Bunkify

Smart Attendance Tracker

by

MADHAV M NAMBIAR

Thrissur,Kerala | mnambiarmadhav@gmail.com | +916238781544
linkedin. madhav-m-nambiar/ | github.Madhav-M-Nambiar

Project Title: Bunkify – Smart Attendance Tracker

Project Overview

Bunkify is a web-based platform developed to assist students in managing and tracking their class attendance in a simple and efficient way. It is specifically designed to help users determine how many classes they can afford to miss—commonly referred to as "bunking"—without dropping below the minimum attendance requirement, which is typically 75% in many educational institutions. The website eliminates the need for manual calculations and provides students with an easy way to stay on top of their attendance goals. With a focus on usability and accessibility, Bunkify offers a practical solution for students who want to manage their schedules more effectively.

Technologies Used

Bunkify is built entirely using front-end web technologies, making it lightweight and easy to run directly in a browser. The structure of the website is created using HTML, styled with CSS, and all interactive functionality is handled through JavaScript. For data storage, Bunkify uses JavaScript's localStorage, allowing the website to save user input locally on the device without needing any server-side storage or database like MySQL. The use of PHP is for the current date and time. This makes the website easy to deploy and run without requiring a backend server or complex hosting setup.

Core Features and Functionalities

Bunkify includes a variety of useful features tailored to student needs. It provides a simulated login interface to give users a personalized experience, though no actual authentication system is implemented. Once logged in, users can add multiple subjects and manually input the total number of classes held and the number of classes attended for each subject. The website calculates the attendance percentage for every subject and determines how many classes the student can miss while staying above the required attendance threshold—or how many they need to attend to reach it.

How It Works

When a user accesses the website, they begin by simulating a login to access their attendance dashboard. From there, they can enter and manage attendance records for various subjects. The logic written in JavaScript performs real-time calculations to determine the user's attendance percentage and provides an overview of how many more classes can be missed safely. Since all data is stored using localStorage, the information is saved locally in the user's browser, allowing them to return later and continue where they left off—provided they use the same device and browser.

User Experience and Workflow

The website is designed with simplicity and clarity in mind. After logging in, users are guided through a straightforward interface where they can add or remove subjects, input attendance data, and instantly view calculated results. The layout is clean and minimal, ensuring users can quickly find and understand the information they need. The website responds immediately to user inputs without page reloads, providing a smooth experience through JavaScript's real-time data handling.

Potential for Future Development

Although the current version of Bunkify is built entirely on the front end, it has considerable potential for future upgrades. Enhancements may include adding real user authentication and cloud-based storage for syncing data across devices. A mobile-friendly version or dedicated mobile app could further increase accessibility. Other ideas include integrating visual charts for attendance trends, customizable attendance thresholds, semester-wise summaries, or even institutional portal integration for automatic data input.

Conclusion

Bunkify is a lightweight and functional website created to solve a very real problem faced by students: tracking attendance and understanding how much flexibility they have with class attendance. Built using HTML, CSS, JavaScript, and basic PHP, it demonstrates how powerful client-side web development can be in building helpful tools without the need for backend complexity. Bunkify not only makes attendance management easier but also showcases practical web development skills in a real-world context.

HTML CODE (PHP):

```
<?php
$currentDate = date("l, F j, Y");
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Bunkify - Attendance Calculator</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.4/css/all.min.css">
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="app-container" id="app-container">
        <div class="current-date">
            <p>Today is: <?php echo $currentDate; ?></p>
        </div>
        <div id="auth-page" class="auth-container">
            <div class="auth-tabs">
                <div class="auth-tab active" data-tab="login">Login</div>
                <div class="auth-tab" data-tab="register">Register</div>
            </div>

            <div id="login-form" class="auth-form">
                <h2 class="auth-heading">Welcome Back to Bunkify</h2>
                <form id="login-form-element">
                    <div class="form-group">
                        <label for="login-email">Email</label>
                        <input type="email" id="login-email" name="email"
class="form-control" placeholder="your@email.com" required>
                    </div>
                    <div class="form-group">
                        <label for="login-password">Password</label>
                        <input type="password" id="login-password"
name="password" class="form-control" placeholder="Enter your
password" required>
                    </div>
                    <div class="form-group">
                        <button type="submit" class="btn btn-primary"
style="width: 100%;">Login</button>
                    </div>
                </form>
            </div>

            <div id="register-form" class="auth-form" style="display:
none;">
                <h2 class="auth-heading">Create Your Bunkify Account</h2>
                <form id="register-form-element">
                    <div class="form-group">
                        <label for="register-name">Full Name</label>
                        <input type="text" id="register-name" name="name"
class="form-control" placeholder="John Doe" required>
                    </div>
                    <div class="form-group">
                        <label for="register-email">Email</label>
                        <input type="email" id="register-email" name="email"
class="form-control" placeholder="your@email.com" required>
                    </div>
                    <div class="form-group">
                        <label for="register-password">Password</label>
                        <input type="password" id="register-password"
name="password" class="form-control" placeholder="Create a password"
required>
                    </div>
                    <div class="form-group">
                        <label for="register-confirm-password">Confirm
Password</label>
                        <input type="password" id="register-confirm-password"
name="confirm_password" class="form-control" placeholder="Confirm
your password" required>
                    </div>
                    <div class="form-group">
                        <button type="submit" class="btn btn-primary"
style="width: 100%;">Create Account</button>
                    </div>
                </form>
            </div>
        </div>

        <div id="app" class="hidden">
            <header>
                <div class="logo">
                    <i class="fas fa-calendar-check"></i>
                    <span>Bunkify</span>
                </div>
                <div class="header-actions">
                    <button class="theme-toggle" id="theme-toggle">
                        <i class="fas fa-moon"></i>
                    </button>
                    <div class="user-menu">
                        <div class="user-icon" id="user-menu-toggle">
                            <i class="fas fa-user"></i>
                        </div>
                        <div class="dropdown-menu" id="user-dropdown">
                            <ul>
                                <li><a href="#" id="profile-link"><i class="fas fa-user-circle"></i> Profile</a></li>
                                <li><a href="#" id="settings-link"><i class="fas fa-cog"></i> Settings</a></li>
                                <li><a href="#" id="logout-link"><i class="fas fa-sign-out-alt"></i> Logout</a></li>
                            </ul>
                        </div>
                    </div>
                </div>
            </header>

            <div class="container">
                <div class="stats-container">
                    <div class="stat-card">
                        <div class="icon"><i class="fas fa-book"></i></div>
                        <div class="value" id="total-subjects">0</div>
                        <div class="label">Subjects</div>
                    </div>
                    <div class="stat-card">
                        <div class="icon"><i class="fas fa-check-circle"></i></div>
                        <div class="value" id="total-attended">0</div>
                        <div class="label">Classes Attended</div>
                    </div>
                    <div class="stat-card">
                        <div class="icon"><i class="fas fa-times-
    circle"></i></div>
                        <div class="value" id="total-bunked">0</div>
                        <div class="label">Classes Bunked</div>
                    </div>
                    <div class="stat-card">
                        <div class="icon"><i class="fas fa-
    percentage"></i></div>
                        <div class="value" id="average-attendance">0%</div>
                        <div class="label">Avg. Attendance</div>
                    </div>
                </div>

                <div class="add-subject">
                    <h3 class="form-title">Add New Subject</h3>
                    <form id="add-subject-form">
                        <div class="form-row">
                            <div class="form-group">
                                <label for="subject-name">Subject Name</label>
                                <input type="text" id="subject-name"
    name="subject_name" class="form-control" placeholder="e.g.
    Mathematics" required>
                            </div>
                            <div class="form-group">
                                <label for="subject-code">Subject Code</label>
                                <input type="text" id="subject-code"
    name="subject_code" class="form-control" placeholder="e.g.
    MATH101" required>
                            </div>
                        </div>
                        <div class="form-row">
                            <div class="form-group">
                                <label for="subject-classes">Initial Classes
    Attended</label>
                                <input type="number" id="subject-classes"
    name="classes_attended" class="form-control" value="0" min="0">
                            </div>
                            <div class="form-group">
                                <label for="total-classes">Total Classes</label>
                                <input type="number" id="total-classes"
    name="total_classes" class="form-control" value="0" min="0">
```

```
                        </div>
                    </div>
                    <div class="form-row">
                        <div class="form-group">
                            <label for="min-attendance">Minimum Attendance
Required (%)</label>
                            <input type="number" id="min-attendance"
name="min_attendance" class="form-control" value="75" min="0"
max="100">
                        </div>
                    </div>
                    <div class="form-group">
                        <button type="submit" class="btn btn-primary"
style="width: 100%;">Add Subject</button>
                    </div>
                </form>
            </div>

            <div class="calendar">
                <div class="calendar-header">
                    <h3 class="calendar-title" id="calendar-month-
year">April 2025</h3>
                    <div class="calendar-nav">
                        <button id="prev-month"><i class="fas fa-chevron-
left"></i></button>
                        <button id="next-month"><i class="fas fa-chevron-
right"></i></button>
                    </div>
                </div>
                <div class="calendar-day-headers">
                    <div>Sun</div>
                    <div>Mon</div>
                    <div>Tue</div>
                    <div>Wed</div>
                    <div>Thu</div>
                    <div>Fri</div>
                    <div>Sat</div>
                </div>
                <div class="calendar-grid" id="calendar-grid"></div>
            </div>

            <h3 class="section-title">Your Subjects</h3>
            <div class="dashboard" id="subjects-dashboard"></div>
```

```
        </div>
      </div>
    </div>

    <div class="modal-backdrop hidden" id="edit-subject-modal">
      <div class="modal">
        <div class="modal-header">
          <h3 class="modal-title">Edit Subject</h3>
          <button class="modal-close" data-
dismiss="modal">&times;</button>
        </div>
        <div class="modal-body">
          <form id="edit-subject-form">
            <div class="form-group">
              <label for="edit-subject-name">Subject Name</label>
              <input type="text" id="edit-subject-name"
name="subject_name" class="form-control" required>
            </div>
            <div class="form-group">
              <label for="edit-subject-code">Subject Code</label>
              <input type="text" id="edit-subject-code"
name="subject_code" class="form-control" required>
            </div>
            <div class="form-row">
              <div class="form-group">
                <label for="edit-subject-classes">Classes
Attended</label>
                <input type="number" id="edit-subject-classes"
name="classes_attended" class="form-control" min="0">
              </div>
              <div class="form-group">
                <label for="edit-total-classes">Total Classes</label>
                <input type="number" id="edit-total-classes"
name="total_classes" class="form-control" min="0">
              </div>
            </div>
            <div class="form-group">
              <label for="edit-min-attendance">Minimum Attendance
Required (%)</label>
              <input type="number" id="edit-min-attendance"
name="min_attendance" class="form-control" min="0" max="100">
            </div>
            <input type="hidden" id="edit-subject-id"
name="subject_id">
          </form>
        </div>
        <div class="modal-footer">
          <button class="btn btn-outline" data-
dismiss="modal">Cancel</button>
          <button type="button" id="save-edit-subject" class="btn btn-
primary">Save Changes</button>
        </div>
      </div>
    </div>

    <div class="modal-backdrop hidden" id="confirm-delete-modal">
      <div class="modal">
        <div class="modal-header">
          <h3 class="modal-title">Confirm Delete</h3>
          <button class="modal-close" data-
dismiss="modal">&times;</button>
        </div>
        <div class="modal-body">
          <p>Are you sure you want to delete this subject? This action
cannot be undone.</p>
          <input type="hidden" id="delete-subject-id"
name="subject_id">
        </div>
        <div class="modal-footer">
          <button class="btn btn-outline" data-
dismiss="modal">Cancel</button>
          <button type="button" id="confirm-delete-btn" class="btn
btn-danger">Delete</button>
        </div>
      </div>
    </div>

    <div class="modal-backdrop hidden" id="attendance-modal">
      <div class="modal">
        <div class="modal-header">
          <h3 class="modal-title">Mark Attendance</h3>
          <button class="modal-close" data-
dismiss="modal">&times;</button>
        </div>
        <div class="modal-body">
```

```
          <form id="attendance-form">
            <div class="form-group">
              <label for="attendance-subject">Subject</label>
              <select id="attendance-subject" name="subject_id"
class="form-control" required></select>
            </div>
            <div class="form-group">
              <label for="attendance-date">Date</label>
              <input type="date" id="attendance-date" name="date"
class="form-control" required>
            </div>
            <div class="form-group">
              <label>Status</label>
              <div class="radio-group">
                <label class="radio-label">
                  <input type="radio" name="status"
value="attended" checked>
                  <span>Attended</span>
                </label>
                <label class="radio-label">
                  <input type="radio" name="status" value="missed">
                  <span>Missed</span>
                </label>
              </div>
            </div>
          </form>
        </div>
        <div class="modal-footer">
          <button class="btn btn-outline" data-
dismiss="modal">Cancel</button>
          <button type="button" id="save-attendance" class="btn btn-
primary">Save</button>
        </div>
      </div>
    </div>

    <div class="modal-backdrop hidden" id="bunk-calculator-modal">
      <div class="modal">
        <div class="modal-header">
          <h3 class="modal-title">Bunk Calculator</h3>
          <button class="modal-close" data-
dismiss="modal">&times;</button>
        </div>
```

```
            <div class="modal-body">
              <div id="bunk-calculator-content">
                <div class="bunk-info">
                  <p>Subject: <strong id="calc-subject-name"></strong></p>
                  <p>Current Attendance: <strong id="calc-current-attendance">0%</strong></p>
                  <p>Minimum Required: <strong id="calc-min-required">75%</strong></p>
                </div>
                <div class="bunk-predictions">
                  <div class="prediction-item">
                    <h4>Safe to Bunk</h4>
                    <div class="prediction-value" id="safe-to-bunk">0 classes</div>
                  </div>
                  <div class="prediction-item">
                    <h4>Must Attend</h4>
                    <div class="prediction-value" id="must-attend">0 classes</div>
                  </div>
                </div>
              </div>
            </div>
            <div class="modal-footer">
              <button class="btn btn-primary" data-dismiss="modal">Close</button>
            </div>
          </div>
        </div>

        <div class="modal-backdrop hidden" id="profile-modal">
          <div class="modal">
            <div class="modal-header">
              <h3 class="modal-title">User Profile</h3>
              <button class="modal-close" data-dismiss="modal">&times;</button>
            </div>
            <div class="modal-body">
              <div class="profile-info">
                <div class="profile-picture">
                  <i class="fas fa-user-circle"></i>
                </div>
```

```
           <div class="profile-details">
             <h3 id="profile-name">User Name</h3>
             <p id="profile-email">user@example.com</p>
           </div>
         </div>
         <form id="profile-form">
           <div class="form-group">
             <label for="profile-update-name">Full Name</label>
             <input type="text" id="profile-update-name"
name="name" class="form-control">
           </div>
           <div class="form-group">
             <label for="profile-update-email">Email</label>
             <input type="email" id="profile-update-email"
name="email" class="form-control">
           </div>
         </form>
       </div>
       <div class="modal-footer">
         <button class="btn btn-outline" data-
dismiss="modal">Cancel</button>
         <button type="button" id="save-profile" class="btn btn-
primary">Save Changes</button>
       </div>
     </div>
   </div>

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

CSS CODE:

```
:root {
   --primary-color: #4361ee;
   --secondary-color: #3f37c9;
   --success-color: #4cc9f0;
   --danger-color: #f72585;
   --warning-color: #f8961e;
   --light-color: #f8f9fa;
   --dark-color: #212529;
   --gray-color: #6c757d;
   --light-gray: #e9ecef;
   --border-radius: 8px;
   --card-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
   --transition: all 0.3s ease;
}

.dark-theme {
   --primary-color: #7209b7;
   --secondary-color: #560bad;
   --light-color: #2b2d42;
   --dark-color: #edf2f4;
   --light-gray: #333;
   --card-shadow: 0 4px 6px rgba(0, 0, 0, 0.3);
}

* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
}

body {
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
   line-height: 1.6;
   color: var(--dark-color);
   background-color: #f5f7fb;
   transition: var(--transition);
```

```
}

body.dark-mode {
   background-color: #121212;
   color: var(--light-color);
}

.hidden {
   display: none !important;
}

.app-container {
   min-height: 100vh;
   display: flex;
   flex-direction: column;
}

.container {
   max-width: 1200px;
   margin: 0 auto;
   padding: 20px;
}

/* Button Styles */
.btn {
   display: inline-block;
   padding: 10px 20px;
   border: none;
   border-radius: var(--border-radius);
   cursor: pointer;
   font-size: 16px;
   font-weight: 500;
   text-align: center;
   transition: var(--transition);
}

.btn-primary {
   background-color: var(--primary-color);
   color: white;
}

.btn-primary:hover {
   background-color: var(--secondary-color);
```

```
}

.btn-outline {
    background-color: transparent;
    border: 1px solid var(--gray-color);
    color: var(--gray-color);
}

.btn-outline:hover {
    background-color: var(--light-gray);
}

.btn-danger {
    background-color: var(--danger-color);
    color: white;
}

.btn-danger:hover {
    background-color: #d00f5f;
}

/* Form Styles */
.form-group {
    margin-bottom: 20px;
}

.form-row {
    display: flex;
    gap: 15px;
}

.form-row .form-group {
    flex: 1;
}

label {
    display: block;
    margin-bottom: 8px;
    font-weight: 500;
}

.form-control {
    width: 100%;
```

```
   padding: 10px 15px;
   border: 1px solid var(--light-gray);
   border-radius: var(--border-radius);
   font-size: 16px;
   transition: var(--transition);
}

.form-control:focus {
   outline: none;
   border-color: var(--primary-color);
   box-shadow: 0 0 0 3px rgba(67, 97, 238, 0.1);
}

.radio-group {
   display: flex;
   gap: 15px;
}

.radio-label {
   display: flex;
   align-items: center;
   gap: 5px;
   cursor: pointer;
}

/* Authentication Styles */
.auth-container {
   max-width: 500px;
   margin: 50px auto;
   background-color: white;
   border-radius: var(--border-radius);
   box-shadow: var(--card-shadow);
   overflow: hidden;
}

body.dark-mode .auth-container {
   background-color: #1e1e1e;
}

.auth-tabs {
   display: flex;
   border-bottom: 1px solid var(--light-gray);
}
```

```
.auth-tab {
    flex: 1;
    padding: 15px;
    text-align: center;
    cursor: pointer;
    font-weight: 500;
    transition: var(--transition);
}

.auth-tab.active {
    background-color: var(--primary-color);
    color: white;
}

.auth-form {
    padding: 25px;
}

.auth-heading {
    margin-bottom: 20px;
    text-align: center;
    color: var(--primary-color);
}

/* Header Styles */
header {
    background-color: white;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    padding: 15px 30px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

body.dark-mode header {
    background-color: #1e1e1e;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);
}

.logo {
    display: flex;
    align-items: center;
    gap: 10px;
    font-size: 24px;
    font-weight: 700;
    color: var(--primary-color);
}

.header-actions {
    display: flex;
    align-items: center;
    gap: 20px;
}

.theme-toggle {
    background: none;
    border: none;
    font-size: 20px;
    color: var(--dark-color);
    cursor: pointer;
}

body.dark-mode .theme-toggle {
    color: var(--light-color);
}

.user-menu {
    position: relative;
}

.user-icon {
    width: 36px;
    height: 36px;
    border-radius: 50%;
    background-color: var(--primary-color);
    color: white;
    display: flex;
    justify-content: center;
    align-items: center;
    cursor: pointer;
    font-size: 18px;
}

.dropdown-menu {
    position: absolute;
    top: 45px;
    right: 0;
    background-color: white;
    border-radius: var(--border-radius);
    box-shadow: var(--card-shadow);
    width: 180px;
    display: none;
    z-index: 100;
}

body.dark-mode .dropdown-menu {
    background-color: #1e1e1e;
}

.dropdown-menu.active {
    display: block;
}

.dropdown-menu ul {
    list-style: none;
}

.dropdown-menu ul li a {
    display: flex;
    align-items: center;
    padding: 12px 15px;
    text-decoration: none;
    color: var(--dark-color);
    gap: 8px;
    transition: var(--transition);
}

body.dark-mode .dropdown-menu ul li a {
    color: var(--light-color);
}

.dropdown-menu ul li a:hover {
    background-color: var(--light-gray);
}

/* Stats Container Styles */
.stats-container {
    display: grid;
```

```
   grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
   gap: 20px;
   margin-bottom: 30px;
}

.stat-card {
   background-color: white;
   border-radius: var(--border-radius);
   padding: 20px;
   display: flex;
   flex-direction: column;
   align-items: center;
   box-shadow: var(--card-shadow);
   transition: var(--transition);
}

body.dark-mode .stat-card {
   background-color: #1e1e1e;
}

.stat-card:hover {
   transform: translateY(-5px);
   box-shadow: 0 8px 15px rgba(0, 0, 0, 0.1);
}

.stat-card .icon {
   font-size: 24px;
   color: var(--primary-color);
   margin-bottom: 10px;
}

.stat-card .value {
   font-size: 28px;
   font-weight: 700;
   margin-bottom: 5px;
}

.stat-card .label {
   font-size: 14px;
   color: var(--gray-color);
}

/* Add Subject Form Styles */
```

```
}

.add-subject {
   background-color: white;
   border-radius: var(--border-radius);
   padding: 25px;
   margin-bottom: 30px;
   box-shadow: var(--card-shadow);
}

body.dark-mode .add-subject {
   background-color: #1e1e1e;
}

.form-title {
   margin-bottom: 20px;
   color: var(--primary-color);
}

/* Calendar Styles */
.calendar {
   background-color: white;
   border-radius: var(--border-radius);
   padding: 25px;
   margin-bottom: 30px;
   box-shadow: var(--card-shadow);
}

body.dark-mode .calendar {
   background-color: #1e1e1e;
}

.calendar-header {
   display: flex;
   justify-content: space-between;
   align-items: center;
   margin-bottom: 20px;
}

.calendar-nav {
   display: flex;
   gap: 10px;
}

.calendar-nav button {
   width: 32px;
   height: 32px;
   border-radius: 50%;
   border: none;
   background-color: var(--light-gray);
   display: flex;
   justify-content: center;
   align-items: center;
   cursor: pointer;
   transition: var(--transition);
}

.calendar-nav button:hover {
   background-color: var(--primary-color);
   color: white;
}

.calendar-day-headers {
   display: grid;
   grid-template-columns: repeat(7, 1fr);
   text-align: center;
   margin-bottom: 10px;
   font-weight: 500;
}

.calendar-grid {
   display: grid;
   grid-template-columns: repeat(7, 1fr);
   gap: 5px;
}

.calendar-day {
   aspect-ratio: 1;
   border: 1px solid var(--light-gray);
   border-radius: 4px;
   display: flex;
   justify-content: center;
   align-items: center;
   cursor: pointer;
   transition: var(--transition);
   position: relative;
}
```

```
.calendar-day:hover {
   background-color: var(--light-gray);
}

.calendar-day.inactive {
   opacity: 0.3;
}

.calendar-day.today {
   background-color: var(--primary-color);
   color: white;
}

.calendar-day .attendance-dot {
   position: absolute;
   bottom: 4px;
   width: 6px;
   height: 6px;
   border-radius: 50%;
}

.attendance-dot.attended {
   background-color: var(--success-color);
}

.attendance-dot.missed {
   background-color: var(--danger-color);
}

/* Dashboard Styles */
.section-title {
   margin: 1.5rem 0 1rem;
   color: var(--primary-color);
}

.dashboard {
   display: grid;
   grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
   gap: 20px;
   margin-bottom: 30px;
}

.subject-card {
    background-color: white;
    border-radius: var(--border-radius);
    overflow: hidden;
    box-shadow: var(--card-shadow);
    transition: var(--transition);
}

body.dark-mode .subject-card {
    background-color: #1e1e1e;
}

.subject-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 8px 15px rgba(0, 0, 0, 0.1);
}

.subject-header {
    background-color: var(--primary-color);
    color: white;
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.subject-name {
    font-weight: 700;
    font-size: 18px;
}

.subject-code {
    font-size: 14px;
    opacity: 0.8;
}

.subject-content {
    padding: 20px;
}

.attendance-info {
    display: flex;
    justify-content: space-between;
    margin-bottom: 15px;
```

```
}

.attendance-item {
   text-align: center;
}

.attendance-value {
   font-size: 24px;
   font-weight: 700;
   margin-bottom: 5px;
}

.attendance-label {
   font-size: 14px;
   color: var(--gray-color);
}

.attendance-progress {
   height: 10px;
   background-color: var(--light-gray);
   border-radius: 5px;
   overflow: hidden;
   margin-bottom: 15px;
}

.progress-bar {
   height: 100%;
   background-color: var(--success-color);}
```

JS CODE:

```
const authPage = document.getElementById('auth-page');
const appPage = document.getElementById('app');
const loginForm = document.getElementById('login-form');
const registerForm = document.getElementById('register-form');
const authTabs = document.querySelectorAll('.auth-tab');
const loginFormElement = document.getElementById('login-form-element');
const registerFormElement = document.getElementById('register-form-element');
const userMenuToggle = document.getElementById('user-menu-toggle');
const userDropdown = document.getElementById('user-dropdown');
const themeToggle = document.getElementById('theme-toggle');
const addSubjectForm = document.getElementById('add-subject-form');
const subjectsDashboard = document.getElementById('subjects-dashboard');
const logoutLink = document.getElementById('logout-link');
const profileLink = document.getElementById('profile-link');
const settingsLink = document.getElementById('settings-link');
const editSubjectModal = document.getElementById('edit-subject-modal');
const confirmDeleteModal = document.getElementById('confirm-delete-modal');
const attendanceModal = document.getElementById('attendance-modal');
const bunkCalculatorModal = document.getElementById('bunk-calculator-modal');
const profileModal = document.getElementById('profile-modal');
const saveEditSubjectBtn = document.getElementById('save-edit-subject');
const confirmDeleteBtn = document.getElementById('confirm-delete-btn');
const saveAttendanceBtn = document.getElementById('save-attendance');
const saveProfileBtn = document.getElementById('save-profile');
const totalSubjectsElement = document.getElementById('total-subjects');
const totalAttendedElement = document.getElementById('total-attended');
const totalBunkedElement = document.getElementById('total-bunked');
const averageAttendanceElement = document.getElementById('average-
attendance');
const calendarGrid = document.getElementById('calendar-grid');
const prevMonthBtn = document.getElementById('prev-month');
const nextMonthBtn = document.getElementById('next-month');
const calendarMonthYearElement =
document.getElementById('calendar-month-year');

let currentUser = null;
let subjects = [];
let attendance = [];
let currentDate = new Date();
let currentMonth = currentDate.getMonth();
let currentYear = currentDate.getFullYear();

themeToggle.addEventListener('click', () => {
  document.body.classList.toggle('dark-mode');
  const isDarkMode = document.body.classList.contains('dark-mode');
  localStorage.setItem('darkMode', isDarkMode);

  if (isDarkMode) {
    themeToggle.innerHTML = '<i class="fas fa-sun"></i>';
  } else {
    themeToggle.innerHTML = '<i class="fas fa-moon"></i>';
  }
});

const initTheme = () => {
  const isDarkMode = localStorage.getItem('darkMode') === 'true';
  if (isDarkMode) {
    document.body.classList.add('dark-mode');
    themeToggle.innerHTML = '<i class="fas fa-sun"></i>';
  }
};

authTabs.forEach(tab => {
  tab.addEventListener('click', () => {
    authTabs.forEach(t => t.classList.remove('active'));
    tab.classList.add('active');

    if (tab.dataset.tab === 'login') {
      loginForm.style.display = 'block';
      registerForm.style.display = 'none';
    } else {
      loginForm.style.display = 'none';
      registerForm.style.display = 'block';
    }
  });
});

userMenuToggle.addEventListener('click', () => {
  userDropdown.classList.toggle('active');
});

document.addEventListener('click', (e) => {
  if (!userMenuToggle.contains(e.target)
&& !userDropdown.contains(e.target)) {
    userDropdown.classList.remove('active');
  }
});

document.querySelectorAll('.modal-close, [data-
dismiss="modal"]').forEach(button => {
  button.addEventListener('click', () => {
    document.querySelectorAll('.modal-backdrop').forEach(modal => {
      modal.classList.add('hidden');
    });
  });
});

loginFormElement.addEventListener('submit', (e) => {
  e.preventDefault();
  const email = document.getElementById('login-email').value;
  const password = document.getElementById('login-password').value;

  const users = JSON.parse(localStorage.getItem('bunkifyUsers') || '[]');
  const user = users.find(u => u.email === email && u.password ===
password);

  if (user) {
    currentUser = user;
    localStorage.setItem('bunkifyCurrentUser', JSON.stringify(user));
    showApp();
    loadUserData();
  } else {
    alert('Invalid email or password');
  }
});

registerFormElement.addEventListener('submit', (e) => {
  e.preventDefault();
  const name = document.getElementById('register-name').value;
  const email = document.getElementById('register-email').value;
  const password = document.getElementById('register-password').value;
  const confirmPassword = document.getElementById('register-confirm-password').value;

  if (password !== confirmPassword) {
    alert('Passwords do not match');
    return;
  }

  const users = JSON.parse(localStorage.getItem('bunkifyUsers') || '[]');
  if (users.some(u => u.email === email)) {
    alert('Email already registered');
    return;
  }

  const newUser = {
    id: Date.now().toString(),
    name,
    email,
    password
  };

  users.push(newUser);
  localStorage.setItem('bunkifyUsers', JSON.stringify(users));

  currentUser = newUser;
  localStorage.setItem('bunkifyCurrentUser', JSON.stringify(newUser));
  showApp();

  registerFormElement.reset();

  initUserData();
});

logoutLink.addEventListener('click', (e) => {
  e.preventDefault();
```

```
    localStorage.removeItem('bunkifyCurrentUser');
    currentUser = null;
    subjects = [];
    attendance = [];
    showAuth();
});

profileLink.addEventListener('click', (e) => {
    e.preventDefault();
    document.getElementById('profile-name').textContent =
currentUser.name;
    document.getElementById('profile-email').textContent =
currentUser.email;
    document.getElementById('profile-update-name').value =
currentUser.name;
    document.getElementById('profile-update-email').value =
currentUser.email;
    profileModal.classList.remove('hidden');
});

saveProfileBtn.addEventListener('click', () => {
    const name = document.getElementById('profile-update-name').value;
    const email = document.getElementById('profile-update-email').value;

    if (!name || !email) {
        alert('Name and email are required');
        return;
    }

    const users = JSON.parse(localStorage.getItem('bunkifyUsers') || '[]');
    const userIndex = users.findIndex(u => u.id === currentUser.id);

    if (userIndex !== -1) {
        users[userIndex].name = name;
        users[userIndex].email = email;
        localStorage.setItem('bunkifyUsers', JSON.stringify(users));

        currentUser.name = name;
        currentUser.email = email;
        localStorage.setItem('bunkifyCurrentUser',
JSON.stringify(currentUser));

        profileModal.classList.add('hidden');
    }
});

addSubjectForm.addEventListener('submit', (e) => {
    e.preventDefault();
    const subjectName = document.getElementById('subject-name').value;
    const subjectCode = document.getElementById('subject-code').value;
    const classesAttended = parseInt(document.getElementById('subject-
classes').value) || 0;
    const totalClasses = parseInt(document.getElementById('total-
classes').value) || 0;
    const minAttendance = parseInt(document.getElementById('min-
attendance').value) || 75;

    const newSubject = {
        id: Date.now().toString(),
        name: subjectName,
        code: subjectCode,
        classesAttended,
        totalClasses,
        minAttendance,
        userId: currentUser.id
    };

    subjects.push(newSubject);
    saveSubjects();
    renderSubjects();
    updateStats();

    addSubjectForm.reset();
    document.getElementById('min-attendance').value = '75';
});

saveEditSubjectBtn.addEventListener('click', () => {
    const subjectId = document.getElementById('edit-subject-id').value;
    const subjectName = document.getElementById('edit-subject-
name').value;
    const subjectCode = document.getElementById('edit-subject-
code').value;
    const classesAttended = parseInt(document.getElementById('edit-
subject-classes').value) || 0;
    const totalClasses = parseInt(document.getElementById('edit-total-
classes').value) || 0;
```

```
  const minAttendance = parseInt(document.getElementById('edit-min-
attendance').value) || 75;

  const subjectIndex = subjects.findIndex(s => s.id === subjectId);

  if (subjectIndex !== -1) {
    subjects[subjectIndex].name = subjectName;
    subjects[subjectIndex].code = subjectCode;
    subjects[subjectIndex].classesAttended = classesAttended;
    subjects[subjectIndex].totalClasses = totalClasses;
    subjects[subjectIndex].minAttendance = minAttendance;

    saveSubjects();
    renderSubjects();
    updateStats();

    editSubjectModal.classList.add('hidden');
  }
});

confirmDeleteBtn.addEventListener('click', () => {
  const subjectId = document.getElementById('delete-subject-id').value;

  subjects = subjects.filter(s => s.id !== subjectId);

  attendance = attendance.filter(a => a.subjectId !== subjectId);

  saveSubjects();
  saveAttendance();
  renderSubjects();
  renderCalendar();
  updateStats();

  confirmDeleteModal.classList.add('hidden');
});

saveAttendanceBtn.addEventListener('click', () => {
  const subjectId = document.getElementById('attendance-
subject').value;
  const date = document.getElementById('attendance-date').value;
  const status =
document.querySelector('input[name="status"]:checked').value;
```

```
    if (!subjectId || !date) {
        alert('Please fill all fields');
        return;
    }

    const existingIndex = attendance.findIndex(a =>
        a.subjectId === subjectId && a.date === date);

    if (existingIndex !== -1) {
        attendance[existingIndex].status = status;
    } else {
        attendance.push({
            id: Date.now().toString(),
            subjectId,
            date,
            status,
            userId: currentUser.id
        });
    }

    updateSubjectAttendance(subjectId);

    saveAttendance();
    renderSubjects();
    renderCalendar();
    updateStats();

    attendanceModal.classList.add('hidden');
});

prevMonthBtn.addEventListener('click', () => {
    currentMonth--;
    if (currentMonth < 0) {
        currentMonth = 11;
        currentYear--;
    }
    renderCalendar();
});

nextMonthBtn.addEventListener('click', () => {
    currentMonth++;
    if (currentMonth > 11) {
        currentMonth = 0;
        currentYear++;
    }
    renderCalendar();
});

function renderCalendar() {
    const firstDay = new Date(currentYear, currentMonth, 1);
    const lastDay = new Date(currentYear, currentMonth + 1, 0);
    const daysInMonth = lastDay.getDate();
    const startDayOfWeek = firstDay.getDay();

    const monthNames = [
        'January', 'February', 'March', 'April', 'May', 'June',
        'July', 'August', 'September', 'October', 'November', 'December'
    ];
    calendarMonthYearElement.textContent =
`${monthNames[currentMonth]} ${currentYear}`;

    calendarGrid.innerHTML = '';

    for (let i = 0; i < startDayOfWeek; i++) {
        const dayElement = document.createElement('div');
        dayElement.className = 'calendar-day inactive';
        calendarGrid.appendChild(dayElement);
    }

    const today = new Date();
    const isCurrentMonth = today.getMonth() === currentMonth &&
today.getFullYear() === currentYear;

    for (let day = 1; day <= daysInMonth; day++) {
        const dayElement = document.createElement('div');
        dayElement.className = 'calendar-day';
        dayElement.textContent = day;

        if (isCurrentMonth && day === today.getDate()) {
            dayElement.classList.add('today');
        }

        const formattedDate = `${currentYear}-${String(currentMonth +
1).padStart(2, '0')}-${String(day).padStart(2, '0')}`;
```

```
    const dayAttendance = attendance.filter(a => a.date ===
formattedDate);

    if (dayAttendance.length > 0) {
      const attended = dayAttendance.filter(a => a.status ===
'attended').length;
      const missed = dayAttendance.filter(a => a.status ===
'missed').length;

      if (attended > 0) {
        const dot = document.createElement('div');
        dot.className = 'attendance-dot attended';
        dayElement.appendChild(dot);
      }

      if (missed > 0) {
        const dot = document.createElement('div');
        dot.className = 'attendance-dot missed';
        dot.style.right = '4px';
        dayElement.appendChild(dot);
      }
    }

    dayElement.addEventListener('click', () => {
      const dateInput = document.getElementById('attendance-date');
      dateInput.value = formattedDate;

      const subjectSelect = document.getElementById('attendance-
subject');
      subjectSelect.innerHTML = '';
      subjects.forEach(subject => {
        const option = document.createElement('option');
        option.value = subject.id;
        option.textContent = subject.name;
        subjectSelect.appendChild(option);
      });

      attendanceModal.classList.remove('hidden');
    });

    calendarGrid.appendChild(dayElement);
  }
}
```

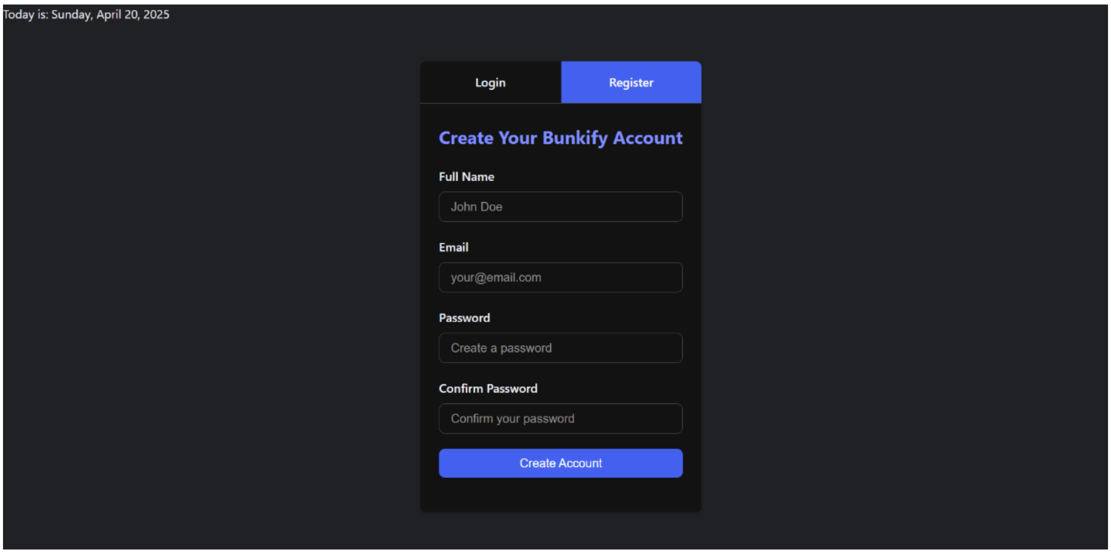
```
function renderSubjects() {
    subjectsDashboard.innerHTML = '';

    if (subjects.length === 0) {
        const message = document.createElement('div');
        message.className = 'no-subjects-message';
        message.textContent = 'You haven\'t added any subjects yet.';
        subjectsDashboard.appendChild(message);
        return;
    }

    subjects.forEach(subject => {
        const attendancePercentage = subject.totalClasses > 0
            ? Math.round((subject.classesAttended / subject.totalClasses) *
100)
            : 0;

        const subjectCard = document.createElement('div');
        subjectCard.className = 'subject-card';

        const headerColor = attendancePercentage >=
subject.minAttendance
            ? 'var(--success-color)'
            : 'var(--danger-color)';

        subjectCard.innerHTML = `
            <div class="subject-header" style="background-color:
${headerColor}">
                <div>
                    <div class="subject-name">${subject.name}</div>
                    <div class="subject-code">${subject.code}</div>
                </div>
                <div class="subject-actions">
                    <button class="btn-icon edit-subject" data-
id="${subject.id}"><i class="fas fa-edit"></i></button>
                    <button class="btn-icon delete-subject" data-
id="${subject.id}"><i class="fas fa-trash"></i></button>
                </div>
            </div>
            <div class="subject-content">
                <div class="attendance-info">
                    <div class="attendance-item">
```

```
                        <div class="attendance-
value">${subject.classesAttended}</div>
                        <div class="attendance-label">Attended</div>
                    </div>
                    <div class="attendance-item">
                        <div class="attendance-value">${subject.totalClasses -
subject.classesAttended}</div>
                        <div class="attendance-label">Bunked</div>
                    </div>
                    <div class="attendance-item">
                        <div class="attendance-
value">${subject.totalClasses}</div>
                        <div class="attendance-label">Total</div>
                    </div>
                </div>

                <div class="attendance-progress">
                    <div class="progress-bar" style="width:
${attendancePercentage}%"></div>
                </div>

                <div class="attendance-status">
                    <div class="attendance-
percentage">${attendancePercentage}% Attendance</div>
                    <div class="min-required">Min. Required:
${subject.minAttendance}%</div>
                </div>

                <button class="btn btn-primary bunk-calculator" data-
id="${subject.id}" style="width: 100%; margin-top: 15px;">
                    Bunk Calculator
                </button>
            </div>
        `;

        subjectsDashboard.appendChild(subjectCard);

        const editBtn = subjectCard.querySelector('.edit-subject');
        const deleteBtn = subjectCard.querySelector('.delete-subject');
        const bunkCalcBtn = subjectCard.querySelector('.bunk-calculator');

        editBtn.addEventListener('click', () =>
openEditSubjectModal(subject));
```

```
    deleteBtn.addEventListener('click', () =>
openDeleteSubjectModal(subject));
    bunkCalcBtn.addEventListener('click', () =>
openBunkCalculatorModal(subject));
  }
}

function openEditSubjectModal(subject) {
  document.getElementById('edit-subject-id').value = subject.id;
  document.getElementById('edit-subject-name').value = subject.name;
  document.getElementById('edit-subject-code').value = subject.code;
  document.getElementById('edit-subject-classes').value =
subject.classesAttended;
  document.getElementById('edit-total-classes').value =
subject.totalClasses;
  document.getElementById('edit-min-attendance').value =
subject.minAttendance;

  editSubjectModal.classList.remove('hidden');
}

function openDeleteSubjectModal(subject) {
  document.getElementById('delete-subject-id').value = subject.id;
  confirmDeleteModal.classList.remove('hidden');
}

function openBunkCalculatorModal(subject) {
  const attendancePercentage = subject.totalClasses > 0
    ? Math.round((subject.classesAttended / subject.totalClasses) * 100)
    : 0;

  document.getElementById('calc-subject-name').textContent =
subject.name;
  document.getElementById('calc-current-attendance').textContent =
`${attendancePercentage}%`;
  document.getElementById('calc-min-required').textContent =
`${subject.minAttendance}%`;

  let safeToBunk = calculateSafeToBunk(subject);
  document.getElementById('safe-to-bunk').textContent =
`${safeToBunk} classes`;

  let mustAttend = calculateMustAttend(subject);
  document.getElementById('must-attend').textContent = `${mustAttend}
classes`;

  bunkCalculatorModal.classList.remove('hidden');
}

function calculateSafeToBunk(subject) {
  if (subject.totalClasses === 0) return 0;

  const minRequired = subject.minAttendance / 100;
  const currentAttendance = subject.classesAttended /
subject.totalClasses;

  if (currentAttendance < minRequired) return 0;

  const futurePossibleBunks = Math.floor(
    (subject.classesAttended - minRequired * subject.totalClasses) /
minRequired
  );

  return Math.max(0, futurePossibleBunks);
}

function calculateMustAttend(subject) {
  if (subject.totalClasses === 0) return 0;

  const minRequired = subject.minAttendance / 100;
  const currentAttendance = subject.classesAttended /
subject.totalClasses;

  if (currentAttendance >= minRequired) return 0;

  const mustAttend = Math.ceil(
    (minRequired * subject.totalClasses - subject.classesAttended) / (1 -
minRequired)
  );

  return Math.max(0, mustAttend);
}

function updateSubjectAttendance(subjectId) {
  const subjectIndex = subjects.findIndex(s => s.id === subjectId);
  if (subjectIndex === -1) return;
  const subjectAttendance = attendance.filter(a => a.subjectId ===
subjectId);
  const attended = subjectAttendance.filter(a => a.status ===
'attended').length;
  const missed = subjectAttendance.filter(a => a.status ===
'missed').length;

  subjects[subjectIndex].classesAttended = attended;
  subjects[subjectIndex].totalClasses = attended + missed;

  saveSubjects();
}

function updateStats() {
  const totalSubjects = subjects.length;

  let totalAttended = 0;
  let totalClasses = 0;

  subjects.forEach(subject => {
    totalAttended += subject.classesAttended;
    totalClasses += subject.totalClasses;
  });

  const totalBunked = totalClasses - totalAttended;
  const averageAttendance = totalClasses > 0
    ? Math.round((totalAttended / totalClasses) * 100)
    : 0;

  totalSubjectsElement.textContent = totalSubjects;
  totalAttendedElement.textContent = totalAttended;
  totalBunkedElement.textContent = totalBunked;
  averageAttendanceElement.textContent = `${averageAttendance}%`;
}

function saveSubjects() {
  localStorage.setItem(`bunkifySubjects_${currentUser.id}`,
JSON.stringify(subjects));
}

function saveAttendance() {
  localStorage.setItem(`bunkifyAttendance_${currentUser.id}`,
JSON.stringify(attendance));
}

function loadUserData() {
  const storedSubjects =
localStorage.getItem(`bunkifySubjects_${currentUser.id}`);
  subjects = storedSubjects ? JSON.parse(storedSubjects) : [];

  const storedAttendance =
localStorage.getItem(`bunkifyAttendance_${currentUser.id}`);
  attendance = storedAttendance ? JSON.parse(storedAttendance) : [];

  renderSubjects();
  renderCalendar();
  updateStats();
}

function initUserData() {
  subjects = [];
  attendance = [];
  saveSubjects();
  saveAttendance();
  renderSubjects();
  renderCalendar();
  updateStats();
}

function showApp() {
  authPage.classList.add('hidden');
  appPage.classList.remove('hidden');
}

function showAuth() {
  authPage.classList.remove('hidden');
  appPage.classList.add('hidden');
}

function checkAuth() {
  const storedUser = localStorage.getItem('bunkifyCurrentUser');
  if (storedUser) {
    currentUser = JSON.parse(storedUser);
    showApp();
```

```
        loadUserData();
    } else {
        showAuth();
    }
}

function init() {
    initTheme();
    checkAuth();
    renderCalendar();
}

window.addEventListener('DOMContentLoaded', init);
```

OUTPUT:



Today is: Sunday, April 20, 2025

Create Your Bunkify Account

Full Name

John Doe

Email

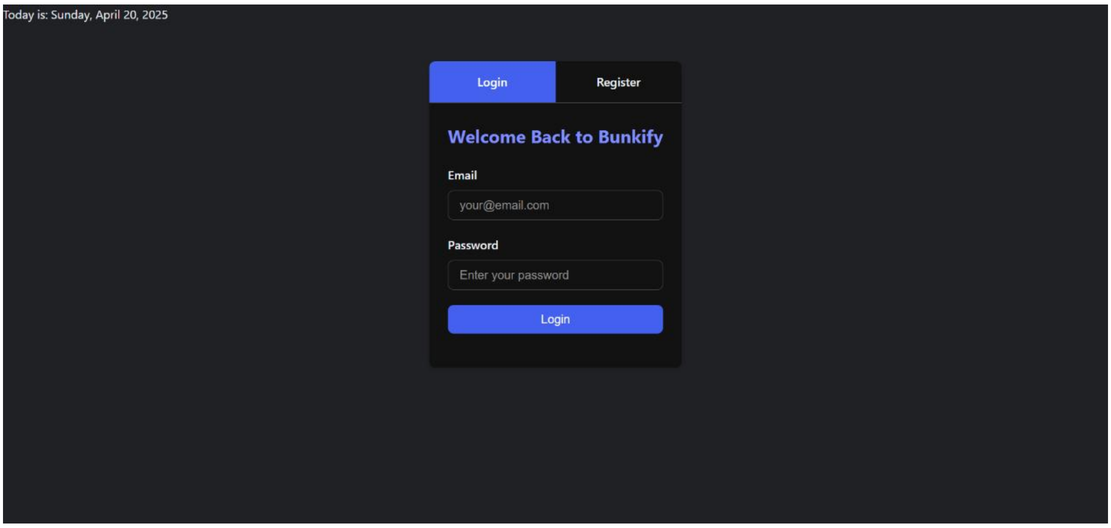
your@email.com

Password

Create a password

Confirm Password

Confirm your password



Today is: Sunday, April 20, 2025

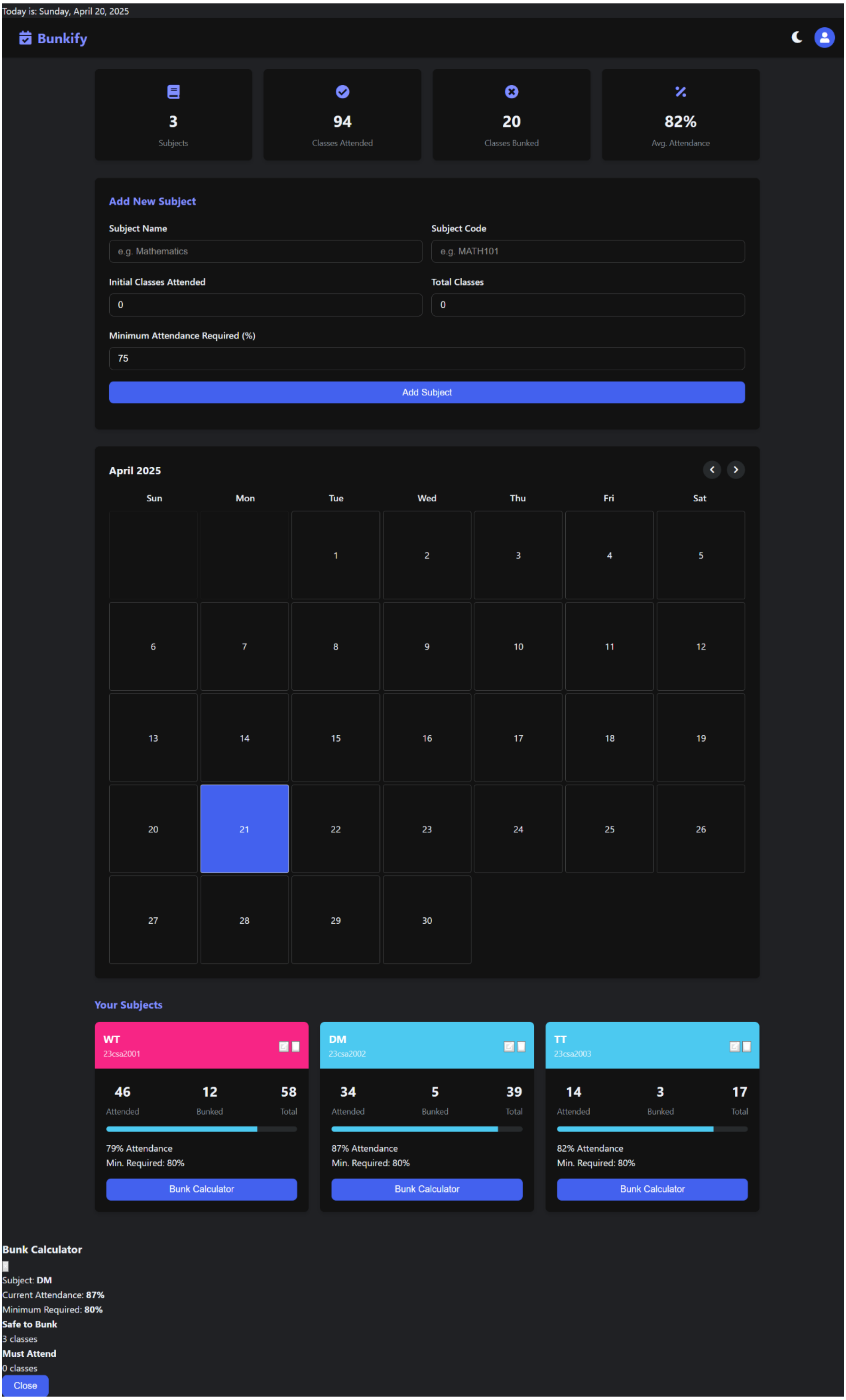
Welcome Back to Bunkify

Email

your@email.com

Password

Enter your password



Today is: Sunday, April 20, 2025

Bunkify

3
Subjects

94
Classes Attended

20
Classes Bunked

82%
Avg. Attendance

Add New Subject

Subject Name: e.g. Mathematics

Subject Code: e.g. MATH101

Initial Classes Attended: 0

Total Classes: 0

Minimum Attendance Required (%): 75

April 2025

Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Your Subjects

WT
23csa2001

46 Attended · 12 Bunked · 58 Total

79% Attendance
Min. Required: 80%

DM
23csa2002

34 Attended · 5 Bunked · 39 Total

87% Attendance
Min. Required: 80%

TT
23csa2003

14 Attended · 3 Bunked · 17 Total

82% Attendance
Min. Required: 80%

Bunk Calculator

Subject: **DM**
Current Attendance: **87%**
Minimum Required: **80%**
Safe to Bunk
3 classes
Must Attend
0 classes