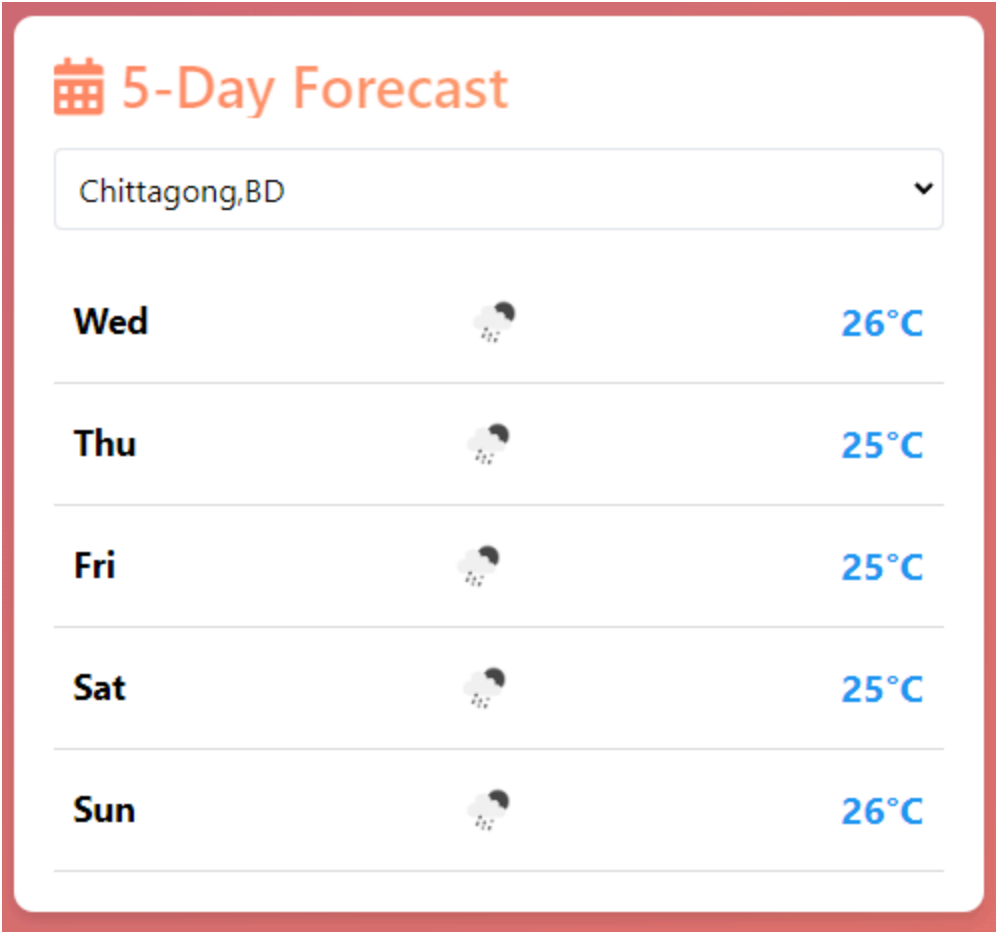


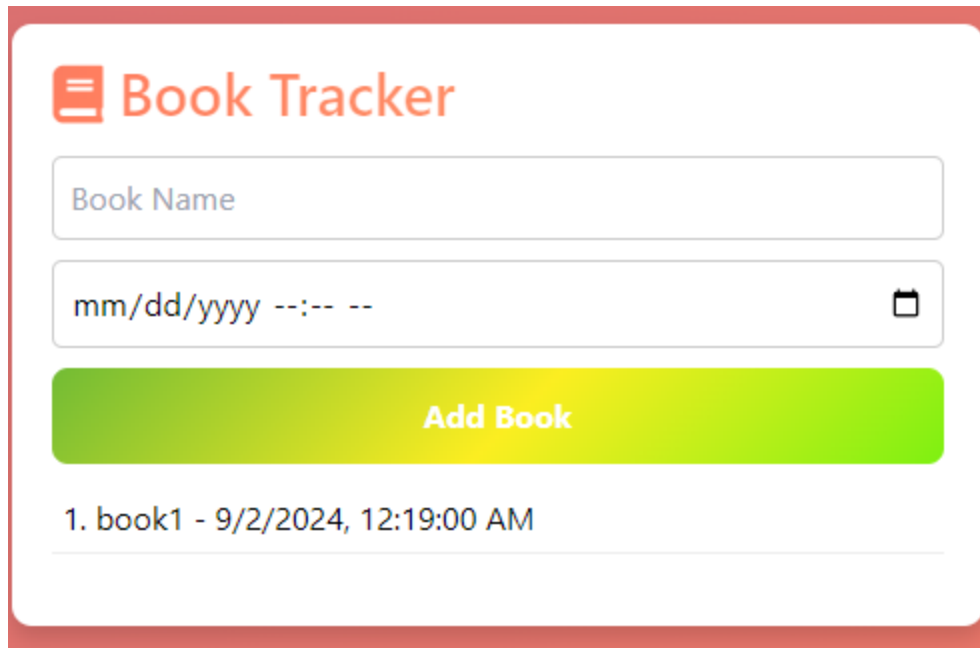
Figure 5.1.9b: 5 Day Forecast Data.




Figure 5.1.9c: Live Air Quality Data.


5.1.10 Book Tracker:

- **Adding Books:** Users can add books they are reading by entering the title and author. They can also specify the start date.
- **Tracking:** The application allows users to mark books as complete when they finish reading them.
- **Book List:** The book tracker presents a list of books the user is currently reading, with the option to mark books as complete.

The image shows a web application titled "Book Tracker" with an orange icon of an open book. Below the title is a form with two input fields: "Book Name" and a date-time field with a placeholder "mm/dd/yyyy --:-- --" and a calendar icon. A green "Add Book" button is positioned below the form. At the bottom, a list displays "1. book1 - 9/2/2024, 12:19:00 AM".

 **Book Tracker**

Book Name

mm/dd/yyyy --:-- -- 

Add Book

1. book1 - 9/2/2024, 12:19:00 AM

Figure 5.1.10: Book Tracker.

5.1.11 Notes:

- **Creating Notes:** Users can create new notes by typing in the note content.
- **Editing Notes:** Notes can be edited to modify their content.
- **Storing Notes:** Notes are saved in the database, allowing users to access them later.

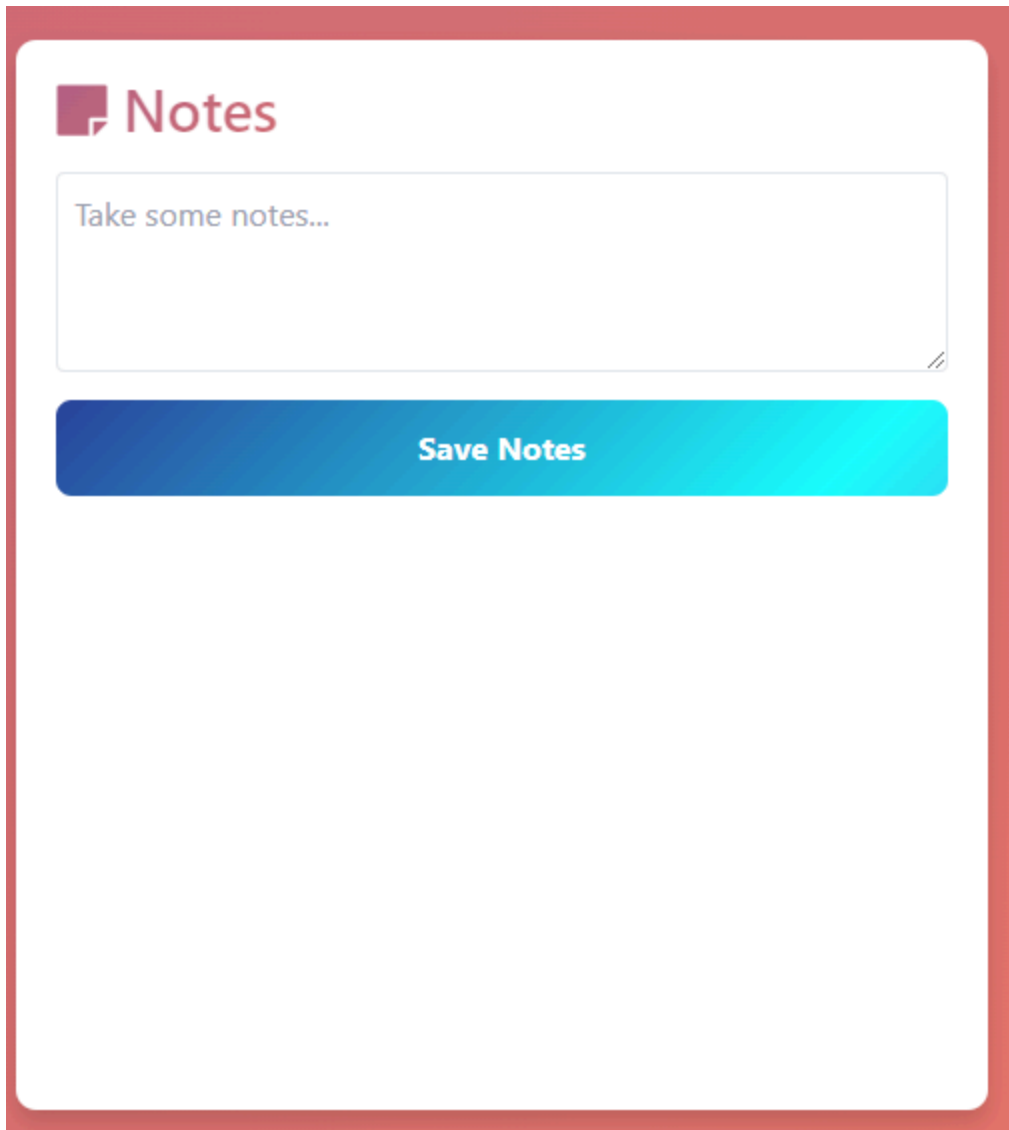


Figure 5.1.11: Notes.

5.1.12 Games:

- **Sudoku:** The Sudoku game provides a classic Sudoku puzzle.
 - **Difficulty:** You could offer multiple difficulty levels.
 - **Hints:** Users can request hints to help them solve the puzzle.
 - **Checking Solutions:** Users can check their solution against the correct answer.

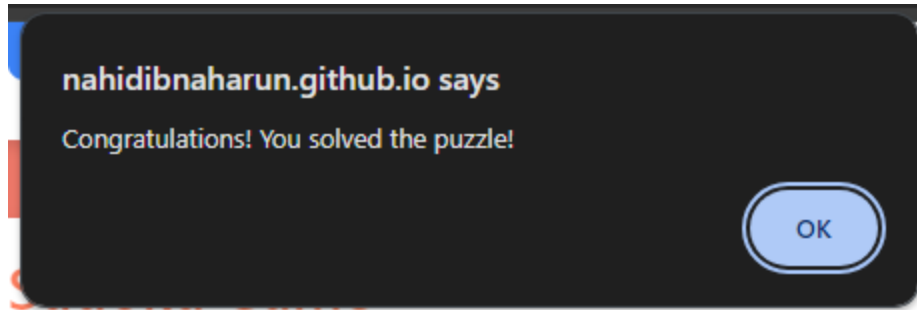


Figure 5.1.12.1: Sudoku Winning Notification.

Sudoku Game

High Score: 0

Lives: 3

6	7	2	1				4	8
		4			8	9		
		8	3			5		7
7	1	3	9	2	4			6
8	5	9	7			4	2	3
4	2						9	
2		7	4		9		3	
9	6		5	3	7	2	8	4
3		5		8		1		9

New Game

Solution

Hint

Check Solved

Figure 5.1.12.2: Sudoku Game.

- **Memory Match:** The Memory Match game presents a board of cards with matching pairs.
 - **Difficulty:** You can offer different board sizes and numbers of cards for varying difficulty levels.
 - **Scorekeeping:** The game tracks the player's score and moves.

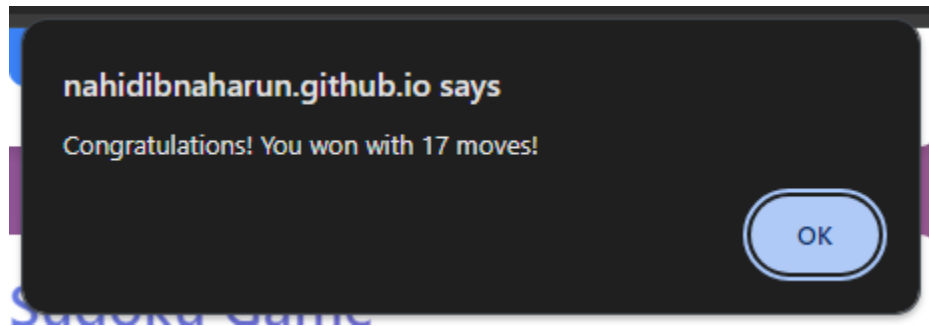


Figure 5.1.12.2: Memory Match Winning Notification.

Memory Match

Choose Difficulty:

Medium (25) ▼

Start Game



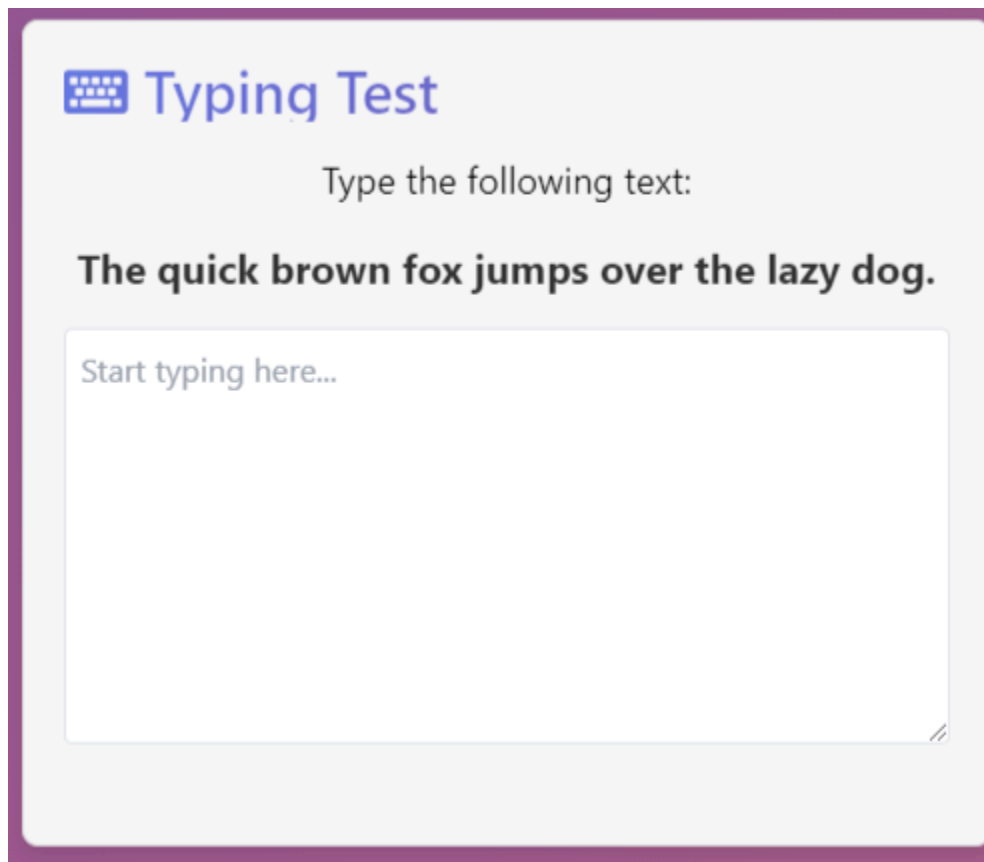
Score: 40

Moves: 17

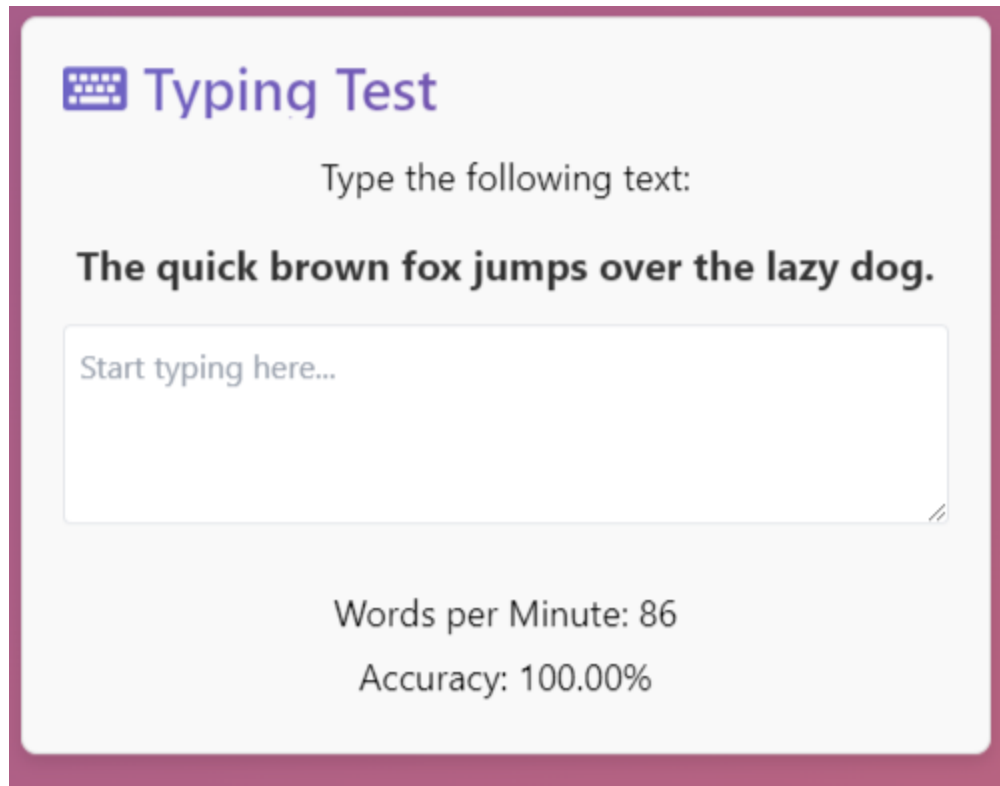
Figure 5.1.12.3: Memory Match .


5.1.13 Typing Test:

- **Text to Type:** The typing test displays a sample paragraph of text for users to type.
- **Time Tracking:** The application tracks the time it takes the user to type the text.
- **Results:** The typing test calculates and displays the user's words per minute (WPM) and accuracy.

A screenshot of a web-based typing test interface. At the top left, there is a blue keyboard icon followed by the title "Typing Test" in a blue sans-serif font. Below the title, the instruction "Type the following text:" is centered in a gray font. Underneath, the target text "The quick brown fox jumps over the lazy dog." is displayed in a bold, black, sans-serif font. Below the target text is a large, white, rounded rectangular text input area with a thin gray border. Inside the input area, the placeholder text "Start typing here..." is visible in a light gray font. The entire interface is enclosed in a light gray rounded rectangle with a thick purple border.

5.1.13a: Typing Test.

A screenshot of a typing test application. At the top left is a purple keyboard icon followed by the title "Typing Test" in purple. Below this, the instruction "Type the following text:" is centered. The text to be typed, "The quick brown fox jumps over the lazy dog.", is displayed in bold black font. A large white text input box with a light gray border contains the placeholder text "Start typing here...". At the bottom, the results "Words per Minute: 86" and "Accuracy: 100.00%" are shown in black text.

 **Typing Test**

Type the following text:

The quick brown fox jumps over the lazy dog.

Start typing here...

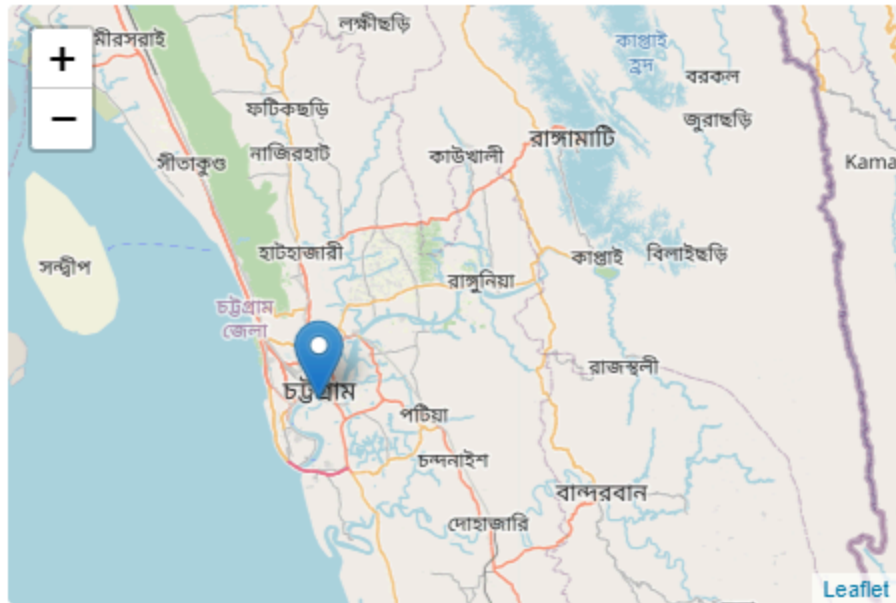
Words per Minute: 86
Accuracy: 100.00%

5.1.13b: Typing Test Result .

5.1.14 Mapping:

- **User Interaction:** Users can click on the map to place markers.
- **Coordinates:** When a marker is placed, the application displays the latitude and longitude coordinates of the selected location.

Location



Latitude: 22.3155

Longitude: 91.8316

Figure 5.1.14: Mapping.

5.1.15 Social:

- Facebook
- Gmail
- Telegram
- Twitter
- Notion
- Discord
- WhatsApp

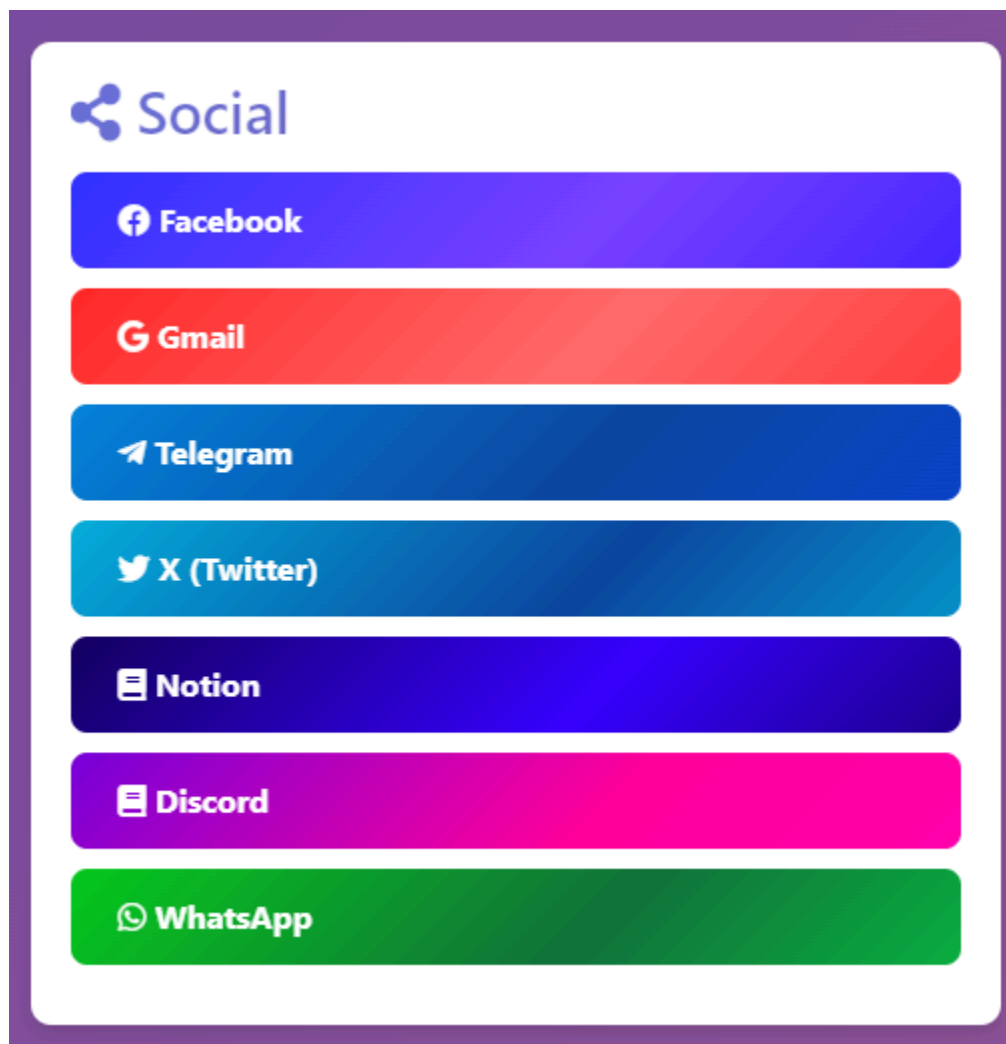


Figure 5.1.15: Social.

5.1.16 Competitive Programming:

- Codeforces
- AtCoder
- LeetCode
- Project Euler
- CodeChef
- HackerRank
- TopCoder



Figure 5.1.16: Competitive Programming.

5.1.17 AI Assistant:

- Gemini
- Ai Studio

- Chatgpt
- Perplexity
- Claude

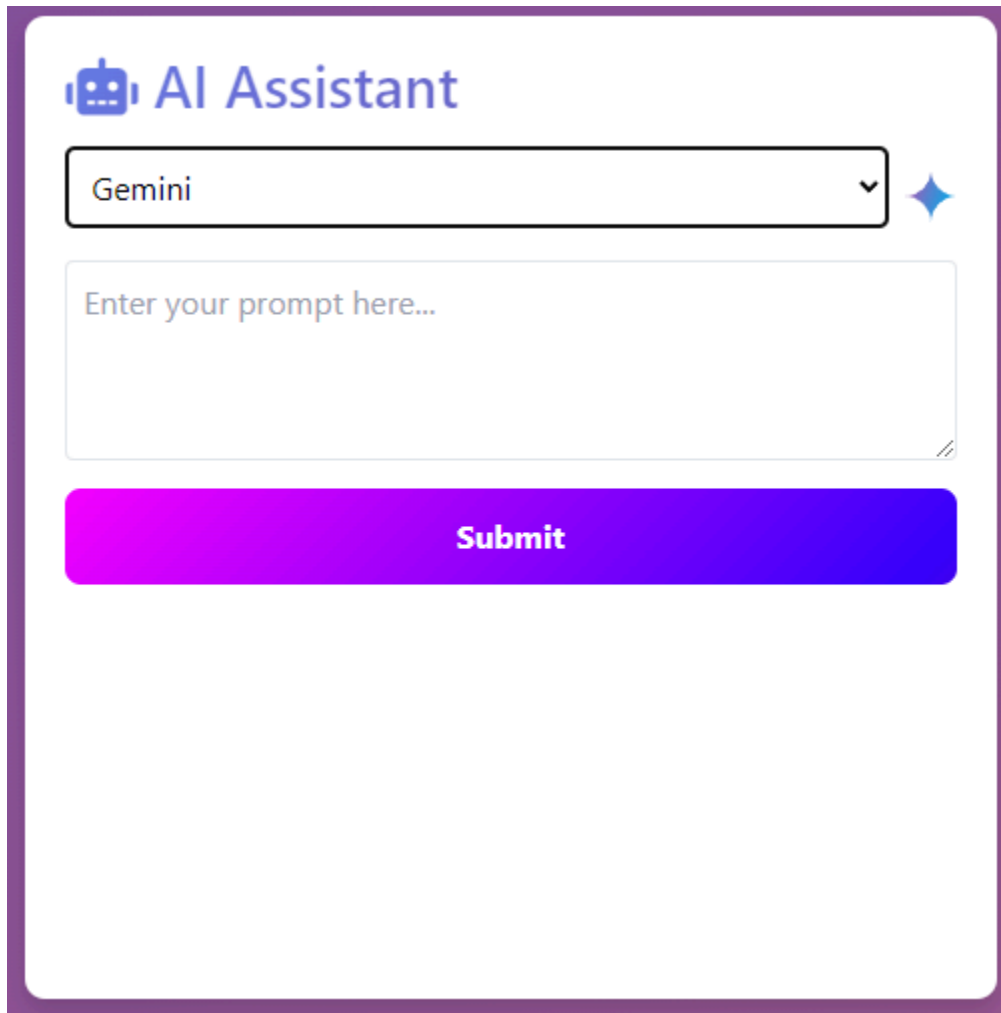
A mockup of an AI Assistant interface. It features a purple border. At the top left is a blue robot icon followed by the text "AI Assistant" in blue. Below this is a white dropdown menu with a black border containing the text "Gemini" and a small downward arrow. To the right of the dropdown is a blue four-pointed star icon. Below the dropdown is a large white text input area with a light gray border and the placeholder text "Enter your prompt here...". At the bottom is a wide, rounded rectangular button with a purple-to-blue gradient and the word "Submit" in white text.

Figure 5.1.17: AI Assistant.

5.2 User Interface (UI) and User Experience (UX):

Signup Page:

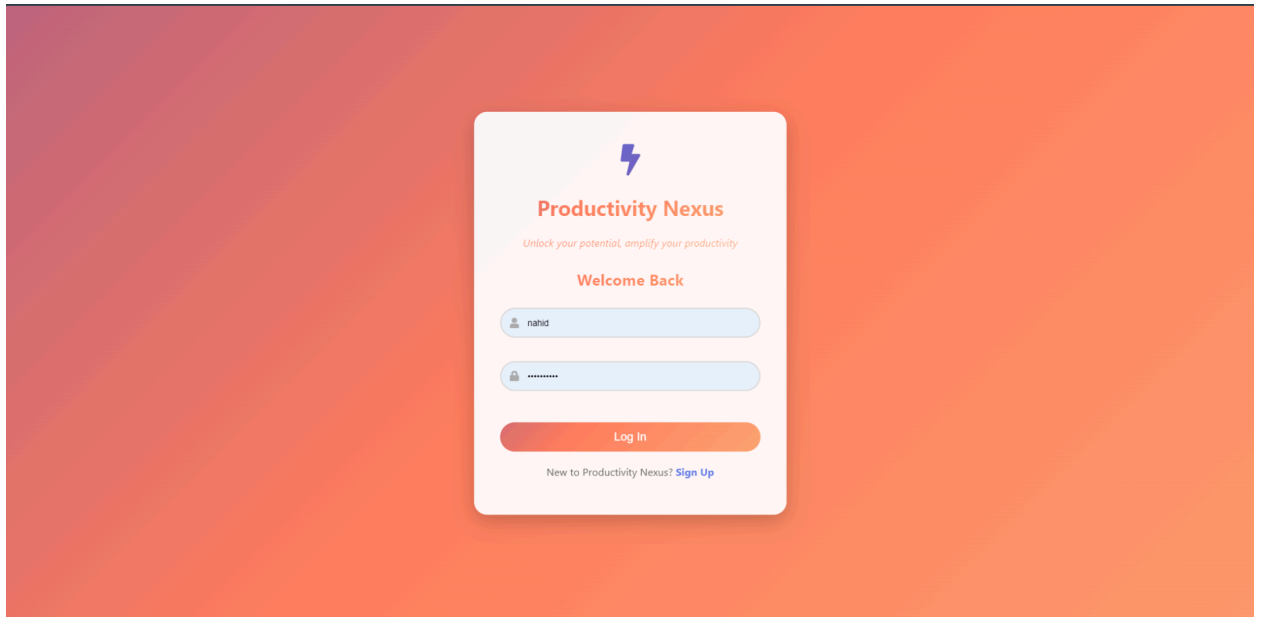


Figure 5.2a: Signup Page.

Dashboard Page:

Figure 5.2b: Dashboard Page.

- **Color Scheme:** Explain your choice of colors (consistent with the overall theme, good contrast for readability, and pleasing to the eye).
- **Layout:** Describe the layout of the dashboard (organized, easy to navigate, logically grouped features).
- **Typography:** Mention the fonts you used (clear, legible, consistent with the brand).
- **Accessibility:** Discuss any steps taken to make the UI accessible (alt text for images, keyboard navigation).
- **User Testing:** Mention any user testing you conducted to gather feedback on UI/UX and how you incorporated those suggestions.

5.3 Technology Stack:

- **HTML:** 5.2
- **CSS:** 4.0 (Tailwind CSS)
- **JavaScript:** ES6
- **Backend Language:** PHP 7.4
- **Database:** MySQL 5.7
- **API:** OpenWeatherMap API (for weather & air pollution), World Map, Foreign Exchange and currency converter.
- **Libraries/Frameworks:** Any additional libraries or frameworks you used (e.g., for form validation, charts, etc.).

6. Results & Analysis

6.1 Evaluation:

- **User Testing:** Describe how you conducted user testing (e.g., recruiting participants, tasks they performed, feedback collected).

- **Self-Evaluation:** Explain how you tested the application's functionality and usability yourself.
- **Comparison to Other Tools:** Did you compare the Productivity Nexus to other productivity tools? Discuss your findings.

6.2 Key Findings:

- **Strengths:** Highlight the strengths of your application based on the evaluation.
 - Did it meet the objectives you set out?
 - What did users like about the app?
- **Weaknesses:** Discuss any weaknesses or areas where improvements can be made.
 - What challenges did you encounter?
 - What user feedback was less positive?

6.3 Data:

- **Usage Statistics:** Include any data you collected about application usage (e.g., number of users, frequency of use, most popular features).
- **User Feedback:** Share any qualitative feedback you collected from users (quotes, general impressions).

7. Conclusion:

7.1 Summary of Contributions:

Summarize the key contributions of your Productivity Nexus project. Highlight the following:

- **Functionality:** What specific features did you develop and implement?
- **Design:** What were your key design decisions for the UI and UX?
- **Innovation:** Did your project introduce any new or innovative approaches to productivity?

7.2 Future Work:

Discuss potential areas for future development and improvement of your project:

- **New Features:** What new features could you add to enhance the user experience?
- **UI Enhancements:** How could you further refine the UI to make it more user-friendly or visually appealing?
- **Mobile Responsiveness:** Consider making the application more responsive for mobile devices.
- **Data Visualization:** Could you improve data visualization with charts or graphs to make the productivity score more engaging?

9.Source:

- Source Codes in Github : [Nahid_Ibna_Harun](#)
- Live Link of Application : [Productivity_Nexus](#)

8.API:

- OpenWeatherMap:
 1. Current Weather : [Current Weather](#)

2.3-hour Forecast 5 days: [3-hour Forecast 5 days](#)

3. Air Pollution API : [Air Pollution API](#)

4.Documentation : [OpenWeatherMap](#)

- Foreign Exchange and currency converter : [Foreign Exchange](#)

1.Documentation : [Fixer.com](#)

- World Map : [Leaflet](#)

1.Documentation : [Leaflet_Documentation](#)

9.Code :

1.Admin.css :

```
/* --- General Styles & Background Animation --- */
body {
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  background-size: 400% 400%;
  animation: gradientAnimation 30s ease infinite; /* Slower animation */
  color: white; /* Set text color to white */
  overflow-x: hidden; /* Prevent horizontal scrollbar */
}

@keyframes gradientAnimation {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
  100% {
    background-position: 0% 50%;
  }
}
```

```
}  
}
```

```
/* --- Main Container (No Background) --- */  
.container {  
  background: none;  
  width: 90%; /* Adjust width as needed */  
  margin: 0 auto;  
  max-width: 1200px; /* Limit maximum width */  
}
```

```
/* --- NEW STYLES FOR ADMIN DASHBOARD --- */  
.logout-button {  
  background-image: linear-gradient(135deg, #ff512f, #dd2476, #ffe607); /*  
Orange gradient */  
  background-size: 200% 100%;  
  animation: buttonGradientAnimation 4s linear infinite;  
  transition: transform 0.3s ease, background-color 0.3s ease;  
  padding: 12px 24px; /* Slightly larger padding */  
  border: none;  
  border-radius: 8px; /* Slightly more rounded corners */  
  color: white;  
  cursor: pointer;  
  font-weight: bold;  
  margin-bottom: 10px;  
}
```

```
.logout-button:hover {  
  transform: scale(1.05);  
}
```

```
.widget {  
  padding: 30px;  
  margin-bottom: 30px;  
}
```

```
#usersTable {
```

```
border-collapse: collapse;
width: 100%;
margin-bottom: 20px;
}

#usersTable th,
#usersTable td {
    text-align: left;
    padding: 8px;
    border-bottom: 1px solid #ddd;
}
```

```
#usersTable th {
    background-color: #f0f0f0;
    color: #333;
    font-weight: bold;
}
```

```
.bg-blue-500 {
    background-image: linear-gradient(135deg, #009245, #fcee21, #03f303);
    background-size: 200% 100%;
    animation: buttonGradientAnimation 4s linear infinite;
    transition: transform 0.3s ease, background-color 0.3s ease;
    padding: 12px 24px;
    border: none;
    border-radius: 8px;
    color: white;
    cursor: pointer;
    font-weight: bold;
    margin-bottom: 10px;
}
```

```
.bg-blue-500:hover {
    transform: scale(1.05);
}
```

2.Admin.html:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Admin Dashboard</title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min
.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.c
ss">
  <link rel="stylesheet" href="admin.css">
</head>
<body>
  <div class="container mx-auto px-4 py-8">
    <div class="flex items-center justify-between mb-8">
      <h1 class="text-3xl font-bold text-white">Admin Dashboard</h1>
      <button onclick="logout()" class="bg-red-500 hover:bg-red-700
text-white font-bold py-2 px-4 rounded">Logout</button>
    </div>

    <!-- User Management Section -->
    <div class="widget p-6 bg-white rounded-lg shadow-md">
      <h2 class="text-2xl font-semibold mb-4 text-gray-800"><i class="fas
fa-users mr-2"></i>User Management</h2>
      <table id="usersTable" class="w-full table-auto text-gray-700">
        <thead>
          <tr>
            <th class="px-4 py-3">ID</th>
            <th class="px-4 py-3">Name</th>
            <th class="px-4 py-3">Username</th>
            <th class="px-4 py-3">Email</th>
            <th class="px-4 py-3">Password</th>
            <th class="px-4 py-3">Actions</th>
          </tr>
        </thead>

```

```

        <tbody></tbody>
    </table>

    <button onclick="createUser()" class="bg-blue-500 hover:bg-blue-700
text-white font-bold py-2 px-4 rounded">
        Create User
    </button>
</div>

</div>

<script src="admin.js"></script>
</body>
</html>

```

3.Admin.js:

```

// Check if admin is logged in
function checkAdminLogin() {
    if (sessionStorage.getItem('admin_logged_in') !== 'true') {
        alert("Please log in as admin first.");
        window.location.href = 'index.html';
    }
}

checkAdminLogin(); // Call this function when the admin page loads

// Function to display users in the table
function displayUsers() {
    const xhr = new XMLHttpRequest();
    xhr.open("GET", "fetch_users.php", true);
    xhr.onload = function() {
        if (xhr.status >= 200 && xhr.status < 400) {
            const users = JSON.parse(xhr.responseText);
            const usersTable =
document.getElementById('usersTable').getElementsByTagName('tbody')[0];
            usersTable.innerHTML = ''; // Clear existing data

```

```

for (let i = 0; i < users.length; i++) {
    const user = users[i];
    const row = usersTable.insertRow();
    const cell1 = row.insertCell();
    const cell2 = row.insertCell();
    const cell3 = row.insertCell();
    const cell4 = row.insertCell();
    const cell5 = row.insertCell();
    const cell6 = row.insertCell();

    cell1.innerHTML = user.id;
    cell2.innerHTML = user.name;
    cell3.innerHTML = user.username;
    cell4.innerHTML = user.email;
    cell5.innerHTML = user.password;
    cell6.innerHTML = `<button onclick="deleteUser(${user.id})"
class="bg-red-500 hover:bg-red-700 text-white font-bold py-2 px-4
rounded">Delete</button>`;
}
} else {
    console.error('Request failed. Returned status of ' + xhr.status);
}
};
xhr.send();
}

```

// Function to delete a user

```

function deleteUser(userId) {
    if (confirm("Are you sure you want to delete this user?")) {
        const xhr = new XMLHttpRequest();
        xhr.open("POST", "delete_user.php", true);
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

        xhr.onload = function() {
            if (xhr.status >= 200 && xhr.status < 400) {
                alert(xhr.responseText);
            }
        };
        xhr.send();
    }
}

```

```

        displayUsers(); // Update the user list
    } else {
        console.error('Request failed. Returned status of ' + xhr.status);
    }
};
xhr.send("userId=" + userId);
}
}

// Function to create a new user
function createUser() {
    const name = prompt("Enter the user's name:");
    const username = prompt("Enter the user's username:");
    const email = prompt("Enter the user's email:");
    const password = prompt("Enter the user's password:");

    if (name && username && email && password) {
        const xhr = new XMLHttpRequest();
        xhr.open("POST", "create_user.php", true);
        xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

        xhr.onload = function() {
            if (xhr.status >= 200 && xhr.status < 400) {
                alert(xhr.responseText);
                displayUsers(); // Update the user list
            } else {
                console.error('Request failed. Returned status of ' + xhr.status);
            }
        };
        xhr.send("name=" + name + "&username=" + username + "&email=" +
email + "&password=" + password);
    }
}

// Function to log out the admin
function logout() {

```



```
// Clear the admin_logged_in session storage variable
sessionStorage.removeItem('admin_logged_in');
window.location.href = 'index.html';
}
```

```
// Display users when the page loads
displayUsers();
```

4.Admin_login.php:

```
<?php
// Include the database connection
include 'db_config.php';

// Check if the admin login form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $adminUsername = $_POST["adminUsername"];
    $adminPassword = $_POST["adminPassword"];

    // Sanitize input (important for security)
    $adminUsername = mysqli_real_escape_string($conn, $adminUsername);
    $adminPassword = mysqli_real_escape_string($conn, $adminPassword);

    // Query the database
    $sql = "SELECT * FROM users WHERE username = '$adminUsername' AND
password = '$adminPassword' AND role = 'admin'";
    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) == 1) {
        // Admin found, redirect to the admin dashboard
        echo "Admin login successful!";
    } else {
        // Incorrect username or password, or not an admin
        echo "Invalid admin username or password.";
    }
}
?>
```

5.Create_user.php:

```
<?php
// Include the database connection
include 'db_config.php';

// Check if the admin login form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $adminUsername = $_POST["adminUsername"];
    $adminPassword = $_POST["adminPassword"];

    // Sanitize input (important for security)
    $adminUsername = mysqli_real_escape_string($conn, $adminUsername);
    $adminPassword = mysqli_real_escape_string($conn, $adminPassword);

    // Query the database
    $sql = "SELECT * FROM users WHERE username = '$adminUsername' AND
password = '$adminPassword' AND role = 'admin'";
    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) == 1) {
        // Admin found, redirect to the admin dashboard
        echo "Admin login successful!";
    } else {
        // Incorrect username or password, or not an admin
        echo "Invalid admin username or password.";
    }
}
?>
```

6.Dashboard.css:

```
/* --- General Styles & Background Animation --- */
body {
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
    background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
    background-size: 400% 400%;
    animation: gradientAnimation 30s ease infinite;
```

```
}
```

```
@keyframes gradientAnimation {  
  0% {  
    background-position: 0% 50%;  
  }  
  50% {  
    background-position: 100% 50%;  
  }  
  100% {  
    background-position: 0% 50%;  
  }  
}
```

```
/* --- Main Container (No Background) --- */  
.container {  
  background: none;  
}
```

```
/* --- Gradient Header --- */  
.gradient-bg {  
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);  
  background-size: 400% 400%;  
  animation: gradientAnimation 30s ease infinite;  
  color: white;  
  padding: 20px 40px;  
  border-radius: 10px 10px 0 0;  
  text-align: center;  
  position: relative;  
}
```

```
.gradient-bg h1 {  
  font-size: 2.2em;  
  font-weight: bold;  
}
```

```
.gradient-bg #motivationalQuote {
```

```
font-size: 1.1em;
font-weight: bold;
}

/* --- Widget Styles --- */
.widget {
background-color: white;
border-radius: 10px;
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
transition: all 0.3s ease;
padding: 20px;
margin-bottom: 20px;
}

.widget:hover {
transform: translateY(-5px);
box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
}

/*Breathing Circle*/
.breathing-circle {
width: 100px;
height: 100px;
border-radius: 50%;
background-color: #4299e1;
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
transition: all 4s ease-in-out;
}

.breathing-circle.inhale {
width: 150px;
height: 150px;
}
```

```
.breathing-circle.hold {  
  width: 150px;  
  height: 150px;  
}
```

```
.breathing-circle.exhale {  
  width: 100px;  
  height: 100px;  
}
```

```
#breathingExercise button {  
  background-image: linear-gradient(135deg, #4299e1, #667eea, #764ba2);  
  background-size: 200% 100%;  
  animation: buttonGradientAnimation 6s linear infinite;  
  transition: transform 0.3s ease, background-color 0.3s ease;  
}
```

```
#breathingExercise button:hover {  
  transform: scale(1.05);  
}
```

```
/* Notes Widget Styles */
```

```
.widget:nth-child(8) {  
  /* Adjust the nth-child selector if your Notes widget is in a different position */  
  background-color: white;  
  border-radius: 10px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  transition: all 0.3s ease;  
  padding: 20px;  
  margin-bottom: 20px;  
}
```

```
.widget:nth-child(8):hover {  
  transform: translateY(-5px);  
  box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);  
}
```

```
.widget:nth-child(8) h2 {
  font-size: 1.8em;
  color: transparent;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-size: 400% 400%;
  animation: gradientAnimation 30s ease infinite;
  margin-bottom: 15px;
}
```

```
.widget:nth-child(8) textarea {
  /* Style the textarea for notes */
  resize: vertical;
  min-height: 100px;
}
```

```
.widget:nth-child(8) button.save-notes-button {
  background-image: linear-gradient(
    135deg,
    #2e3192,
    #1bffff,
    #9c27b0
  ); /* New gradient colors */
  background-size: 200% 100%;
  animation: buttonGradientAnimation 6s linear infinite;
}
```

```
/* --- Widget Titles --- */
```

```
.widget h2 {
  font-size: 1.8em;
  color: transparent;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-size: 400% 400%;
  animation: gradientAnimation 30s ease infinite;
  margin-bottom: 15px;
}
```

```
}
```

```
/* --- All Buttons (Common Styles) --- */
```

```
.timer-controls button,
```

```
.widget button,
```

```
.flex button,
```

```
.social-links a,
```

```
li button,
```

```
.logout-button {
```

```
padding: 12px 24px; /* Slightly larger padding */
```

```
border: none;
```

```
border-radius: 8px;
```

```
color: white;
```

```
cursor: pointer;
```

```
font-weight: bold;
```

```
margin-bottom: 10px;
```

```
background-size: 200% 100%;
```

```
animation: buttonGradientAnimation 6s linear infinite; /* Faster animation
```

```
*/
```

```
transition: transform 0.3s ease, background-color 0.3s ease;
```

```
}
```

```
.timer-controls button:hover,
```

```
.widget button:hover,
```

```
.flex button:hover,
```

```
.social-links a:hover,
```

```
li button:hover,
```

```
.logout-button:hover {
```

```
transform: scale(1.05);
```

```
}
```

```
@keyframes buttonGradientAnimation {
```

```
0% {
```

```
background-position: 0% 50%;
```

```
}
```

```
50% {
```

```
background-position: 100% 50%;
```

```

}
100% {
  background-position: 0% 50%;
}
}

/* --- Specific Button Gradients (3-color) --- */
.timer-controls button {
  background-image: linear-gradient(135deg, #1f23f0, #5413ec, #14f7eb);
}

.timer-controls .stop-button {
  background-image: linear-gradient(135deg, #00e1ff, #8e04ff, #3700ff);
}

.flex.justify-center button {
  background-image: linear-gradient(135deg, #d914fc, #ff8800, #a807ff);
}

.widget:nth-child(4) button {
  background-image: linear-gradient(135deg, #f86464, #ff1414, #0073ff);
}

.widget:nth-child(2) button {
  background-image: linear-gradient(135deg, #009245, #fcee21, #03f303);
}

.widget:nth-child(7) button {
  background-image: linear-gradient(135deg, #5900ff, #05ffd5, #b13ddf);
}

.widget:nth-child(3) button {
  background-image: linear-gradient(135deg, #ff512f, #dd2476, #ffe607);
}

/* --- Widget:nth-child(8) Button (Save Notes button) --- */
.widget:nth-child(8) button.save-notes-button {

```



```
background-image: linear-gradient(
  135deg,
  #2e3192,
  #1bffff,
  #9c27b0
); /* New gradient colors */
background-size: 200% 100%; /* Same as other animated buttons */
animation: buttonGradientAnimation 6s linear infinite; /* Same animation as
other buttons */
}
```

```
/* --- Social Links with Animated Gradients --- */
```

```
.social-links a {
  display: block;
  color: white;
  text-decoration: none;
}
```

```
.social-links .facebook {
  background-image: linear-gradient(135deg, #052aff, #4c00ffbd, #0206ff);
}
```

```
.social-links .gmail {
  background-image: linear-gradient(135deg, #ff0000, #ff000093, #ff0505);
}
```

```
.social-links .telegram {
  background-image: linear-gradient(135deg, #00aaff, #0d47a1, #0a43ff);
}
```

```
.social-links .twitter {
  background-image: linear-gradient(135deg, #00eeff, #0d47a1, #00f7ff);
}
```

```
.social-links .notion {
  background-image: linear-gradient(135deg, #000000, #3801ff, #000000);
}
```

```
.social-links .whatsapp {
  background-image: linear-gradient(135deg, #00ff0d, #13733b, #00ff4c);
}
```

```
.social-links .discord {
```

```
background-image: linear-gradient(135deg, #2600ff, #ff009d, #ff00bf);
}
```

```
/* --- Logout Button --- */
.logout-button {
  position: absolute;
  top: 10px;
  right: 10px;
  background-image: linear-gradient(135deg, #0004ff, #4c00ff, #feb47b,
  #ff4e22);
}
```

```
/* --- Progress Bars with Background Animation --- */
.progress {
  height: 10px;
  border-radius: 9999px;
  overflow: hidden;
  position: relative;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  background-size: 400% 400%;
  animation: progressBackgroundAnimation 30s ease infinite; /* Animation
  slowed down */
}
```

```
.progress > div {
  height: 100%;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  background-size: 400% 400%;
  animation: gradientFill 1s ease-out;
  transition: width 0.5s ease-in-out;
}
```

```
@keyframes gradientFill {
  0% {
    background-position: 0% 50%;
  }
  100% {
```

```
    background-position: 100% 50%;  
  }  
}
```

```
@keyframes progressBackgroundAnimation {  
  0% {  
    background-position: 0% 50%;  
  }  
  50% {  
    background-position: 100% 50%;  
  }  
  100% {  
    background-position: 0% 50%;  
  }  
}
```

```
/* --- Align Set Timer button --- */  
.flex.justify-center button {  
  margin-left: 10px;  
}
```

```
/* --- Original Delete Button Styles (from Tailwind CSS) --- */  
li .delete-button,  
.flex .text-red-500 {  
  /* Use default Tailwind styles for delete buttons */  
}
```

```
/* Calculator Widget Styles */  
#calculator {  
  background-color: white;  
  border-radius: 10px;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  transition: all 0.3s ease;  
  padding: 20px;  
  margin-bottom: 20px;
```

```

text-align: center; /* Center align the content */
}

#calculator:hover {
  transform: translateY(-5px);
  box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
}

#calculator h2 {
  font-size: 1.8em;
  color: transparent;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-size: 400% 400%;
  animation: gradientAnimation 30s ease infinite; /* Slower animation */
  margin-bottom: 15px;
}

#display {
  font-size: 2em;
  background-color: white; /* White background for the input */
  color: black; /* Set text color to black */
  text-align: right; /* Align the text to the right */
  padding: 10px;
  border: 1px solid #ccc;
  width: 90%;
  margin: 0 auto 10px auto;
  border-radius: 5px;
}

/* Cool Calculator Button Styles (Large and Rectangular) */
.calc-button {
  padding: 16px 36px; /* Adjusted padding for rectangle shape */
  border: none;
  border-radius: 10px; /* More rounded corners */
  color: white;

```

```
cursor: pointer;
font-weight: bold;
font-size: 1.4em; /* Larger font size */
background-size: 200% 100%;
animation: buttonGradientAnimation 6s linear infinite;
transition: transform 0.3s ease, background-color 0.3s ease;
width: calc(100% / 2 - 22px); /* Adjust for 2 columns */ /* Changed width
for 2 columns */
display: flex; /* Make the button a flex container */
justify-content: center; /* Center the content horizontally */
align-items: center; /* Center the content vertically */
}
```

```
.calc-button:hover {
  transform: scale(1.1); /* Bigger on hover */
  box-shadow: 0 6px 12px rgba(0, 0, 0, 0.3); /* Stronger hover shadow */
}
```

/* Specific Colors for Calculator Buttons */

```
.calc-button:nth-child(1),
.calc-button:nth-child(2),
.calc-button:nth-child(3),
.calc-button:nth-child(5),
.calc-button:nth-child(6),
.calc-button:nth-child(7),
.calc-button:nth-child(9),
.calc-button:nth-child(10),
.calc-button:nth-child(11),
.calc-button:nth-child(13),
.calc-button:nth-child(14) {
  background-image: linear-gradient(135deg, #004dff, #00ff9d, #00ff9d); /*
Number button gradient (Blue-Green) */
}
```

```
.calc-button:nth-child(4),
.calc-button:nth-child(8),
.calc-button:nth-child(12),
```

```
.calc-button:nth-child(16),
.calc-button:nth-child(17) { /* Clear button */
  background-image: linear-gradient(135deg, #ff512f, #dd2476, #ffe607); /*
Operator button gradient (Red-Orange) */
}
```

```
.calc-button:nth-child(15) {
  background-image: linear-gradient(135deg, #009245, #fcee21, #03f303); /*
Equal button gradient (Green) */
}
```

```
/* --- Animations --- */
@keyframes buttonGradientAnimation {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
  100% {
    background-position: 0% 50%;
  }
}
```

```
@keyframes gradientAnimation {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
  100% {
    background-position: 0% 50%;
  }
}
```

```

/* Book Tracker Widget Styles */
.book-tracker {
  background-color: white;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  transition: all 0.3s ease;
  padding: 20px;
  margin-bottom: 20px;
}

.book-tracker:hover {
  transform: translateY(-5px);
  box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
}

.book-tracker h2 {
  font-size: 1.8em;
  color: transparent;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-size: 400% 400%;
  animation: gradientAnimation 30s ease infinite; /* Slower animation */
  margin-bottom: 15px;
}

.book-tracker input[type="text"],
.book-tracker input[type="datetime-local"] {
  margin-bottom: 10px;
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

.book-tracker ul {
  list-style: none;
  padding: 0;

```

```
}
```

```
.book-tracker li {  
  padding: 5px;  
  border-bottom: 1px solid #eee;  
}
```

```
.book-tracker button {  
  padding: 12px 24px;  
  border: none;  
  border-radius: 8px;  
  color: white;  
  cursor: pointer;  
  font-weight: bold;  
  background-image: linear-gradient(  
    135deg,  
    #009245,  
    #fcee21,  
    #03f303  
  ); /* Book tracker button gradient */  
  background-size: 200% 100%;  
  animation: buttonGradientAnimation 6s linear infinite;  
  transition: transform 0.3s ease, background-color 0.3s ease;  
}
```

```
.book-tracker button:hover {  
  transform: scale(1.05);  
}
```

```
/* Animation for the gradient */  
@keyframes gradientAnimation {  
  0% {  
    background-position: 0% 50%;  
  }  
  50% {  
    background-position: 100% 50%;  
  }  
}
```



```
100% {  
    background-position: 0% 50%;  
}  
}
```

```
/* Animation for the button gradient */  
@keyframes buttonGradientAnimation {  
    0% {  
        background-position: 0% 50%;  
    }  
    50% {  
        background-position: 100% 50%;  
    }  
    100% {  
        background-position: 0% 50%;  
    }  
}  
/* ... (other styles) ... */
```

```
/* Breathing Circle */  
.breathing-container {  
    width: 150px;  
    height: 150px;  
    position: relative; /* Set the parent to relative */  
}
```

```
.breathing-circle {  
    width: 100px;  
    height: 100px;  
    border-radius: 50%;  
    position: absolute; /* Keep circle absolute */  
    top: 50%;  
    left: 50%;  
    transform: translate(-50%, -50%);  
    transition: all 4s ease-in-out;  
    background-image: linear-gradient(135deg, #0400ff, #8400ff, #ff0000);  
    background-size: 400% 400%;
```

```
}
```

```
.breathing-circle.inhale {  
  width: 150px;  
  height: 150px;  
}
```

```
.breathing-circle.hold {  
  width: 150px;  
  height: 150px;  
}
```

```
.breathing-circle.exhale {  
  width: 100px;  
  height: 100px;  
}
```

```
#breathingExercise button {  
  background-image: linear-gradient(135deg, #00ff15, #8400ff, #ff00ff);  
  background-size: 200% 100%;  
  animation: buttonGradientAnimation 6s linear infinite;  
  transition: transform 0.3s ease, background-color 0.3s ease;  
}
```

```
#breathingExercise button:hover {  
  transform: scale(1.05);  
}
```

```
/* Animation for the breathing circle gradient */  
@keyframes breathingCircleGradient {  
  0% {  
    background-position: 0% 50%;  
  }  
  50% {  
    background-position: 100% 50%;  
  }  
  100% {
```

```

    background-position: 0% 50%;
  }
}

/* ... (other styles) ... */

/* Notes Widget Styles */
#notesWidget {
  background-color: white;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
  transition: all 0.3s ease;
  padding: 20px;
  margin-bottom: 20px;
}

#notesWidget:hover {
  transform: translateY(-5px);
  box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
}

#notesWidget h2 {
  font-size: 1.8em;
  color: transparent;
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-size: 400% 400%;
  animation: gradientAnimation 30s ease infinite; /* Slower animation */
  margin-bottom: 15px;
}

#notesWidget textarea {
  /* Style the textarea for notes */
  resize: vertical; /* Allow vertical resizing */
  min-height: 100px; /* Set a minimum height for the textarea */
}

```

```
#notesWidget button.save-notes-button {
padding: 12px 24px;
border: none;
border-radius: 8px;
color: white;
cursor: pointer;
font-weight: bold;
background-image: linear-gradient(
    135deg,
    #2e3192,
    #1bffff,
    #9c27b0
); /* New gradient colors */
background-size: 200% 100%; /* Same as other animated buttons */
animation: buttonGradientAnimation 6s linear infinite; /* Same animation as
other buttons */
transition: transform 0.3s ease, background-color 0.3s ease;
}
```

```
#notesWidget button.save-notes-button:hover {
transform: scale(1.05);
}
```

```
/* Animation for the gradient */
@keyframes gradientAnimation {
0% {
background-position: 0% 50%;
}
50% {
background-position: 100% 50%;
}
100% {
background-position: 0% 50%;
}
}
```

```

/* Animation for the button gradient */
@keyframes buttonGradientAnimation {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
  100% {
    background-position: 0% 50%;
  }
}
/* box game*/
.game-container {
  display: grid;
  grid-template-columns: repeat(4, 1fr); /* 4 columns */
  grid-gap: 10px;
  width: 300px;
  margin: 20px auto;
}

.card {
  width: 70px;
  height: 70px;
  background-color: #ddd;
  border-radius: 10px;
  display: flex;
  justify-content: center;
  align-items: center;
  cursor: pointer;
  transition: transform 0.2s;
  position: relative; /* For the back face */
}

.card.flipped {
  transform: rotateY(180deg);
}

```

```
.card.matched {
  background-image: linear-gradient(135deg, #f800e3, #23ff1b, #d900ff);
  pointer-events: none; /* Disable clicks on matched cards */
}

.card-content {
  font-size: 2em;
  color: #333;
  transform: rotateY(180deg); /* Initially hidden */
  transition: transform 0.2s;
  position: absolute; /* For the front face */
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  backface-visibility: hidden; /* Hide the back face when flipped */
}

.card.flipped .card-content {
  transform: rotateY(0deg); /* Reveal when flipped */
}

.score-container {
  text-align: center;
  margin-top: 10px;
}

.level-select {
  text-align: center;
  margin-bottom: 20px;
}

.level-select label {
  display: block;
```

```
margin-bottom: 5px;
}
```

```
.level-select select {
padding: 5px;
margin-bottom: 10px;
border: 1px solid #ccc;
border-radius: 5px;
}
```

```
.level-select button {
padding: 8px 15px;
background-image: linear-gradient(135deg, #0011ff, #ff1b1b, #d900ff);
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
}
```

```
/* ... (your existing CSS) ... */
```

```
/* --- Competitive Programming Links --- */
.social-links .codeforces {
background-image: linear-gradient(
135deg,
#fbff00,
#0011ff,
#ff0000
); /* Blue-ish theme */
}
```

```
.social-links .atcoder {
background-image: linear-gradient(
135deg,
#000000,
#99878d,
#ffffff
}
```

```
); /* Pinkish theme */  
}
```

```
.social-links .leetcode {  
  background-image: linear-gradient(  
    135deg,  
    #000000,  
    #754eb3,  
    #929292  
  ); /* Blue theme */  
}
```

```
.social-links .projecteuler {  
  background-image: linear-gradient(  
    135deg,  
    #ff8800,  
    #5e5e5e,  
    #000000  
  ); /* Dark gray theme */  
}
```

```
.social-links .codechef {  
  background-image: linear-gradient(  
    135deg,  
    #ffffff,  
    #964b00,  
    #000000  
  ); /* Orange-brown theme */  
}
```

```
.social-links .hackerrank {  
  background-image: linear-gradient(  
    135deg,  
    #00ff22,  
    #f3f3f3,  
    #000000  
  ); /* Red-orange theme */
```



```
}
```

```
.social-links .topcoder {  
  background-image: linear-gradient(  
    135deg,  
    #0370ff,  
    #01ff01,  
    #ffef13,  
    #ff0000  
  ); /* Green theme */  
}  
/* Sudoku Styles */  
#sudoku-board {  
  display: grid;  
  grid-template-columns: repeat(9, 1fr);  
  gap: 2px;  
  max-width: 360px;  
  margin: auto;  
  padding: 10px;  
  border-radius: 10px;  
  background-color: #fafafa;  
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);  
  border: 2px solid #333; /* Overall border */  
}
```

```
#sudoku-board .cell {  
  width: 35px;  
  height: 35px;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  background-color: #f5f5f5;  
  border: 1px solid #999;  
  font-size: 18px;  
  font-weight: bold;  
  border-radius: 5px;  
  transition: background-color 0.3s ease;
```

```
}
```

```
#sudoku-board .cell.selected {  
  background-color: #d1e7dd;  
}
```

```
#sudoku-board .cell.initial {  
  background-color: #e0e0e0;  
  color: #555;  
}
```

```
#sudoku-board .cell.error {  
  background-color: #f8d7da;  
  color: #842029;  
}
```

```
#sudoku-board .cell:nth-child(3n+1):not(:first-child):not(:nth-child(10n+1))  
{  
  border-left: 2px solid #333;  
}
```

```
#sudoku-board .cell:nth-child(n+19):nth-child(-n+21),  
#sudoku-board .cell:nth-child(n+37):nth-child(-n+39),  
#sudoku-board .cell:nth-child(n+55):nth-child(-n+57) {  
  border-bottom: 2px solid #333;  
}
```

```
/* High Score and Lives Stylish */
```

```
#high-score, #lives {  
  color: #ff5722;  
  font-size: 1.8em;  
  text-shadow: 2px 2px 3px rgba(0, 0, 0, 0.2);  
  background: linear-gradient(to right, #ffe259, #ffa751);  
  -webkit-background-clip: text;  
  -webkit-text-fill-color: transparent;  
  animation: gradientBackgroundAnimation 3s ease-in-out infinite;  
}
```

```
@keyframes gradientBackgroundAnimation {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
  100% {
    background-position: 0% 50%;
  }
}
```

```
/* Button Styles with Gradient Animations */
#new-game,
#solve,
#hint,
#check-solved {
  background-size: 200% 100%;
  animation: buttonGradientAnimation 4s linear infinite;
  transition: transform 0.3s ease, background-color 0.3s ease;
  padding: 12px 24px; /* Slightly larger padding */
  border: none;
  border-radius: 8px; /* Slightly more rounded corners */
  color: white;
  cursor: pointer;
  font-weight: bold;
  margin-bottom: 10px;
}
```

```
@keyframes buttonGradientAnimation {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
}
```

```

    100% {
      background-position: 0% 50%;
    }
  }

  /* Specific Gradients for Sudoku Buttons */
  #new-game {
    background-image: linear-gradient(135deg, #009245, #fcee21, #03f303); /*
    Green */
  }

  #solve {
    background-image: linear-gradient(135deg, #ff512f, #dd2476, #ffe607); /*
    Orange */
  }

  #hint {
    background-image: linear-gradient(135deg, #f86464, #ff1414, #0073ff); /*
    Red */
  }

  #check-solved {
    background-image: linear-gradient(135deg, #5900ff, #05ffd5, #b13ddf); /*
    Blue */
  }

  #new-game:hover,
  #solve:hover,
  #hint:hover,
  #check-solved:hover {
    transform: scale(1.05);
  }

  /* Winning animation */
  @keyframes winAnimation {
    0% {
      transform: scale(1);

```

```
}
50% {
  transform: scale(1.1);
}
100% {
  transform: scale(1);
}
}
```

```
.sudoku-win {
  animation: winAnimation 0.5s infinite alternate;
}
```

```
/* ... other styles for index.html ... */
.tab-icon {
  width: 30px;
  height: 30px;
  background-image:
url('file:///D:/download/naheed/project_prime/monday-icon-svgrepo-com.svg');
  background-repeat: no-repeat;
  background-size: contain;
  background-position: center;
  display: inline-block;
  margin-right: 8px;
  animation: iconAnimation 2s linear infinite;
}
```

```
@keyframes iconAnimation {
  0% {
    transform: scale(1);
    filter: hue-rotate(0deg);
  }
  50% {
    transform: scale(1.1);
```

```

    filter: hue-rotate(180deg);
  }
  100% {
    transform: scale(1);
    filter: hue-rotate(360deg);
  }
}

/* typing test */

#typingTestWidget {
  background-color: #f9f9f9;
  border: 1px solid #ddd;
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

#textToType {
  color: #333;
  margin: 0;
}

textarea {
  min-height: 100px;
  font-size: 16px;
  resize: vertical;
}

#result p {
  margin: 4px 0;
}

/* ... weather 5 day ... */

#fiveDayForecast ul {
  list-style: none;

```

```
padding: 0;
}

#fiveDayForecast .forecast-item {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 10px;
  border-bottom: 1px solid #ddd;
}

#fiveDayForecast .day {
  font-weight: bold;
  font-size: 1.1em;
}

#fiveDayForecast .forecast-icon {
  width: 40px;
  height: 40px;
}

#fiveDayForecast .temp {
  font-weight: bold;
  font-size: 1.2em;
  color: #2196F3; /* A distinct blue color */
}

/* ... air pollution ... */

#airPollutionWidget ul {
  list-style: none;
  padding: 0;
}

#airPollutionWidget .pollutant-item {
  display: flex;
  justify-content: space-between;
```

```
    align-items: center;
    padding: 5px;
    border-bottom: 1px solid #ddd;
}
```

```
#airPollutionWidget .pollutant-name {
    font-weight: bold;
}
```

```
#airPollutionWidget .pollutant-value {
    font-size: 0.9em;
}
```

```
/* MAP */
```

```
#map-widget {
    border: 1px solid #ddd;
    border-radius: 8px;
    padding: 16px;
    background-color: #f9f9f9;
    margin-bottom: 20px;
}
```

```
#map {
    height: 400px;
    width: 100%;
    border-radius: 4px;
}
```

```
.input-group {
    margin-top: 20px;
}
```

```
input {
    padding: 10px;
    width: 100%;
}
```



```
border: 1px solid #ccc;
border-radius: 4px;
}
```

7. Dashboard.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Improved Productivity Dashboard</title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min
.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min.c
ss">
  <link rel="stylesheet"
href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />
  <link rel="icon" type="image/svg+xml"
href="https://raw.githubusercontent.com/nahidibnaharun/Productivity_Nex
us_2.0/47e576223ef09fbde6ed1faaf3264ddf69891722/monday-icon-svgrepo
-com.svg">
  <link rel="stylesheet" href="dashboard.css">

</head>
<body class="bg-gray-100">
  <!-- Gradient Header with Logout Button -->
  <div class="gradient-bg text-white py-6 px-4 shadow-lg">
    <h1 class="text-3xl font-bold text-center mb-2">Productivity
Dashboard</h1>
    <div id="motivationalQuote" class="text-center italic"></div>
    <button class="logout-button" onclick="logout()">Logout</button>
  </div>

  <!-- Main Container -->
```

```

<div class="container mx-auto px-4 py-8">
  <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">

    <!-- Clock Widget -->
    <div class="widget p-6">
      <h2 class="text-xl font-semibold mb-4"><i class="far fa-clock
mr-2"></i>Clock</h2>
      <div id="clock" class="text-4xl font-bold text-center mb-2"></div>
      <div id="date" class="text-center text-gray-600"></div>
    </div>

    <!-- Tasks Widget (Updated) -->
    <div class="widget p-6">
      <h2 class="text-xl font-semibold mb-4"><i class="fas fa-tasks
mr-2"></i>Tasks</h2>
      <ul id="taskList" class="mb-4"></ul>
      <input type="text" id="newTask" placeholder="New task"
class="w-full p-2 mb-2 border rounded">
      <select id="taskType" class="w-full p-2 mb-2 border rounded">
        <option value="Work">Work</option>
        <option value="School">School</option>
        <option value="Personal">Personal</option>
      </select>
      <select id="taskImportance" class="w-full p-2 mb-2 border
rounded">
        <option value="1">1 (Low)</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5 (High)</option>
      </select>
      <button onclick="addTask()" class="w-full bg-green-500 text-white
p-2 rounded hover:bg-green-600 add-button">Add Task</button>
    </div>

    <!-- Upcoming Events Widget -->

```

```

<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4"><i class="fas
fa-calendar-alt mr-2"></i>Upcoming Events</h2>
  <ul id="eventList" class="mb-4"></ul>
  <input type="text" id="newEvent" placeholder="New event"
class="w-full p-2 mb-2 border rounded">
  <input type="date" id="eventDate" class="w-full p-2 mb-2 border
rounded">
  <button onclick="addEvent()" class="w-full bg-indigo-500
text-white p-2 rounded hover:bg-indigo-600">Add Event</button>
</div>

```

```

<!-- Reminders Widget (Updated) -->
<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4"><i class="far fa-bell
mr-2"></i>Reminders</h2>
  <ul id="reminderList" class="mb-4"></ul>
  <input type="text" id="newReminder" placeholder="New reminder"
class="w-full p-2 mb-2 border rounded">
  <input type="datetime-local" id="reminderTime" class="w-full p-2
mb-2 border rounded">
  <button onclick="addReminder()" class="w-full bg-blue-500
text-white p-2 rounded hover:bg-blue-600">Add Reminder</button>
</div>

```

```

<!-- Productivity Score Widget -->
<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4"><i class="fas fa-chart-line
mr-2"></i>Productivity Score</h2>
  <div class="text-center">
    <div id="productivityScore" class="text-5xl font-bold
mb-2">0%</div>
    <div class="progress-bar">
      <div id="productivityProgress" class="progress" style="width:
0%"></div>
    </div>
  </div>

```

```

</div>

<!-- Focus Timer Widget -->
<div class="widget p-6">
    <h2 class="text-xl font-semibold mb-4"><i class="fas fa-hourglass-half mr-2"></i>Focus Timer</h2>
    <div class="timer-controls">
        <button onclick="startFocusTimer()"
class="start-button">Start</button>
        <button onclick="stopFocusTimer()"
class="stop-button">Stop</button>
    </div>
    <div id="focusTimer" class="text-4xl font-bold text-center
mb-4">25:00</div>
    <div class="flex justify-center">
        <input type="number" id="timerMinutes" value="25" min="1"
class="w-16 border rounded px-2 py-1">
        <button onclick="updateTimer()" class="bg-purple-500 text-white
p-2 rounded hover:bg-purple-600">Set Timer</button>
    </div>
</div>

<!-- Water Intake Tracker Widget -->
<div class="widget p-6">
    <h2 class="text-xl font-semibold mb-4"><i class="fas fa-tint
mr-2"></i>Water Intake Tracker</h2>
    <div id="waterIntake" class="text-3xl font-bold text-center
mb-2">0 / 8 glasses</div>
    <div class="progress-bar mb-4">
        <div id="waterProgress" class="progress" style="width:
0% "></div>
    </div>
    <button onclick="addWater()" class="w-full bg-blue-500 text-white
p-2 rounded hover:bg-blue-600">Add Glass</button>
</div>

<!-- Currency Exchange -->

```

```

<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4"><i class="fas
fa-money-bill-wave mr-2"></i>Currency Converter</h2>

  <div class="mb-2">
    <label for="fromCurrencySelect">From:</label>
    <select id="fromCurrencySelect" class="w-full p-2 border
rounded"></select>
  </div>

  <div class="mb-2">
    <label for="toCurrencySelect">To:</label>
    <select id="toCurrencySelect" class="w-full p-2 border
rounded"></select>
  </div>

  <div class="mb-2">
    <label for="amountInput">Amount:</label>
    <input type="number" id="amountInput" value="1" min="0"
class="w-full p-2 border rounded">
  </div>

  <p>Result: <span id="conversionResult">0.00</span></p>
</div>

```

```

<!-- Weather Widget -->
<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4"><i class="fas
fa-cloud-sun-rain mr-2"></i>Weather</h2>
  <select id="citySelect" class="w-full p-2 mb-4 border rounded"
onchange="fetchWeather()">
    <option value="Dhaka,BD">Dhaka,BD</option>
    <option value="Chittagong,BD">Chittagong,BD</option>
    <option value="Khulna,BD">Khulna,BD</option>
    <option value="Rangpur,BD">Rangpur,BD</option>
    <option value="Bogra,BD">Bogra,BD</option>

```

```

<option value="Mymensingh,BD">Mymensingh,BD</option>
<option value="Netrakona,BD">Netrakona,BD</option>
<option value="Sylhet,BD">Sylhet,BD</option>
<option value="Habiganj,BD">Habiganj,BD</option>
<option value="Rajshahi,BD">Rajshahi,BD</option>
<option value="Tongi,BD">Tongi,BD</option>
<option value="Narsingdi,BD">Narsingdi,BD</option>
<option value="Farīdpur,BD">Farīdpur,BD</option>
<option value="Jessore,BD">Jessore,BD</option>
<option value="Comilla,BD">Comilla,BD</option>
<option value="Feni,BD">Feni,BD</option>
<option value="Bāndarban,BD">Bāndarban,BD</option>
<option value="Cox's Bāzār,BD">Cox's Bāzār,BD</option>
</select>
<div id="weather" class="text-center">
  <p id="weatherLocation" class="text-2xl font-bold"></p>
  <p id="weatherTemp" class="text-3xl font-bold"></p>
  <p id="weatherDescription" class="text-xl"></p>
</div>
</div>

```

```

<!-- 5-Day Weather Forecast Widget -->
<div class="widget p-6" id="fiveDayForecast">
  <h2 class="text-xl font-semibold mb-4"><i class="fas fa-calendar-alt
mr-2"></i>5-Day Forecast</h2>
  <select id="citySelectAirForecast" class="w-full p-2 mb-4 border
rounded" onchange="fetchFiveDayForecast()">
    <option value="Dhaka,BD">Dhaka,BD</option>
    <option value="Chittagong,BD">Chittagong,BD</option>
    <option value="Khulna,BD">Khulna,BD</option>
    <option value="Rangpur,BD">Rangpur,BD</option>
    <option value="Bogra,BD">Bogra,BD</option>
    <option value="Mymensingh,BD">Mymensingh,BD</option>
    <option value="Netrakona,BD">Netrakona,BD</option>
    <option value="Sylhet,BD">Sylhet,BD</option>
    <option value="Habiganj,BD">Habiganj,BD</option>

```

```

<option value="Rajshahi,BD">Rajshahi,BD</option>
<option value="Tongi,BD">Tongi,BD</option>
<option value="Narsingdi,BD">Narsingdi,BD</option>
<option value="Farīdpur,BD">Farīdpur,BD</option>
<option value="Jessore,BD">Jessore,BD</option>
<option value="Comilla,BD">Comilla,BD</option>
<option value="Feni,BD">Feni,BD</option>
<option value="Bāndarban,BD">Bāndarban,BD</option>
<option value="Cox's Bāzār,BD">Cox's Bāzār,BD</option>
</select>
<ul id="forecastList" class="list-none p-0"></ul>
</div>

```

```

<!-- Air Pollution Widget -->
<div class="widget p-6" id="airPollutionWidget">
  <h2 class="text-xl font-semibold mb-4"><i class="fas fa-smog
mr-2"></i>Air Quality</h2>
  <div class="flex items-center mb-2">
    <label for="citySelectAirQuality" class="mr-2">City:</label>
    <select id="citySelectAirQuality" class="w-full p-2 mb-2 border
rounded" onchange="fetchAirPollutionData()">
      <option value="Dhaka,BD">Dhaka,BD</option>
      <option value="Chittagong,BD">Chittagong,BD</option>
      <option value="Khulna,BD">Khulna,BD</option>
      <option value="Rangpur,BD">Rangpur,BD</option>
      <option value="Bogra,BD">Bogra,BD</option>
      <option value="Mymensingh,BD">Mymensingh,BD</option>
      <option value="Netrakona,BD">Netrakona,BD</option>
      <option value="Sylhet,BD">Sylhet,BD</option>
      <option value="Habiganj,BD">Habiganj,BD</option>
      <option value="Rajshahi,BD">Rajshahi,BD</option>
      <option value="Tongi,BD">Tongi,BD</option>
      <option value="Narsingdi,BD">Narsingdi,BD</option>
      <option value="Farīdpur,BD">Farīdpur,BD</option>
      <option value="Jessore,BD">Jessore,BD</option>
      <option value="Comilla,BD">Comilla,BD</option>
      <option value="Feni,BD">Feni,BD</option>
    </select>
  </div>

```

```

        <option value="Bāndarban,BD">Bāndarban,BD</option>
        <option value="Cox's Bāzār,BD">Cox's Bāzār,BD</option>
    </select>
</div>
<div id="airQualityData" class="text-center">
    <p id="aqiIndex" class="text-2xl font-bold mb-2"></p>
    <p id="airQualityDescription" class="text-lg mb-2"></p>
    <ul id="pollutantList" class="list-none p-0"></ul>
</div>
</div>

<!-- Book Tracker Widget -->
<div class="widget p-6 book-tracker">
    <h2 class="text-xl font-semibold mb-4"><i class="fas fa-book
mr-2"></i>Book Tracker</h2>
    <input type="text" id="newBookName" placeholder="Book Name"
class="w-full p-2 mb-2 border rounded">
    <input type="datetime-local" id="bookDateTime" class="w-full p-2
mb-2 border rounded">
    <button onclick="addBook()" class="w-full bg-blue-500 text-white
p-2 rounded hover:bg-blue-600">Add Book</button>
    <ul id="bookList" class="mb-4"></ul>
</div>

<!-- Social Links Widget -->
<div class="widget p-6">
    <h2 class="text-xl font-semibold mb-4"><i class="fas fa-share-alt
mr-2"></i>Social</h2>
    <ul class="social-links">
        <li><a href="https://www.facebook.com/messages"
target="__blank" class="facebook"><i class="fab fa-facebook"></i>
Facebook</a></li>
        <li><a href="https://mail.google.com" target="__blank"
class="gmail"><i class="fab fa-google"></i> Gmail</a></li>
        <li><a href="https://web.telegram.org" target="__blank"
class="telegram"><i class="fab fa-telegram-plane"></i> Telegram</a></li>

```



```

        <li><a href="https://www.x.com" target="_blank"
class="twitter"><i class="fab fa-twitter"></i> X (Twitter)</a></li>
        <li><a href="https://www.notion.so" target="_blank"
class="notion"><i class="fas fa-book"></i> Notion</a></li>
        <li><a href="https://discord.com/app" target="_blank"
class="discord"><i class="fas fa-book"></i> Discord</a></li>
        <li><a href="https://web.whatsapp.com" target="_blank"
class="whatsapp"><i class="fab fa-whatsapp"></i> WhatsApp</a></li>
    </ul>
</div>

```

```

    <!-- Competitive Programming Widget -->
<div class="widget p-6">
    <h2 class="text-xl font-semibold mb-4"><i class="fas fa-code
mr-2"></i>Competitive Programming</h2>
    <ul class="social-links">
        <li><a href="https://codeforces.com/" target="_blank"
class="codeforces"><i class="fas fa-code"></i> Codeforces</a></li>
        <li><a href="https://atcoder.jp/" target="_blank"
class="atcoder"><i class="fas fa-code"></i> AtCoder</a></li>
        <li><a href="https://leetcode.com/" target="_blank"
class="leetcode"><i class="fas fa-code"></i> LeetCode</a></li>
        <li><a href="https://projecteuler.net/" target="_blank"
class="projecteuler"><i class="fas fa-code"></i> Project Euler</a></li>
        <li><a href="https://www.codechef.com/" target="_blank"
class="codechef"><i class="fas fa-code"></i> CodeChef</a></li>
        <li><a href="https://www.hackerrank.com/" target="_blank"
class="hackerrank"><i class="fas fa-code"></i> HackerRank</a></li>
        <li><a href="https://www.topcoder.com/" target="_blank"
class="topcoder"><i class="fas fa-code"></i> TopCoder</a></li>
    </ul>
</div>

```

```

<!-- AI Widget -->
<div class="widget p-6">
    <h2 class="text-xl font-semibold mb-4"><i class="fas fa-robot
mr-2"></i>AI Assistant</h2>
    <div class="flex items-center mb-2">

```

```

    <select id="aiPlatformSelect" class="w-full p-2 mb-2 border rounded"
onchange="updateAIIcon()">
      <option value="gemini">Gemini</option>
      <option value="aistudio">AI Studio</option>
      <option value="openai">ChatGPT</option>
      <option value="perplexity">Perplexity</option>
      <option value="claude">Claude</option>
    </select>
    <span id="aiIcon" class="ml-2"><i class="fas
fa-question-circle"></i></span> </div>
    <textarea id="aiPrompt" placeholder="Enter your prompt here..."
class="w-full p-2 mb-2 border rounded"></textarea>
    <button id="submitPrompt" class="w-full p-2 rounded hover:scale-105
animated-button mb-2">Submit</button>
</div>

```

```

<!-- Mindfulness Breathing Exercise Widget -->
<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4"><i class="fas fa-lungs
mr-2"></i>Mindfulness Breathing</h2>
  <div id="breathingExercise" class="text-center">
    <div class="breathing-container mx-auto mb-4">
      <div id="breathingCircle" class="breathing-circle"></div>
    </div>
    <div id="breathingText" class="text-2xl font-bold mb-4">Breathe
In</div>
    <div id="cycleCounter" class="text-lg mb-4">Cycles: 0</div>
    <button id="startBreathing" class="bg-blue-500 text-white p-2
rounded hover:bg-blue-600 mr-2">Start Exercise</button>
    <button id="resetBreathing" class="bg-red-500 text-white p-2
rounded hover:bg-red-600">Reset</button>
  </div>
</div>
<!-- calculator -->
<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4">
    <i class="fas fa-calculator mr-2"></i>Calculator

```

```

</h2>
<div class="calculator">
  <input
    type="text"
    id="display"
    class="w-full p-2 mb-4 border rounded text-right"
    readonly
  />
  <div class="grid grid-cols-4 gap-2">
    <button class="calc-button"
onclick="appendNumber('7')">7</button>
    <button class="calc-button"
onclick="appendNumber('8')">8</button>
    <button class="calc-button"
onclick="appendNumber('9')">9</button>
    <button class="calc-button"
onclick="appendOperator('/')">/</button>
    <button class="calc-button"
onclick="appendNumber('4')">4</button>
    <button class="calc-button"
onclick="appendNumber('5')">5</button>
    <button class="calc-button"
onclick="appendNumber('6')">6</button>
    <button class="calc-button"
onclick="appendOperator('*')">*</button>
    <button class="calc-button"
onclick="appendNumber('1')">1</button>
    <button class="calc-button"
onclick="appendNumber('2')">2</button>
    <button class="calc-button"
onclick="appendNumber('3')">3</button>
    <button class="calc-button"
onclick="appendOperator('-')">-</button>
    <button class="calc-button"
onclick="appendNumber('0')">0</button>
    <button class="calc-button"
onclick="appendNumber('.')">.</button>

```

```

        <button class="calc-button" onclick="calculate()">=</button>
        <button class="calc-button"
onclick="appendOperator('+')">+</button>
    </div>
    <button
        class="w-full bg-blue-500 text-white p-2 rounded
hover:bg-blue-600 mt-4"
        onclick="clearDisplay()"
    >
        Clear
    </button>
</div>
</div>

<!-- Notes Widget -->
<div class="widget p-6" id="notesWidget">
    <h2 class="text-xl font-semibold mb-4"><i class="fas
fa-sticky-note mr-2"></i>Notes</h2>
    <textarea id="quickNotes" placeholder="Take some notes..."
class="w-full p-2 mb-2 border rounded"></textarea>
    <button onclick="saveNotes()" class="w-full bg-blue-500 text-white
p-2 rounded hover:bg-blue-600 save-notes-button">Save Notes</button>
</div>

<!--mind games--->
<div class="widget p-6">
    <h2 class="text-xl font-semibold mb-4"><i class="fas fa-puzzle-piece
mr-2"></i>Memory Match</h2>
    <div class="level-select">
        <label for="level">Choose Difficulty:</label>
        <select id="level">
            <option value="easy">Easy (16)</option>
            <option value="medium">Medium (25)</option>
            <option value="hard">Hard (36)</option>
        </select>
        <button id="startButton">Start Game</button>
    </div>
    <div class="game-container" id="gameContainer"></div>

```

```

<div class="score-container">
  <p>Score: <span id="score">0</span></p>
  <p>Moves: <span id="moves">0</span></p>
</div>
</div>

```

```

<!-- Sudoku Widget -->
<div id="sudoku-widget" class="widget p-6">
  <h2 class="text-xl font-semibold mb-4">Sudoku Game</h2>
  <div id="high-score" class="text-center text-2xl font-bold mb-2">High
Score: 0</div>
  <div id="lives" class="text-center text-2xl font-bold mb-4">Lives: 3</div>
  <div id="sudoku-board" class="grid grid-cols-9 gap-1 mx-auto"></div>
  <div class="flex justify-between flex-wrap mt-4">
    <button id="new-game" class="w-full bg-gradient-to-r from-green-400
via-green-500 to-green-600 text-white p-2 rounded hover:scale-105
animated-button mb-2">New Game</button>
    <button id="solve" class="w-full bg-gradient-to-r from-red-400
via-red-500 to-red-600 text-white p-2 rounded hover:scale-105
animated-button mb-2">Solution</button>
    <button id="hint" class="w-full bg-gradient-to-r from-yellow-400
via-yellow-500 to-yellow-600 text-white p-2 rounded hover:scale-105
animated-button mb-2">Hint</button>
    <button id="check-solved" class="w-full bg-gradient-to-r from-blue-400
via-blue-500 to-blue-600 text-white p-2 rounded hover:scale-105
animated-button mb-2">Check Solved</button>
  </div>
</div>

```

```

<!-- MAP -->
<div class="widget p-6">
  <h2 class="text-xl font-semibold mb-4"><i class="fas fa-map-marker-alt
mr-2"></i>Location</h2>
  <div id="map" style="height: 300px;"></div>
  <p class="mt-2">Latitude: <span id="latitude"></span></p>
  <p class="mt-1">Longitude: <span id="longitude"></span></p>
</div>

```

```

<!-- Typing Test Widget -->
<div class="widget p-6" id="typingTestWidget">
  <h2 class="text-xl font-semibold mb-4"><i class="fas fa-keyboard
mr-2"></i>Typing Test</h2>
  <div class="text-center mb-4">
    <p id="typingTestText" class="text-lg mb-4">Type the following
text:</p>
    <p id="textToType" class="font-bold text-xl mb-4">The quick brown fox
jumps over the lazy dog.</p>
  </div>
  <textarea id="userInput" placeholder="Start typing here..." class="w-full
p-2 mb-2 border rounded" oninput="checkTyping()"></textarea>
  <div id="result" class="text-center mt-4">
    <p id="wordsPerMinute" class="text-lg"></p>
    <p id="accuracy" class="text-lg"></p>
  </div>
</div>

</div>
<script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
<script src="dashboard.js"></script>
</body>
</html>

```

8.Dashboard.js:
 // Clock and Date

```

function updateClockAndDate() {
    const now = new Date();
    document.getElementById('clock').textContent =
now.toLocaleTimeString();
    document.getElementById('date').textContent =
now.toLocaleDateString('en-US', {
        weekday: 'long',
        year: 'numeric',
        month: 'long',
        day: 'numeric'
    });
}
setInterval(updateClockAndDate, 1000);

```

```

// Fetch Weather Data
function fetchWeather() {
    const apiKey = 'API_KEY'; // Your API key
    const city = document.getElementById('citySelect').value; // Get selected city
    const apiUrl =
`https://api.openweathermap.org/data/2.5/weather?q=${city}&units=metric&
appid=${apiKey}`;

```

```

    fetch(apiUrl)
        .then(response => response.json())
        .then(data => {
            document.getElementById('weatherLocation').textContent =
`${data.name}, ${data.sys.country}`;
            document.getElementById('weatherTemp').textContent =
`${Math.round(data.main.temp)}°C`;
            document.getElementById('weatherDescription').textContent =
data.weather[0].description;
        })
        .catch(error => console.error('Error fetching weather data:', error));
}

```

```

// Initialize

```

```

window.onload = function() {
  updateClockAndDate();
  updateProductivityScore();
  const savedNotes = localStorage.getItem('quickNotes');

  // Add event listener to citySelect
  document.getElementById('citySelect').addEventListener('change',
  fetchWeather);

  // Fetch weather for the default city when the page loads
  fetchWeather();

  document.getElementById('newTask').addEventListener('keyup',
  function(event) {
    if (event.key === 'Enter') {
      addTask();
    }
  });
  document.getElementById('newReminder').addEventListener('keyup',
  function(event) {
    if (event.key === 'Enter') {
      addReminder();
    }
  });

  initCalculator(); // Initialize the calculator widget
};

// Motivational Quotes
const quotes = [
  "The secret of getting ahead is getting started. - Mark Twain",
  "It always seems impossible until it's done. - Nelson Mandela",
  "Don't watch the clock; do what it does. Keep going. - Sam Levenson",
  "The future depends on what you do today. - Mahatma Gandhi",
  "Believe you can and you're halfway there. - Theodore Roosevelt"

```



```

];
function setRandomQuote() {
    const quoteElement = document.getElementById('motivationalQuote');
    quoteElement.textContent = quotes[Math.floor(Math.random() *
quotes.length)];
}
setRandomQuote();
setInterval(setRandomQuote, 3600000); // Update every 6 hours

// Reminders (Updated)
function addReminder() {
    const reminderInput = document.getElementById('newReminder');
    const reminderTimeInput = document.getElementById('reminderTime');
    const reminderList = document.getElementById('reminderList');

    if (reminderInput.value.trim() !== '' && reminderTimeInput.value.trim() !==
'') {
        const reminderText = `${reminderInput.value} - ${new
Date(reminderTimeInput.value).toLocaleString()}`;
        const li = document.createElement('li');
        li.innerHTML = `
            <div class="flex items-center justify-between py-2">
                <div class="flex items-center"> <!-- Added this inner div -->
                    <input type="checkbox" onchange="updateProductivityScore()"
class="mr-2">
                        <span>${reminderText}</span>
                    </div>
                    <button onclick="this.closest('li').remove();
updateProductivityScore()" class="text-red-500 hover:text-red-700">
                        <i class="fas fa-trash"></i>
                    </button>
                </div>
            `;
        reminderList.appendChild(li);
        reminderInput.value = '';
        reminderTimeInput.value = '';
        updateProductivityScore();
    }
}

```

```
}  
}
```

```
// Tasks (Updated)  
function addTask() {  
  const taskInput = document.getElementById('newTask');  
  const taskTypeSelect = document.getElementById('taskType');  
  const taskImportanceSelect = document.getElementById('taskImportance');  
  const taskList = document.getElementById('taskList');  
  
  if (taskInput.value.trim() !== '') {  
    const taskText = taskInput.value;  
    const taskType = taskTypeSelect.value;  
    const taskImportance = taskImportanceSelect.value;  
  
    const li = document.createElement('li');  
    li.innerHTML = `  
      <div class="flex items-center justify-between py-2">  
        <div class="flex items-center"> <!-- Added this inner div -->  
          <input type="checkbox" onchange="updateProductivityScore()  
class="mr-2" data-importance="${taskImportance}">  
          <span>${taskText} (${taskType}) - Importance:  
${taskImportance}</span>  
        </div>  
        <button onclick="this.closest('li').remove();  
updateProductivityScore()" class="text-red-500 hover:text-red-700">  
          <i class="fas fa-trash"></i>  
        </button>  
      </div>  
    `;  
    taskList.appendChild(li);  
  
    taskInput.value = '';  
    updateProductivityScore();  
  }  
}
```

```

// Water Intake Tracker
let waterIntake = 0;
function addWater() {
  if (waterIntake < 8) {
    waterIntake++;
    updateWaterIntake();
    updateProductivityScore();
  }
}
function updateWaterIntake() {
  document.getElementById('waterIntake').textContent = `${waterIntake} / 8
  glasses`;
  document.getElementById('waterProgress').style.width = `${(waterIntake /
  8) * 100}%`;
}

// Focus Timer
let focusTimer;
let focusTimeRemaining = 1500;
function startFocusTimer() {
  clearInterval(focusTimer);
  focusTimer = setInterval(() => {
    if (focusTimeRemaining > 0) {
      focusTimeRemaining--;
      updateFocusTimer();
    } else {
      clearInterval(focusTimer);
      alert('Focus session completed!');
      updateProductivityScore();
    }
  }, 1000);
}
function stopFocusTimer() {
  clearInterval(focusTimer);
}
function updateFocusTimer() {
  const minutes = Math.floor(focusTimeRemaining / 60);

```

```
    const seconds = focusTimeRemaining % 60;
    document.getElementById('focusTimer').textContent =
` ${minutes.toString().padStart(2, '0')}: ${seconds.toString().padStart(2, '0')} `;
}
```

```
// Logout
function logout() {
    window.location.href = "index.html";
}
```

```
// Productivity Score (Corrected)
function updateProductivityScore() {
    const tasks = document.querySelectorAll('#taskList
input[type="checkbox"]'); // Select all checkboxes
    let totalScore = 0;
    let completedTasks = 0;

    tasks.forEach(task => {
        const importance = parseInt(task.dataset.importance);
        totalScore += importance;

        // Check if the task is checked *before* adding to completedTasks
        if (task.checked) {
            completedTasks += importance;
        }
    });
}
```

```
const reminders = document.querySelectorAll('#reminderList
input[type="checkbox"]'); // Select all reminder checkboxes
reminders.forEach(reminder => {
    totalScore += 1;

    // Check if the reminder is checked *before* adding to completedTasks
    if (reminder.checked) {
        completedTasks += 1;
    }
});
```

```

// Calculate productivity score (taking task importance into account)
let productivityScore = 0;
if (totalScore > 0) {
    productivityScore = Math.round((completedTasks / totalScore) * 100);
}

document.getElementById('productivityScore').textContent =
`$${productivityScore}%`;
document.getElementById('productivityProgress').style.width =
`$${productivityScore}%`;
}

// Initialize
window.onload = function() {
    updateClockAndDate();
    updateProductivityScore();
    const savedNotes = localStorage.getItem('quickNotes');

    // Initialize Weather
    fetchWeather();

    document.getElementById('newTask').addEventListener('keyup',
function(event) {
    if (event.key === 'Enter') {
        addTask();
    }
});
    document.getElementById('newReminder').addEventListener('keyup',
function(event) {
    if (event.key === 'Enter') {
        addReminder();
    }
});

    initCalculator(); // Initialize the calculator widget
};

```

```

// Timer updates
function updateTimer() {
    const timerMinutes = document.getElementById('timerMinutes').value;
    focusTimeRemaining = timerMinutes * 60;
    updateFocusTimer();
}

// Upcoming Events
function addEvent() {
    const eventInput = document.getElementById('newEvent');
    const eventDateInput = document.getElementById('eventDate');
    const eventList = document.getElementById('eventList');

    if (eventInput.value.trim() !== '' && eventDateInput.value.trim() !== '') {
        const eventText = `${eventInput.value} - ${eventDateInput.value}`;
        const li = document.createElement('li');
        li.innerHTML = `${eventText} <button onclick="deleteEvent(this)"
class="delete-button"><i class="fas fa-trash"></i></button>`;
        eventList.appendChild(li);
        eventInput.value = '';
        eventDateInput.value = '';
    }
}

function deleteEvent(button) {
    const li = button.parentNode;
    li.remove();
}

// Book Tracker
let books = []; // Array to store book data

function addBook() {
    const bookNameInput = document.getElementById('newBookName');
    const bookDateTimeInput = document.getElementById('bookDateTime');

```

```

const bookList = document.getElementById('bookList');

if (bookNameInput.value.trim() !== '' && bookDateTimeInput.value.trim()
!== '') {
  const newBook = {
    name: bookNameInput.value,
    dateTime: new Date(bookDateTimeInput.value)
  };

  books.push(newBook);
  updateBookList();

  bookNameInput.value = '';
  bookDateTimeInput.value = '';
}
}

```

```

function updateBookList() {
  const bookList = document.getElementById('bookList');
  bookList.innerHTML = ''; // Clear the list

  // Sort books by date/time (ascending)
  books.sort((a, b) => a.dateTime - b.dateTime);

  books.forEach((book, index) => {
    const li = document.createElement('li');
    li.textContent = `${index + 1}. ${book.name} -
${book.dateTime.toLocaleString()}`;
    bookList.appendChild(li);
  });
}

```

```

// Calculator Widget Logic
let display = document.getElementById('display');
let previousOperator = null;
let previousOperand = null;

```

```
function appendNumber(number) {  
  display.value += number;  
}
```

```
function appendOperator(operator) {  
  if (previousOperator) {  
    calculate();  
  }  
  previousOperator = operator;  
  previousOperand = parseFloat(display.value);  
  display.value += operator;  
}
```

```
function calculate() {  
  let currentOperand =  
parseFloat(display.value.substring(display.value.lastIndexOf(previousOperato  
r) + 1));  
  let result;
```

```
  switch (previousOperator) {  
    case '+':  
      result = previousOperand + currentOperand;  
      break;  
    case '-':  
      result = previousOperand - currentOperand;  
      break;  
    case '*':  
      result = previousOperand * currentOperand;  
      break;  
    case '/':  
      if (currentOperand === 0) {  
        result = "Error";  
      } else {  
        result = previousOperand / currentOperand;  
      }  
      break;
```



```

    }

    display.value = result;
    previousOperator = null;
    previousOperand = null;
}

function clearDisplay() {
    display.value = '';
}

// Mindfulness & Meditation Widget Script

let breathingInterval;
let isBreathing = false;
let cycles = 0;

function startBreathingExercise() {
    if (isBreathing) {
        stopBreathingExercise();
    } else {
        isBreathing = true;
        document.getElementById('startBreathing').textContent = 'Stop Exercise';
        breathingCycle();
        breathingInterval = setInterval(breathingCycle, 12000); // 12 seconds per
full cycle
    }
}

function breathingCycle() {
    const breathingText = document.getElementById('breathingText');
    const breathingCircle = document.getElementById('breathingCircle');

    // Breathe In
    breathingText.textContent = 'Breathe In';
    breathingCircle.classList.remove('exhale', 'hold');

```

```

breathingCircle.classList.add('inhale');

// Hold
setTimeout(() => {
  breathingText.textContent = 'Hold';
  breathingCircle.classList.remove('inhale', 'exhale');
  breathingCircle.classList.add('hold');
}, 4000);

// Breathe Out
setTimeout(() => {
  breathingText.textContent = 'Breathe Out';
  breathingCircle.classList.remove('inhale', 'hold');
  breathingCircle.classList.add('exhale');
}, 8000);

// Increment cycle counter
setTimeout(() => {
  cycles++;
  document.getElementById('cycleCounter').textContent = `Cycles:
${cycles}`;
}, 11900);
}

function stopBreathingExercise() {
  clearInterval(breathingInterval);
  document.getElementById('startBreathing').textContent = 'Start Exercise';
  document.getElementById('breathingText').textContent = 'Breathe In';
  document.getElementById('breathingCircle').classList.remove('inhale',
'hold', 'exhale');
  isBreathing = false;
}

function resetBreathingExercise() {
  stopBreathingExercise();
  cycles = 0;
  document.getElementById('cycleCounter').textContent = 'Cycles: 0';
}

```

```

}
// games
const gameContainer = document.getElementById('gameContainer');
const scoreElement = document.getElementById('score');
const movesElement = document.getElementById('moves');
const levelSelect = document.getElementById('level');
const startButton = document.getElementById('startButton');

let score = 0;
let moves = 0;
let flippedCards = [];
let matchedCards = [];
let gameBoard = [];
let numCards;

const symbols = ['🍎', '🍌', '🍇', '🍓', '🍉', '🍍', '🍒', '🍑', '🍊', '🍋', '🍈',
'🥥', '🥝', '🥑', '🍆', '🥕'];

// Function to start the game based on selected level
startButton.addEventListener('click', () => {
  const selectedLevel = levelSelect.value;
  switch (selectedLevel) {
    case 'easy':
      numCards = 8;
      break;
    case 'medium':
      numCards = 10;
      break;
    case 'hard':
      numCards = 12;
      break;
  }
  startGame();
});

// Function to create a card element
function createCard(symbol) {

```

```

const card = document.createElement('div');
card.classList.add('card');
card.innerHTML = `
  <div class="card-content">${symbol}</div>
`;
card.addEventListener('click', flipCard);
return card;
}

```

```

// Function to shuffle the array (Fisher-Yates Shuffle)
function shuffle(array) {
  for (let i = array.length - 1; i > 0; i--) {
    const j = Math.floor(Math.random() * (i + 1));
    [array[i], array[j]] = [array[j], array[i]];
  }
  return array;
}

```

```

// Function to start the game
function startGame() {
  score = 0;
  moves = 0;
  flippedCards = [];
  matchedCards = [];
  scoreElement.textContent = score;
  movesElement.textContent = moves;

  gameBoard = shuffle(symbols.slice(0,
numCards)).concat(shuffle(symbols.slice(0, numCards))); // Create pairs
  gameContainer.innerHTML = ''; // Clear previous game

  // Create and add cards to the game board
  gameBoard.forEach(symbol => {
    const card = createCard(symbol);
    gameContainer.appendChild(card);
  });
}

```

```

// Function to flip a card
function flipCard() {
  if (flippedCards.length < 2 && !matchedCards.includes(this) &&
!flippedCards.includes(this)) {
    this.classList.add('flipped');
    flippedCards.push(this);

    if (flippedCards.length === 2) {
      moves++;
      movesElement.textContent = moves;

      // Check if the cards match
      const card1 =
flippedCards[0].querySelector('.card-content').textContent;
      const card2 =
flippedCards[1].querySelector('.card-content').textContent;

      if (card1 === card2) {
        setTimeout(() => {
          flippedCards[0].classList.add('matched');
          flippedCards[1].classList.add('matched');
          matchedCards.push(...flippedCards);
          flippedCards = [];

          score += 10;
          scoreElement.textContent = score;

          // Check if the game is over
          if (matchedCards.length === gameBoard.length) {
            setTimeout(() => {
              alert(`Congratulations! You won with ${moves} moves!`);
              startGame(); // Start a new game
            }, 500);
          }
        }, 1000); // Wait for 1 second before checking for match
      } else {

```

```

        setTimeout(() => {
            flippedCards[0].classList.remove('flipped');
            flippedCards[1].classList.remove('flipped');
            flippedCards = [];
        }, 1000); // Wait for 1 second before flipping back
    }
}
}
}

```

```

// Initially hide the game board
gameContainer.style.display = "none"; // Hide the board at the start

```

```

// Start the game when the "Start Game" button is clicked
startButton.addEventListener('click', () => {
    gameContainer.style.display = "grid"; // Show the game board
    startGame();
});

```

```

// Sudoku grid
// Sudoku Code with Lives, High Score Saving, and Improved Styling
const board = document.getElementById('sudoku-board');
const newGameBtn = document.getElementById('new-game');
const solveBtn = document.getElementById('solve');
const hintBtn = document.getElementById('hint');
const checkSolvedBtn = document.getElementById('check-solved');
const highScoreElement = document.getElementById('high-score');
const livesElement = document.getElementById('lives');
let selectedCell = null;
let puzzle = [];
let solution = [];
let highScore = localStorage.getItem('highScore') || 0;
let lives = 3;
let hintsUsed = 0;

```

```

function generatePuzzle() {
    const base = [

```

```
[5,3,4,6,7,8,9,1,2],  
[6,7,2,1,9,5,3,4,8],  
[1,9,8,3,4,2,5,6,7],  
[8,5,9,7,6,1,4,2,3],  
[4,2,6,8,5,3,7,9,1],  
[7,1,3,9,2,4,8,5,6],  
[9,6,1,5,3,7,2,8,4],  
[2,8,7,4,1,9,6,3,5],  
[3,4,5,2,8,6,1,7,9]  
];
```

```
for (let i = 0; i < 9; i += 3) {  
  const rows = [i, i + 1, i + 2];  
  for (let j = 0; j < 2; j++) {  
    const r1 = Math.floor(Math.random() * 3);  
    const r2 = Math.floor(Math.random() * 3);  
    [base[rows[r1]], base[rows[r2]]] = [base[rows[r2]], base[rows[r1]]];  
  }  
}
```

```
solution = base;  
puzzle = JSON.parse(JSON.stringify(base));
```

```
for (let i = 0; i < 40; i++) {  
  const row = Math.floor(Math.random() * 9);  
  const col = Math.floor(Math.random() * 9);  
  puzzle[row][col] = 0;  
}  
}
```

```
function renderBoard() {  
  board.innerHTML = '';  
  for (let i = 0; i < 9; i++) {  
    for (let j = 0; j < 9; j++) {  
      const cell = document.createElement('div');  
      cell.classList.add('cell');  
      cell.dataset.row = i;
```

```

    cell.dataset.col = j;
    if (puzzle[i][j] !== 0) {
        cell.textContent = puzzle[i][j];
        cell.classList.add('initial');
    }
    cell.addEventListener('click', selectCell);
    board.appendChild(cell);
}
}
}

```

```

function selectCell(event) {
    if (selectedCell) {
        selectedCell.classList.remove('selected');
    }
    selectedCell = event.target;
    selectedCell.classList.add('selected');
}

```

```

function handleKeyPress(event) {
    if (selectedCell && !selectedCell.classList.contains('initial')) {
        const key = event.key;
        if (key >= '1' && key <= '9') {
            const row = parseInt(selectedCell.dataset.row);
            const col = parseInt(selectedCell.dataset.col);
            if (puzzle[row][col] === solution[row][col]) return;
            puzzle[row][col] = parseInt(key);
            selectedCell.textContent = key;
            selectedCell.classList.remove('error');
            if (puzzle[row][col] !== solution[row][col]) {
                selectedCell.classList.add('error');
                reduceLife();
            } else {
                checkWin();
            }
        } else if (key === 'Backspace' || key === 'Delete') {
            const row = parseInt(selectedCell.dataset.row);

```



```

    const col = parseInt(selectedCell.dataset.col);
    puzzle[row][col] = 0;
    selectedCell.textContent = '';
    selectedCell.classList.remove('error');
  }
}

```

```

function newGame() {
  lives = 3;
  hintsUsed = 0;
  updateLives();
  generatePuzzle();
  renderBoard();
}

```

```

function solve() {
  puzzle = JSON.parse(JSON.stringify(solution));
  renderBoard();
  resetScore();
}

```

```

function hint() {
  if (hintsUsed >= 3) return;
  hintsUsed++;
  let emptyCells = [];
  for (let i = 0; i < 9; i++) {
    for (let j = 0; j < 9; j++) {
      if (puzzle[i][j] === 0) {
        emptyCells.push({ row: i, col: j });
      }
    }
  }
  if (emptyCells.length > 0) {
    const randomCell = emptyCells[Math.floor(Math.random() *
emptyCells.length)];

```

```

    puzzle[randomCell.row][randomCell.col] =
solution[randomCell.row][randomCell.col];
    renderBoard();
}
updateHighScore(-5); // Deduct 5 points for each hint
}

```

```

function checkSolved() {
    let solved = true;
    for (let i = 0; i < 9; i++) {
        for (let j = 0; j < 9; j++) {
            if (puzzle[i][j] !== solution[i][j]) {
                solved = false;
                break;
            }
        }
        if (!solved) break;
    }
    if (solved) {
        alert('Congratulations! You solved the puzzle!');
        updateHighScore();
        winAnimation();
    } else {
        alert('Puzzle is not solved correctly yet. ');
    }
}

```

```

function reduceLife() {
    lives--;
    updateLives();
    if (lives === 0) {
        alert('Game Over! You ran out of lives. ');
        resetScore();
        solve(); // Show the solution
    }
}

```

```
function updateLives() {  
  livesElement.textContent = `Lives: ${lives}`;  
}
```

```
function updateHighScore(change = 10) {  
  highScore += change;  
  highScoreElement.textContent = `High Score: ${highScore}`;  
  localStorage.setItem('highScore', highScore);  
}
```

```
function resetScore() {  
  highScore = 0;  
  highScoreElement.textContent = `High Score: 0`;  
  localStorage.setItem('highScore', highScore);  
}
```

```
function checkWin() {  
  let currentScore = calculateScore();  
  if (currentScore > highScore) {  
    highScore = currentScore;  
    highScoreElement.textContent = `High Score: ${highScore}`;  
    localStorage.setItem('highScore', highScore);  
  }  
}
```

```
function calculateScore() {  
  let score = 0;  
  for (let i = 0; i < 9; i++) {  
    for (let j = 0; j < 9; j++) {  
      if (puzzle[i][j] === solution[i][j]) {  
        score++;  
      }  
    }  
  }  
  return score;  
}
```

```

newGameBtn.addEventListener('click', newGame);
solveBtn.addEventListener('click', solve);
hintBtn.addEventListener('click', hint);
checkSolvedBtn.addEventListener('click', checkSolved);
document.addEventListener('keydown', handleKeyPress);

// Initialize the game and load the saved high score
newGame();
highScoreElement.textContent = `High Score: ${highScore}`;

// Add these to your window.onload function
document.getElementById('startBreathing').addEventListener('click',
startBreathingExercise);
document.getElementById('resetBreathing').addEventListener('click',
resetBreathingExercise);

// AI Widget
const aiPlatformSelect = document.getElementById('aiPlatformSelect');
const aiPrompt = document.getElementById('aiPrompt');
const submitPrompt = document.getElementById('submitPrompt');
const aiIcon = document.getElementById('aiIcon');

// Function to update the AI icon based on the selected platform
function updateAIIcon() {
    const selectedPlatform = aiPlatformSelect.value;

    switch (selectedPlatform) {
        case 'gemini':
            aiIcon.innerHTML = '';
            submitPrompt.style.backgroundImage = 'linear-gradient(135deg,
#f700ff, #3700fd, #ff0000)';
            break;
        case 'aistudio':

```

```

        aiIcon.innerHTML = '';
        submitPrompt.style.backgroundImage = 'linear-gradient(135deg,
#1eff00, #00e1ff, #001aff)';
        break;
    case 'openai':
        aiIcon.innerHTML = '';
        submitPrompt.style.backgroundImage = 'linear-gradient(135deg,
#000000, #9c9494, #000000)';
        break;
    case 'perplexity':
        aiIcon.innerHTML = '<i class="fas fa-question-circle"></i>';
        aiIcon.innerHTML = '';

        aiIcon.querySelector('img').style.width = '24px';
        aiIcon.querySelector('img').style.height = '24px';
        break;
    case 'claude':
        aiIcon.innerHTML = '';
        submitPrompt.style.backgroundImage = 'linear-gradient(135deg,
#ff9900, #fdc000, #ff0000)';

        aiIcon.querySelector('img').style.width = '24px';
        aiIcon.querySelector('img').style.height = '24px';
        break;
    default:
        aiIcon.innerHTML = '';
        submitPrompt.style.backgroundImage = 'linear-gradient(135deg,
#667eea, #764ba2, #ff7e5f, #feb47b)';

```

```

    }
  }

  updateAIIcon();

  aiPlatformSelect.addEventListener('change', updateAIIcon);

  submitPrompt.addEventListener('click', () => {
    const selectedPlatform = aiPlatformSelect.value;
    const prompt = aiPrompt.value;

    let url = '';

    switch (selectedPlatform) {
      case 'gemini':
        url = 'https://gemini.google.com/app/' +
        encodeURIComponent(prompt);
        break;
      case 'aistudio':
        url = 'https://aistudio.google.com/app/prompts/new_chat/' +
        encodeURIComponent(prompt);
        break;
      case 'openai':
        url = 'https://chat.openai.com/chat?q=' +
        encodeURIComponent(prompt);
        break;
      case 'perplexity':
        url = 'https://www.perplexity.ai/search?q=' +
        encodeURIComponent(prompt);
        break;
      case 'claude':
        url = 'https://claude.ai/search?q=' + encodeURIComponent(prompt);
        break;
      default:
        url = 'https://www.google.com/search?q=' +
        encodeURIComponent(prompt);
    }
  }

```

```

    window.open(url, '__blank');
    aiPrompt.value = '';
  });

// Typing Test Widget

let startTime;
let endTime;
const typingTestText = document.getElementById('textToType').textContent;

function checkTyping() {
  const userInput = document.getElementById('userInput').value;

  if (!startTime) {
    startTime = new Date();
  }

  if (userInput === typingTestText) {
    endTime = new Date();
    calculateResults();
  }
}

function calculateResults() {
  const timeDiff = (endTime - startTime) / 1000;
  const words = typingTestText.split(' ').length;
  const wordsPerMinute = Math.round((words / timeDiff) * 60);
  const accuracy = ((typingTestText ===
document.getElementById('userInput').value) ? 100 : 0).toFixed(2);

  document.getElementById('wordsPerMinute').textContent = `Words per
Minute: ${wordsPerMinute}`;
  document.getElementById('accuracy').textContent = `Accuracy:
${accuracy}%`;

```

```

    startTime = null;
    endTime = null;
    document.getElementById('userInput').value = '';
}

```

```

// Fetch 5-Day Forecast Data
function fetchFiveDayForecast() {
    const apiKey = 'API_KEY';
    const city = document.getElementById('citySelectAirForecast').value;
    const apiUrl =
`https://api.openweathermap.org/data/2.5/forecast?q=${city}&units=metric&
appid=${apiKey}`;

```

```

    fetch(apiUrl)
        .then(response => response.json())
        .then(data => {
            const forecastList = document.getElementById('forecastList');
            forecastList.innerHTML = '';

            // Group forecasts by day
            for (let i = 0; i < data.list.length; i += 8) {
                const forecast = data.list[i];
                const day = new Date(forecast.dt * 1000).toLocaleDateString('en-US', {
                    weekday: 'short' });
                const temp = Math.round(forecast.main.temp);
                const icon =
`http://openweathermap.org/img/wn/${forecast.weather[0].icon}@2x.png`;

                const forecastItem = document.createElement('li');
                forecastItem.classList.add('forecast-item');
                forecastItem.innerHTML = `
                    <div class="day">${day}</div>
                    

```



```

        <div class="temp">${temp}°C</div>
    `;
    forecastList.appendChild(forecastItem);
}
})
.catch(error => console.error('Error fetching weather data:', error));
}

// Initialize the 5-Day forecast
window.onload = function() {
    fetchFiveDayForecast();

    document.getElementById('citySelectAirForecast').addEventListener('change'
, fetchFiveDayForecast);
};

```

```

function fetchAirPollutionData() {
    const apiKey = 'API_KEY';
    const city = document.getElementById('citySelectAirQuality').value;

    fetch(`http://api.openweathermap.org/geo/1.0/direct?q=${city}&limit=1&appi
d=${apiKey}`)
        .then(response => response.json())
        .then(data => {
            if (data.length > 0) {
                const lat = data[0].lat;
                const lon = data[0].lon;

                fetchAirPollutionDataByCoords(lat, lon);
            } else {
                document.getElementById('aqiIndex').textContent = "City not found";
                document.getElementById('airQualityDescription').textContent = "";
                document.getElementById('pollutantList').innerHTML = '';
            }
        })
    }

```



```

        airQualityDescriptionText = "Moderate - Members of sensitive
groups may experience health effects. The general public is not likely to be
affected.";
        break;
    case 4:
        airQualityDescriptionText = "Poor - Increased likelihood of health
effects. The general public may begin to experience health effects; members of
sensitive groups may experience more serious health effects.";
        break;
    case 5:
        airQualityDescriptionText = "Very Poor - Health warnings of
emergency conditions. Everyone may experience more serious health
effects.";
        break;
    default:
        airQualityDescriptionText = "Data Unavailable";
    }
    airQualityDescription.textContent = airQualityDescriptionText;

    // Display pollutants
    pollutantList.innerHTML = '';
    for (const pollutant in data.list[0].components) {
        const pollutantItem = document.createElement('li');
        pollutantItem.classList.add('pollutant-item');
        pollutantItem.innerHTML = `
            <div class="pollutant-name">${pollutant.toUpperCase()}</div>
            <div
class="pollutant-value">${data.list[0].components[pollutant].toFixed(2)}
µg/m³</div>
        `;
        pollutantList.appendChild(pollutantItem);
    }
} else {
    aqiIndex.textContent = "Data Unavailable";
    airQualityDescription.textContent = "";
    pollutantList.innerHTML = "";
}

```

```

    })
    .catch(error => console.error('Error fetching air pollution data:', error));
}

// Initialize the Air Pollution Widget
window.onload = function() {
    fetchFiveDayForecast();

    document.getElementById('citySelectAirForecast').addEventListener('change'
    , fetchFiveDayForecast);

    fetchAirPollutionData();

    document.getElementById('citySelectAirQuality').addEventListener('change'
    , fetchAirPollutionData);
};

// Currency Converter Widget
const fromCurrencySelect = document.getElementById('fromCurrencySelect');
const toCurrencySelect = document.getElementById('toCurrencySelect');
const amountInput = document.getElementById('amountInput');
const conversionResult = document.getElementById('conversionResult');

let latestRates = {};

// Fetch Latest Exchange Rates (with a CORS proxy)
function fetchLatestRates() {
    const apiKey = 'null';
    const apiUrl = `https://data.fixer.io/api/latest?access_key=${apiKey}`;

    fetch(apiUrl)
    .then(response => {
        if (!response.ok) {
            throw new Error(`HTTP error! status: ${response.status}`);
        }
    })

```

```

    return response.json();
  })
  .then(data => {
    if (data.success) {
      latestRates = data.rates;
      populateCurrencyDropdown(fromCurrencySelect, latestRates);
      populateCurrencyDropdown(toCurrencySelect, latestRates);
      calculateConversion();
    } else {
      console.error('Fixer API Error:', data.error);
      conversionResult.textContent = `Error: ${data.error.info}`;
    }
  })
  .catch(error => {
    console.error('Network error:', error);
    conversionResult.textContent = 'Error: Unable to fetch data';
  });
}

```

// Function to populate dropdowns

```

function populateCurrencyDropdown(dropdown, rates) {
  dropdown.innerHTML = '';

  for (const currencyCode in rates) {
    const option = document.createElement('option');
    option.value = currencyCode;
    option.text = currencyCode;
    dropdown.add(option);
  }
}

```

// Calculate the conversion based on latest rates

```

function calculateConversion() {
  const fromCurrency = fromCurrencySelect.value;
  const toCurrency = toCurrencySelect.value;
  const amount = parseFloat(amountInput.value);

```

```

    if (latestRates && fromCurrency && toCurrency) {
        const conversionRate = latestRates[toCurrency] /
latestRates[fromCurrency];
        const convertedAmount = amount * conversionRate;
        conversionResult.textContent = `Converted Amount:
${convertedAmount.toFixed(2)}`;
    } else {
        conversionResult.textContent = 'Error: Rates not available.';
    }
}

// Event listeners for real-time conversion
window.onload = function() {
    fetchLatestRates();

    fromCurrencySelect.addEventListener('change', calculateConversion);
    toCurrencySelect.addEventListener('change', calculateConversion);
    amountInput.addEventListener('input', calculateConversion);
};

// ... (Your Other JavaScript Code) ...

// Event listeners for real-time conversion
window.onload = function() {
    fetchLatestRates();

    fromCurrencySelect.addEventListener('change', calculateConversion);
    toCurrencySelect.addEventListener('change', calculateConversion);
    amountInput.addEventListener('input', calculateConversion);
    initMap();
};

// Map Widget Logic
let map;

```

```

let marker;

function initMap() {
    map = L.map('map').setView([23.7104, 90.4125], 13);

    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        maxZoom: 19
    }).addTo(map);

    map.on('click', function(e) {
        if (marker) {
            map.removeLayer(marker);
        }

        marker = L.marker([e.latlng.lat, e.latlng.lng]).addTo(map);

        // Update the displayed coordinates
        document.getElementById('latitude').textContent = e.latlng.lat.toFixed(4);
        document.getElementById('longitude').textContent =
e.latlng.lng.toFixed(4);
    });
}

```

9.Db_config.php:

```

<?php
$servername = "localhost";
$username = "root"; // Default MySQL username
$password = ""; // Default MySQL password (if you changed it, use your
password)
$dbname = "productivity_db"; // The name of your database

// Create a connection
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check the connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

```

```
}  
?>
```

10.Delete_user.php:

```
<?php  
// Include the database connection  
include 'db_config.php';  
  
$userId = $_POST["userId"];  
  
$sql = "DELETE FROM users WHERE id = $userId";  
  
if (mysqli_query($conn, $sql)) {  
    echo "User deleted successfully.";  
} else {  
    echo "Error deleting user: " . mysqli_error($conn);  
}  
?>
```

11.Fetch_users.php:

```
<?php  
// Include the database connection  
include 'db_config.php';  
  
$sql = "SELECT * FROM users";  
$result = mysqli_query($conn, $sql);  
  
$users = array();  
while ($row = mysqli_fetch_assoc($result)) {  
    $users[] = $row;  
}  
  
echo json_encode($users);  
?>
```

12.Index.css:

```
/* General Styles */
```



```
body {  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);  
  background-size: 300% 300%;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  height: 100vh;  
  margin: 0;  
  color: #333;  
  animation: gradientAnimation 15s ease infinite;  
}
```

```
@keyframes gradientAnimation {  
  0% {  
    background-position: 0% 50%;  
  }  
  50% {  
    background-position: 100% 50%;  
  }  
  100% {  
    background-position: 0% 50%;  
  }  
}
```

```
.container {  
  background-color: rgba(255, 255, 255, 0.95);  
  border-radius: 20px;  
  padding: 40px;  
  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);  
  text-align: center;  
  max-width: 400px;  
  width: 100%;  
  transition: background-color 0.3s ease, transform 0.3s ease;  
}
```

```
.container:hover {
```

```
background-color: rgba(255, 255, 255, 0.85);
transform: scale(1.02);
}

.gradient-text {
background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
background-size: 300% 300%;
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;
animation: gradientTextAnimation 15s ease infinite;
}

@keyframes gradientTextAnimation {
0% {
background-position: 0% 50%;
}
50% {
background-position: 100% 50%;
}
100% {
background-position: 0% 50%;
}
}

.input-group {
position: relative;
margin-bottom: 20px;
}

.input-group i {
position: absolute;
left: 15px;
top: 50%;
transform: translateY(-50%);
color: #aaa;
}
```

```
input {  
  width: 100%;  
  padding: 12px 40px;  
  margin: 8px 0;  
  border: 2px solid #ddd;  
  border-radius: 50px;  
  box-sizing: border-box;  
  transition: all 0.3s ease;  
  font-size: 16px;  
  color: #333;  
  background-clip: padding-box;  
}
```

```
.input-animate::placeholder {  
  color: transparent;  
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);  
  background-size: 300% 300%;  
  -webkit-background-clip: text;  
  animation: placeholderAnimation 15s ease infinite;  
}
```

```
@keyframes placeholderAnimation {  
  0% {  
    background-position: 0% 50%;  
  }  
  50% {  
    background-position: 100% 50%;  
  }  
  100% {  
    background-position: 0% 50%;  
  }  
}
```

```
.input-animate:focus {  
  border-color: #667eea;  
  outline: none;  
  animation: inputFocusAnimation 0.5s ease-out forwards;
```

```
}
```

```
@keyframes inputFocusAnimation {
```

```
  0% {
```

```
    transform: scale(1);
```

```
    box-shadow: 0 0 5px rgba(102, 126, 234, 0.5);
```

```
  }
```

```
  100% {
```

```
    transform: scale(1.05);
```

```
    box-shadow: 0 0 15px rgba(102, 126, 234, 0.7);
```

```
  }
```

```
}
```

```
button {
```

```
  background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
```

```
  background-size: 300% 300%;
```

```
  color: white;
```

```
  padding: 12px 30px;
```

```
  border: none;
```

```
  border-radius: 50px;
```

```
  cursor: pointer;
```

```
  font-size: 18px;
```

```
  transition: all 0.3s ease;
```

```
  width: 100%;
```

```
  margin-top: 20px;
```

```
  animation: buttonGradientAnimation 15s ease infinite;
```

```
}
```

```
.btn-animate:hover {
```

```
  transform: translateY(-3px);
```

```
  box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);
```

```
}
```

```
@keyframes buttonGradientAnimation {
```

```
  0% {
```

```
    background-position: 0% 50%;
```

```
  }
```

```
50% {
  background-position: 100% 50%;
}
100% {
  background-position: 0% 50%;
}
}

.logo {
  font-size: 3em;
  animation: logoPulse 1.5s ease-in-out infinite;
}

@keyframes logoPulse {
  50% {
    transform: scale(1.1);
  }
}

.switch {
  margin-top: 20px;
  color: #666;
}

.switch a {
  color: #667eea;
  text-decoration: none;
  font-weight: bold;
  transition: color 0.3s ease;
}

.switch a:hover {
  color: #764ba2;
}

.tagline {
```

```
font-style: italic;
color: transparent;
background: linear-gradient(135deg, #667eea, #764ba2, #ff7e5f, #feb47b);
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;
background-size: 400% 400%;
animation: gradientAnimation 15s ease infinite;
margin-bottom: 30px;
}
```

```
.tab-icon {
width: 30px;
height: 30px;
background-image:
url('file:///D:/download/naheed/project_prime/monday-icon-svgrepo-com.svg');
background-repeat: no-repeat;
background-size: contain;
background-position: center;
display: inline-block;
margin-right: 8px;
animation: iconAnimation 2s linear infinite;
}
```

```
@keyframes iconAnimation {
0% {
transform: scale(1);
filter: hue-rotate(0deg);
}
50% {
transform: scale(1.1);
filter: hue-rotate(180deg);
}
100% {
transform: scale(1);
filter: hue-rotate(360deg);
}
```

```
}
```

13.Index.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Productivity Nexus</title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min
.css">
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fullcalendar/5.11.3/main.min.css
">
  <link rel="icon" type="image/svg+xml"
href="https://raw.githubusercontent.com/nahidibnahanarun/Productivity_Nex
us_2.0/47e576223ef09fbde6ed1faaf3264ddf69891722/monday-icon-svgrepo
-com.svg">
  <link rel="stylesheet" href="index.css">
</head>
<body>
  <div class="container">
    <div class="logo gradient-text"><i class="fas fa-bolt"></i></div>
    <h1 class="gradient-text">Productivity Nexus</h1>
    <p class="tagline">Unlock your potential, amplify your productivity</p>

    <div id="loginForm">
      <h2 class="gradient-text">Welcome Back</h2>
      <div class="input-group">
        <i class="fas fa-user"></i>
        <input type="text" id="loginUsername" placeholder="Username"
required class="input-animate">
      </div>
      <div class="input-group">
        <i class="fas fa-lock"></i>
        <input type="password" id="loginPassword"
placeholder="Password" required class="input-animate">

```

```

</div>
<button class="btn-animate" onclick="login()">Log In</button>
<p class="switch">New to Productivity Nexus? <a href="#"
onclick="showRegistration()">Sign Up</a></p>
</div>

<div id="registrationForm" style="display: none;">
  <h2 class="gradient-text">Join Us Today</h2>
  <div class="input-group">
    <i class="fas fa-user"></i>
    <input type="text" id="regName" placeholder="Full Name" required
class="input-animate">
  </div>
  <div class="input-group">
    <i class="fas fa-user-circle"></i>
    <input type="text" id="regUsername" placeholder="Username"
required class="input-animate">
  </div>
  <div class="input-group">
    <i class="fas fa-envelope"></i>
    <input type="email" id="regEmail" placeholder="Email" required
class="input-animate">
  </div>
  <div class="input-group">
    <i class="fas fa-lock"></i>
    <input type="password" id="regPassword" placeholder="Password"
required class="input-animate">
  </div>
  <button class="btn-animate" onclick="register()">Sign Up</button>
  <p class="switch">Already have an account? <a href="#"
onclick="showLogin()">Log In</a></p>
</div>

<div id="adminLoginForm" style="display: none;">
  <h2 class="gradient-text">Admin Login</h2>
  <div class="input-group">
    <i class="fas fa-user"></i>

```



```

        <input type="text" id="adminUsername" placeholder="Admin
Username" required class="input-animate">
    </div>
    <div class="input-group">
        <i class="fas fa-lock"></i>
        <input type="password" id="adminPassword" placeholder="Admin
Password" required class="input-animate">
    </div>
    <button class="btn-animate" onclick="adminLogin()">Log
In</button>
    <p class="switch">Back to User Login? <a href="#"
onclick="showLogin()">User Login</a></p>
</div>

    <div class="admin-link">
        <a href="#" onclick="showAdminLogin()">Admin Login</a>
    </div>

</div>
<script src="index.js"></script>
</body>
</html>

```

14.Index.js:

```

let users = JSON.parse(localStorage.getItem('users')) || [];

function showLogin() {
    document.getElementById('loginForm').style.display = 'block';
    document.getElementById('registrationForm').style.display = 'none';
    document.getElementById('adminLoginForm').style.display = 'none';
}

function showRegistration() {
    document.getElementById('loginForm').style.display = 'none';
    document.getElementById('registrationForm').style.display = 'block';
    document.getElementById('adminLoginForm').style.display = 'none';
}

```

```
}
```

```
function showAdminLogin() {  
    document.getElementById('loginForm').style.display = 'none';  
    document.getElementById('registrationForm').style.display = 'none';  
    document.getElementById('adminLoginForm').style.display = 'block';  
}
```

```
function login() {  
    const username = document.getElementById('loginUsername').value;  
    const password = document.getElementById('loginPassword').value;  
  
    const xhr = new XMLHttpRequest();  
    xhr.open("POST", "login.php", true);  
    xhr.setRequestHeader("Content-type",  
"application/x-www-form-urlencoded");  
  
    xhr.onreadystatechange = function() {  
        if (xhr.readyState == 4 && xhr.status == 200) {  
  
            if (xhr.responseText === "Login successful!") {  
                sessionStorage.setItem('logged_in', true);  
                sessionStorage.setItem('username', username);  
                window.location.href = 'dashboard.html';  
            } else {  
                alert(xhr.responseText);  
            }  
        } else {  
  
            alert(xhr.responseText);  
        }  
    };  
  
    xhr.send("username=" + username + "&password=" + password);  
}
```

```
function adminLogin() {
```

```

const adminUsername =
document.getElementById('adminUsername').value;
const adminPassword = document.getElementById('adminPassword').value;

const xhr = new XMLHttpRequest();
xhr.open("POST", "admin_login.php", true);
xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

xhr.onreadystatechange = function() {
    if (xhr.readyState == 4 && xhr.status == 200) {

        if (xhr.responseText === "Admin login successful!") {
            sessionStorage.setItem('admin_logged_in', true);
            window.location.href = 'admin.html';
        } else {
            alert(xhr.responseText);
        }
    } else {
        alert(xhr.responseText);
    }
};

xhr.send("adminUsername=" + adminUsername + "&adminPassword=" +
adminPassword);
}

function validateEmail(email) {
    const re = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    return re.test(String(email).toLowerCase());
}

function validatePassword(password) {
    const re = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}$/;
    return re.test(password);
}

```

```

function register() {
    const name = document.getElementById('regName').value;
    const username = document.getElementById('regUsername').value;
    const email = document.getElementById('regEmail').value;
    const password = document.getElementById('regPassword').value;

    if (!validateEmail(email)) {
        alert('Please enter a valid email address');
        return;
    }

    if (!validatePassword(password)) {
        alert('Password must be at least 8 characters long and contain at least one
number, one uppercase and one lowercase letter');
        return;
    }

    const xhr = new XMLHttpRequest();
    xhr.open("POST", "register.php", true);
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            alert(xhr.responseText);
            if (xhr.responseText === "Registration successful!") {
                showLogin();
            }
        } else {
            alert(xhr.responseText);
        }
    };

    xhr.send("name=" + name + "&username=" + username + "&email=" +
email + "&password=" + password);
}

```

```

showLogin();

document.addEventListener('keydown', function(event) {
    if (event.key === 'Enter') {
        if (document.getElementById('loginForm').style.display === 'block') {
            login();
        } else if (document.getElementById('registrationForm').style.display ===
'block') {
            register();
        } else if (document.getElementById('adminLoginForm').style.display ===
'block') {
            adminLogin();
        }
    }
});

```

15.Login.php:

```

<?php
// Include the database connection
include 'db_config.php';

// Check if the login form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST["username"];
    $password = $_POST["password"];

    // Sanitize input (important for security)
    $username = mysqli_real_escape_string($conn, $username);
    $password = mysqli_real_escape_string($conn, $password);

    // Query the database
    $sql = "SELECT * FROM users WHERE username = '$username' AND
password = '$password'";
    $result = mysqli_query($conn, $sql);

    if (mysqli_num_rows($result) == 1) {

```

```
// User found, redirect to the dashboard
echo "Login successful!";
} else {
    // Incorrect username or password
    echo "Invalid username or password.";
}
}
?>
```

16.Productivity_db.sql:

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Sep 10, 2024 at 06:28 PM
-- Server version: 10.4.32-MariaDB
-- PHP Version: 8.2.12
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `productivity_db`
--
```

```
--  
-- Table structure for table `users`  
--
```

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `username` varchar(255) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `password` varchar(255) NOT NULL,  
  `role` varchar(20) DEFAULT 'user'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_general_ci;
```

```
--  
-- Dumping data for table `users`  
--
```

```
INSERT INTO `users` (`id`, `name`, `username`, `email`, `password`, `role`)  
VALUES  
(1, 'test1', 'test1', 'n@gmail.com', 'test1', 'user'),  
(2, 'Admin User', 'admin', 'admin@example.com', 'admin', 'admin'),  
(4, 'test2', 'test2', 'n@gmail.com', 'test2', 'user'),  
(5, 'test3', 'test3', 'test@gmail.com', 'test3', 'user');
```

```
--  
-- Indexes for dumped tables  
--
```

```
--  
-- Indexes for table `users`  
--
```

```
ALTER TABLE `users`  
  ADD PRIMARY KEY (`id`),  
  ADD UNIQUE KEY `username` (`username`);
```

```
--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;
```

17.Register.php:

```
<?php
// Include the database connection
include 'db_config.php';

// Check if the registration form was submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST["name"];
    $username = $_POST["username"];
    $email = $_POST["email"];
    $password = $_POST["password"];

    // Sanitize input (important for security)
    $name = mysqli_real_escape_string($conn, $name);
    $username = mysqli_real_escape_string($conn, $username);
    $email = mysqli_real_escape_string($conn, $email);
    $password = mysqli_real_escape_string($conn, $password);

    // Check if the username already exists
```



```
$sql = "SELECT * FROM users WHERE username = '$username'";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    echo "Username already exists.";
} else {
    // Insert the new user into the database
    $sql = "INSERT INTO users (name, username, email, password) VALUES
('$name', '$username', '$email', '$password')";

    if (mysqli_query($conn, $sql)) {
        echo "Registration successful!";
    } else {
        echo "Error: " . mysqli_error($conn);
    }
}

mysqli_close($conn);
}
?>
```