

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки

Кваліфікаційна робота

Комп'ютерна система зберігання та обробки файлів

Виконала:

студентка гр. КІУКІ-21-4

Провоторова А.А.

Керівник:

доц. каф. ЕОМ

Сорокін А.Р.

Мета та завдання роботи

2

Метою кваліфікаційної роботи є розробка комп'ютерної системи для зберігання та обробки файлів статистичних даних, що стосуються управління особистим часом, із використанням технології SaaS.

Завдання включають:

- ❖ розробку архітектури інформаційної системи з підтримкою SaaS-моделі;
- ❖ проектування та реалізацію реляційної бази даних для зберігання статистичних відомостей;
- ❖ створення клієнтської частини програми на основі PHP для забезпечення інтерактивної взаємодії з користувачем;
- ❖ реалізацію функціоналу, що дозволяє вносити, аналізувати та візуалізувати дані про використання часу;
- ❖ забезпечення зручного доступу до сервісу для кінцевих користувачів через веб-інтерфейс.

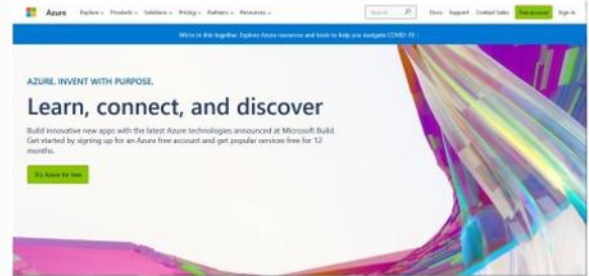
Технологія SaaS

3



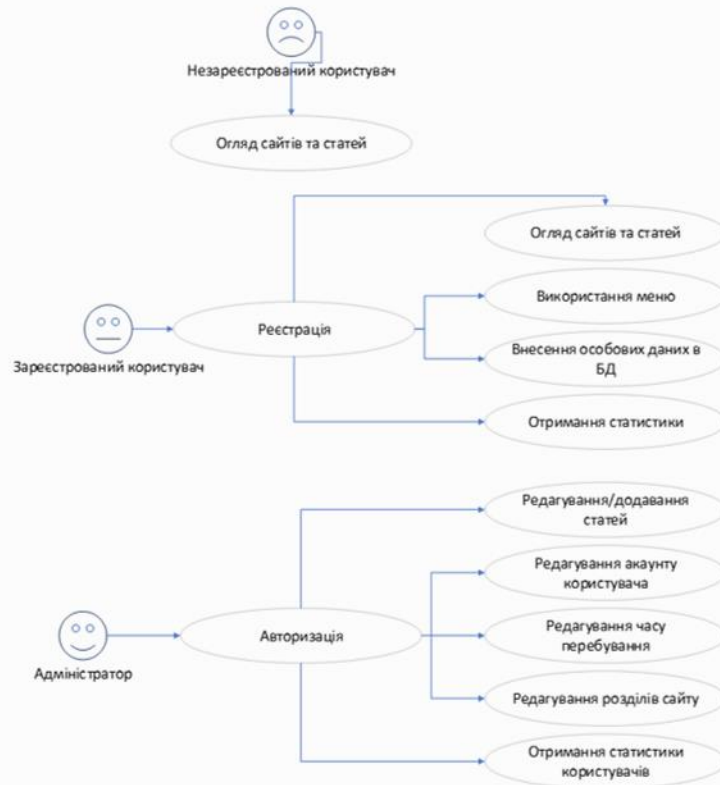
Існуючі SaaS - платформи

4



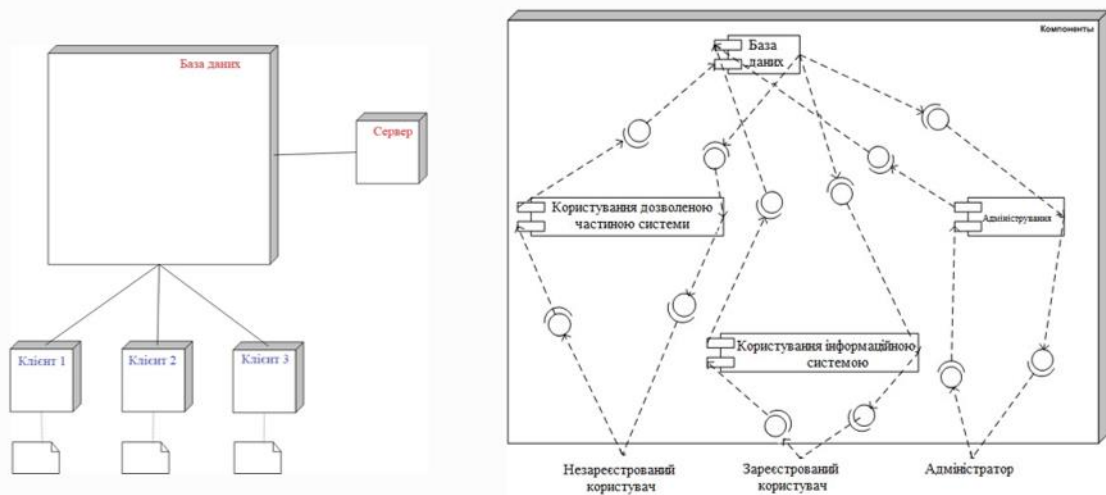
Діаграма використання предметної області

5

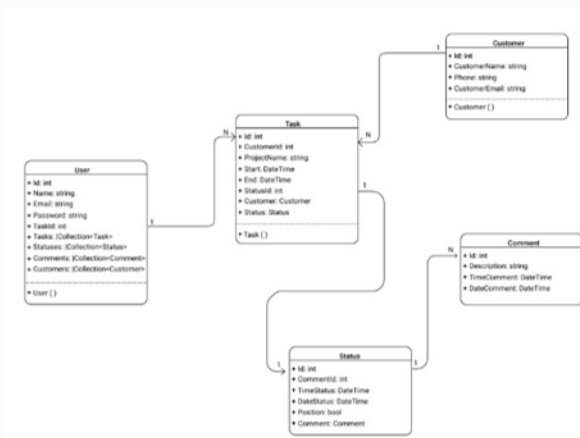


Діаграма розгортання

6



Структурна схема класів та архітектура ПЗ



Структурна послідовність

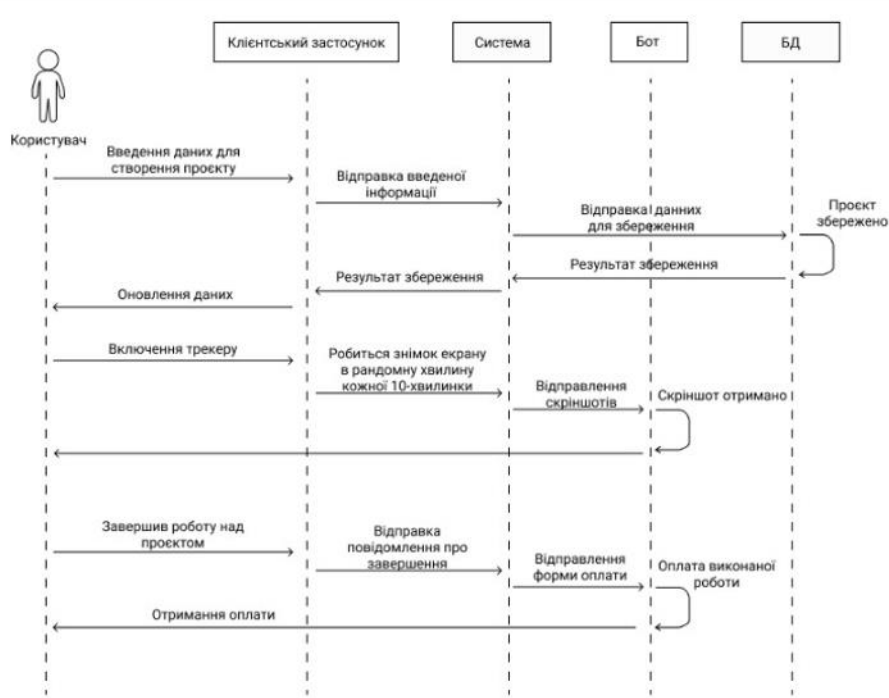


Схема структурних компонентів

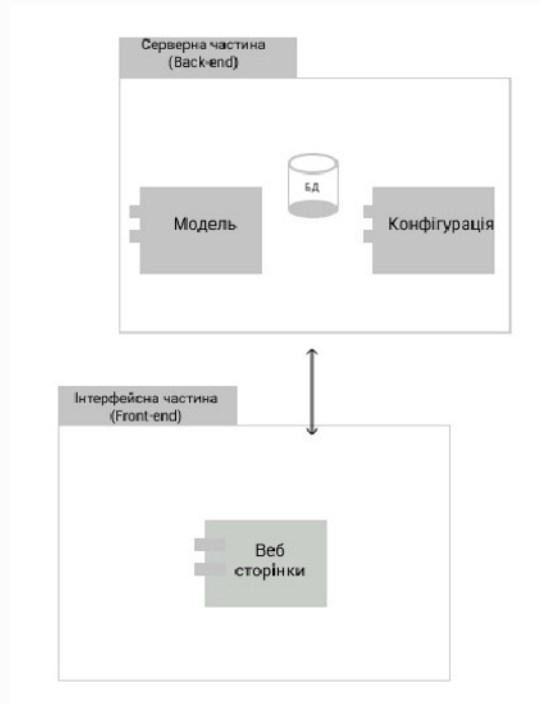
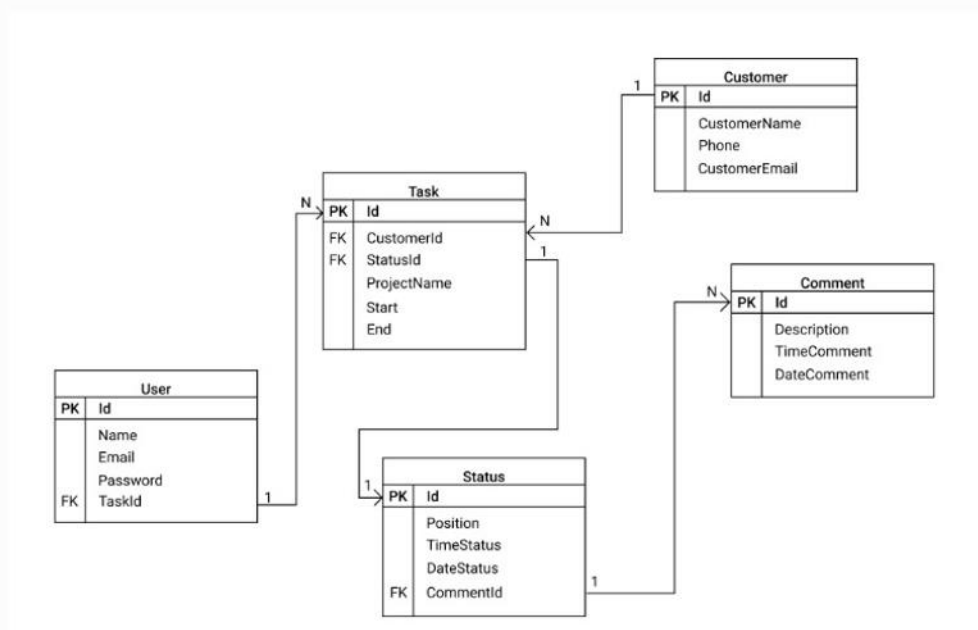


Схема БД



Результати роботи

11

The screenshot displays the 'TM System' interface. On the left, a navigation menu includes 'Панель управління', 'Активності', 'Файли', and 'Аналітика'. A user profile dropdown menu is open, showing 'Користувач', 'Гість', 'Користувач', and 'Адміністратор'. The main content area is divided into several sections:

- Завантаження файлів:** A button labeled 'Вибрати файл' and a note 'Підприємств всі типи файлів'.
- Мої файли (2):** A list of files:
 - zml_36_тождень.pdf (240 KB, Завантажено 2025-06-23)
 - планування_касу.бкс (50 KB, Завантажено 2025-06-22)
- Аналіза використання часу:** A pie chart showing time distribution across activities:
 - Робота: 8г
 - Навчання: 2г
 - Відпочинок: 2г
 - Пробіжка: 2г
 - Підготовка до заисту: 7г
 - Щось масмо додати :): 1г
- Додати нову активність:** A form with fields for 'Щось масмо додати :)', a date/time selector (showing 11:00 AM and 12:00 PM), and a '+ Додати' button.
- Список активностей:** A table with columns: НАЗВА, ДАТА, ПОЧАТОК, КІНЕЦЬ, ТРИВАЛІСТЬ.

НАЗВА	ДАТА	ПОЧАТОК	КІНЕЦЬ	ТРИВАЛІСТЬ
Робота	2025-06-25	09:00	17:00	8г
Навчання	2025-06-25	19:00	21:00	2г
Відпочинок	2025-06-25	21:00	23:00	2г
Пробіжка	2025-01-07	09:00	11:00	2г
Підготовка до заисту	2025-02-06	11:00	18:00	7г
- Панель адміністратора:** A sidebar menu with 'Панель управління', 'Активності', 'Аналітика', and 'Адміністрування' (selected). Below it, a 'Панель адміністратора' section includes:
 - Управління користувачами: Переглядає та редагує список користувачів.
 - Контроль надзвичайних ситуацій: Підприємств систем та мереж.
 - Системні налаштування: Конфігурація системи.

Висновки

12

Розроблено комп'ютерну систему для зберігання та обробки файлів статистичних даних, що стосуються управління особистим з використанням SaaS технологій. Розроблена система демонструє високий рівень функціональної інтеграції, об'єднуючи модулі управління активностями, аналітичної обробки даних, управління файлами та адміністративного керування в єдину цілісну платформу. Реалізована архітектура забезпечує масштабованість системи та можливість подальшого розширення функціональності без суттєвих змін у базовій структурі програмного коду. Аналітичний компонент системи реалізує комплексну візуалізацію даних через динамічне генерування кругових діаграм та статистичних звітів, що дозволяє користувачам отримувати наочне представлення про структуру власної активності та приймати обґрунтовані рішення щодо оптимізації розподілу часу. Алгоритми обчислення ключових показників ефективності забезпечують точність аналітичних розрахунків та релевантність генерованих рекомендацій. Модуль управління файлами розширює функціональні можливості системи, надаючи користувачам інструменти для централізованого зберігання та організації документів, пов'язаних із процесами планування та контролю часу. Інтеграція файлового менеджера з основною аналітичною системою створює синергетичний ефект, підвищуючи загальну ефективність робочих процесів користувачів. Система безпеки та розмежування доступу реалізована через багаторівневу модель авторизації, що забезпечує захист персональних даних користувачів та адміністративних функцій системи. Диференційований підхід до надання привілеїв користувачам різних ролей гарантує відповідність принципам інформаційної безпеки та мінімізує ризики несанкціонованого доступу до критично важливих компонентів системи.

ДОДАТОК Б

Програмний код

```

<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>TM System - Система управління часом</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header class="header">
    <div class="logo">TM System</div>
    <div class="user-controls">
      <select class="user-select" id="userRole">
        <option value="guest">Гість</option>
        <option value="user">Користувач</option>
        <option value="admin">Адміністратор</option>
      </select>
      <span id="userIcon">👤 </span>
    </div>
  </header>

  <div class="container">
    <nav class="sidebar">
      <button class="nav-item active" data-
section="dashboard">🏠 Панель управління</button>
      <button class="nav-item" data-section="activities"
id="activitiesNav">📅 Активності</button>
      <button class="nav-item" data-section="analytics"
id="analyticsNav">📊 Аналітика</button>
      <button class="nav-item hidden" data-section="admin"
id="adminNav">⚙️ Адміністрування</button>
    </nav>

    <main class="main-content">
      <!-- Dashboard Section -->
      <section id="dashboard" class="section active">
        <h2>Панель управління</h2>
        <div class="stats-grid">
          <div class="stat-card blue">
            <div>Загальний час</div>
            <div class="stat-value"
id="totalHours">0г</div>
          </div>
          <div class="stat-card green">

```

```

        <div>Активності</div>
        <div class="stat-value"
id="totalActivities">0</div>
    </div>
    <div class="stat-card purple">
        <div>Середня тривалість</div>
        <div class="stat-value"
id="avgDuration">0г</div>
    </div>
</div>
<div id="dashboardChart"></div>
</section>

<!-- Activities Section -->
<section id="activities" class="section">
    <h2>Управління активностями</h2>

    <div class="form-section">
        <h3>Додати нову активність</h3>
        <div class="form-grid">
            <div class="form-group">
                <label class="form-label">Назва
активності</label>
                <input type="text" class="form-
input" id="activityName" placeholder="Що маємо додати :)">
            </div>
            <div class="form-group">
                <label class="form-
label">Дата</label>
                <input type="date" class="form-
input" id="activityDate">
            </div>
            <div class="form-group">
                <label class="form-label">Час
початку</label>
                <input type="time" class="form-
input" id="startTime">
            </div>
            <div class="form-group">
                <label class="form-label">Час
завершення</label>
                <input type="time" class="form-
input" id="endTime">
            </div>
            <div class="form-group">
                <button class="btn btn-primary"
onclick="addActivity()">+ Додати</button>
            </div>
        </div>
    </div>

    <h3>Список активностей</h3>
    <table class="table">

```

```

        <thead>
            <tr>
                <th>Назва</th>
                <th>Дата</th>
                <th>Початок</th>
                <th>Кінець</th>
                <th>Тривалість</th>
            </tr>
        </thead>
        <tbody id="activitiesTable">
            <!-- Activities will be inserted here -->
    >
            </tbody>
        </table>
    </section>

    <!-- Analytics Section -->
    <section id="analytics" class="section">
        <h2>Аналіз використання часу</h2>
        <div class="chart-container">
            <svg class="pie-chart" id="analyticsChart"
viewBox="0 0 200 200">
                <!-- Chart will be generated here -->
            </svg>
            <div class="legend" id="chartLegend">
                <!-- Legend will be generated here -->
            </div>
        </div>
    </section>

    <!-- Admin Section -->
    <section id="admin" class="section">
        <h2>Панель адміністратора</h2>
        <div class="admin-grid">
            <div class="admin-card">
                <h3>👤 Управління користувачами</h3>
                <p>Перегляд та редагування облікових
записів</p>
            </div>
            <div class="admin-card">
                <h3>📄 Контент-менеджмент</h3>
                <p>Редагування статей та новин</p>
            </div>
            <div class="admin-card">
                <h3>⚙️ Системні налаштування</h3>
                <p>Конфігурація системи</p>
            </div>
        </div>
    </section>
</main>
</div>

```

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

---

/* styles.css */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI',
  Roboto, sans-serif;
  background-color: #f5f5f5;
  line-height: 1.6;
}

.header {
  background: white;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  padding: 1rem 2rem;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.logo {
  display: flex;
  align-items: center;
  font-size: 1.5rem;
  font-weight: bold;
  color: #2563eb;
}

.logo::before {
  content: "🔴";
  margin-right: 0.5rem;
}

.user-controls {
  display: flex;
  align-items: center;
  gap: 1rem;
}

.user-select {
  padding: 0.5rem;
  border: 1px solid #d1d5db;
  border-radius: 0.5rem;
  background: white;
}
```

```
}

.container {
  max-width: 1200px;
  margin: 2rem auto;
  padding: 0 2rem;
  display: grid;
  grid-template-columns: 250px 1fr;
  gap: 2rem;
}

.sidebar {
  background: white;
  border-radius: 0.5rem;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  padding: 1rem;
  height: fit-content;
}

.nav-item {
  display: block;
  width: 100%;
  padding: 0.75rem 1rem;
  margin-bottom: 0.5rem;
  border: none;
  background: none;
  text-align: left;
  border-radius: 0.5rem;
  cursor: pointer;
  transition: all 0.2s;
  color: #374151;
  text-decoration: none;
}

.nav-item:hover {
  background-color: #f3f4f6;
}

.nav-item.active {
  background-color: #dbeafe;
  color: #1d4ed8;
}

.main-content {
  background: white;
  border-radius: 0.5rem;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  padding: 2rem;
}

.section {
  display: none;
}
```

```
.section.active {
  display: block;
}

.stats-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 1rem;
  margin-bottom: 2rem;
}

.stat-card {
  padding: 1.5rem;
  border-radius: 0.5rem;
  background: linear-gradient(135deg, #667eea 0%, #764ba2
100%);
  color: white;
}

.stat-card.blue {
  background: linear-gradient(135deg, #3b82f6 0%, #1d4ed8
100%);
}

.stat-card.green {
  background: linear-gradient(135deg, #10b981 0%, #059669
100%);
}

.stat-card.purple {
  background: linear-gradient(135deg, #8b5cf6 0%, #7c3aed
100%);
}

.stat-value {
  font-size: 2rem;
  font-weight: bold;
  margin-top: 0.5rem;
}

.form-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
  gap: 1rem;
  margin-bottom: 1rem;
}

.form-group {
  display: flex;
  flex-direction: column;
}
```

```
.form-label {
  margin-bottom: 0.5rem;
  font-weight: 500;
  color: #374151;
}

.form-input {
  padding: 0.75rem;
  border: 1px solid #d1d5db;
  border-radius: 0.5rem;
  background: white;
}

.btn {
  padding: 0.75rem 1.5rem;
  border: none;
  border-radius: 0.5rem;
  cursor: pointer;
  font-weight: 500;
  transition: all 0.2s;
}

.btn-primary {
  background-color: #3b82f6;
  color: white;
}

.btn-primary:hover {
  background-color: #2563eb;
}

.table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 2rem;
}

.table th,
.table td {
  padding: 1rem;
  text-align: left;
  border-bottom: 1px solid #e5e7eb;
}

.table th {
  background-color: #f9fafb;
  font-weight: 600;
  color: #374151;
}

.chart-container {
  display: flex;
  align-items: center;
}
```

```
        gap: 2rem;
        margin-top: 2rem;
    }

    .pie-chart {
        width: 300px;
        height: 300px;
    }

    .legend {
        flex: 1;
    }

    .legend-item {
        display: flex;
        align-items: center;
        margin-bottom: 1rem;
        padding: 0.5rem;
        background: #f9fafb;
        border-radius: 0.5rem;
    }

    .legend-color {
        width: 20px;
        height: 20px;
        border-radius: 50%;
        margin-right: 1rem;
    }

    .admin-grid {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
        gap: 1rem;
    }

    .admin-card {
        padding: 1.5rem;
        border: 1px solid #e5e7eb;
        border-radius: 0.5rem;
        background: white;
    }

    .admin-card h3 {
        margin-bottom: 0.5rem;
        color: #1f2937;
    }

    .admin-card p {
        color: #6b7280;
        font-size: 0.9rem;
    }

    .hidden {
```

```

        display: none !important;
    }

    @media (max-width: 768px) {
        .container {
            grid-template-columns: 1fr;
        }

        .form-grid {
            grid-template-columns: 1fr;
        }

        .chart-container {
            flex-direction: column;
        }
    }
}

---

// script.js
// Application state
let currentUser = 'guest';
let activities = [
    { id: 1, name: 'Робота', date: '2025-06-25', startTime:
'09:00', endTime: '17:00', duration: 480 },
    { id: 2, name: 'Навчання', date: '2025-06-25', startTime:
'19:00', endTime: '21:00', duration: 120 },
    { id: 3, name: 'Відпочинок', date: '2025-06-25', startTime:
'21:00', endTime: '23:00', duration: 120 },
    { id: 4, name: 'Пробіжка', date: '2025-01-07', startTime:
'09:00', endTime: '11:00', duration: 120 },
    { id: 5, name: 'Підготовка до заліку', date: '2025-02-06',
startTime: '11:00', endTime: '18:00', duration: 420 }
];

// User role management
document.getElementById('userRole').addEventListener('change',
function(e) {
    currentUser = e.target.value;
    updateUserInterface();
});

function updateUserInterface() {
    const activitiesNav =
document.getElementById('activitiesNav');
    const analyticsNav =
document.getElementById('analyticsNav');
    const adminNav = document.getElementById('adminNav');

    if (currentUser === 'guest') {
        activitiesNav.classList.add('hidden');
        analyticsNav.classList.add('hidden');
        adminNav.classList.add('hidden');
    }
}

```

```

    } else if (currentUser === 'user') {
        activitiesNav.classList.remove('hidden');
        analyticsNav.classList.remove('hidden');
        adminNav.classList.add('hidden');
    } else if (currentUser === 'admin') {
        activitiesNav.classList.remove('hidden');
        analyticsNav.classList.remove('hidden');
        adminNav.classList.remove('hidden');
    }
}

// Navigation
document.querySelectorAll('.nav-item').forEach(item => {
    item.addEventListener('click', function() {
        const targetSection = this.dataset.section;

        // Update active nav item
        document.querySelectorAll('.nav-item').forEach(nav =>
nav.classList.remove('active'));
        this.classList.add('active');

        // Show target section
        document.querySelectorAll('.section').forEach(section =>
section.classList.remove('active'));

document.getElementById(targetSection).classList.add('active');

        // Update content based on section
        if (targetSection === 'dashboard') {
            updateDashboard();
        } else if (targetSection === 'activities') {
            updateActivitiesTable();
        } else if (targetSection === 'analytics') {
            updateAnalyticsChart();
        }
    });
});

// Activity management
function addActivity() {
    const name = document.getElementById('activityName').value;
    const date = document.getElementById('activityDate').value;
    const startTime =
document.getElementById('startTime').value;
    const endTime = document.getElementById('endTime').value;

    if (!name || !date || !startTime || !endTime) {
        alert('Будь ласка, заповніть всі поля');
        return;
    }

    const start = new Date(`${date}T${startTime}`);
    const end = new Date(`${date}T${endTime}`);

```

```

    const duration = (end - start) / (1000 * 60); // minutes

    if (duration <= 0) {
        alert('Час завершення повинен бути пізніше часу
початку');
        return;
    }

    const newActivity = {
        id: Date.now(),
        name,
        date,
        startTime,
        endTime,
        duration
    };

    activities.push(newActivity);

    // Clear form
    document.getElementById('activityName').value = '';
    document.getElementById('activityDate').value = '';
    document.getElementById('startTime').value = '';
    document.getElementById('endTime').value = '';

    updateActivitiesTable();
    updateDashboard();
}

function updateActivitiesTable() {
    const tbody = document.getElementById('activitiesTable');
    tbody.innerHTML = '';

    activities.forEach(activity => {
        const row = document.createElement('tr');
        row.innerHTML = `
            <td>${activity.name}</td>
            <td>${activity.date}</td>
            <td>${activity.startTime}</td>
            <td>${activity.endTime}</td>
            <td>${Math.round(activity.duration / 60 * 10) /
10}</td>
        `;
        tbody.appendChild(row);
    });
}

// Dashboard updates
function updateDashboard() {
    const totalMinutes = activities.reduce((sum, activity) =>
sum + activity.duration, 0);
    const totalHours = Math.round(totalMinutes / 60 * 10) / 10;
    const avgDuration = activities.length > 0 ?

```

```

Math.round((totalMinutes / activities.length) / 60 * 10) / 10 :
0;

    document.getElementById('totalHours').textContent =
totalHours + 'r';
    document.getElementById('totalActivities').textContent =
activities.length;
    document.getElementById('avgDuration').textContent =
avgDuration + 'r';

    // Create dashboard chart
    createChart('dashboardChart');
}

// Analytics chart
function updateAnalyticsChart() {
    createChart('analyticsChart');
}

function createChart(containerId) {
    const container = document.getElementById(containerId);
    if (!container) return;

    // Group activities by name
    const activitySummary = activities.reduce((acc, activity) =>
{
    acc[activity.name] = (acc[activity.name] || 0) +
activity.duration;
    return acc;
}, {});

    const totalTime =
Object.values(activitySummary).reduce((sum, time) => sum + time,
0);
    const colors = ['#3B82F6', '#10B981', '#F59E0B', '#EF4444',
'#8B5CF6', '#6366F1'];

    // Clear existing chart
    if (container.tagName === 'SVG') {
        container.innerHTML = '';
    } else {
        container.innerHTML = '<svg class="pie-chart" viewBox="0
0 200 200"></svg>';
        const svg = container.querySelector('svg');
        container = svg;
    }

    let startAngle = 0;
    let colorIndex = 0;

    Object.entries(activitySummary).forEach(([name, time]) => {
        const percentage = time / totalTime;
        const angle = percentage * 360;

```

```

        const x1 = 100 + 80 * Math.cos((startAngle - 90) *
Math.PI / 180);
        const y1 = 100 + 80 * Math.sin((startAngle - 90) *
Math.PI / 180);
        const x2 = 100 + 80 * Math.cos((startAngle + angle - 90)
* Math.PI / 180);
        const y2 = 100 + 80 * Math.sin((startAngle + angle - 90)
* Math.PI / 180);

        const largeArcFlag = angle > 180 ? 1 : 0;

        const path =
document.createElementNS('http://www.w3.org/2000/svg', 'path');
        path.setAttribute('d', `M 100 100 L ${x1} ${y1} A 80 80
0 ${largeArcFlag} 1 ${x2} ${y2} Z`);
        path.setAttribute('fill', colors[colorIndex %
colors.length]);
        path.setAttribute('stroke', 'white');
        path.setAttribute('stroke-width', '2');

        container.appendChild(path);

        startAngle += angle;
        colorIndex++;
    });

    // Update legend
    updateChartLegend(activitySummary, colors);
}

function updateChartLegend(activitySummary, colors) {
    const legend = document.getElementById('chartLegend');
    if (!legend) return;

    legend.innerHTML = '';
    let colorIndex = 0;

    Object.entries(activitySummary).forEach(([name, time]) => {
        const hours = Math.round(time / 60 * 10) / 10;
        const legendItem = document.createElement('div');
        legendItem.className = 'legend-item';
        legendItem.innerHTML = `
        <div class="legend-color" style="background-color:
${colors[colorIndex % colors.length]}"></div>
        <div style="flex: 1; display: flex; justify-content:
space-between;">
            <span>${name}</span>
            <span>${hours}r</span>
        </div>
        `;
        legend.appendChild(legendItem);
        colorIndex++;
    });
}

```

```
    });  
}  
  
// Initialize the application  
function init() {  
    updateUserInterface();  
    updateDashboard();  
    updateActivitiesTable();  
  
    // Set today's date as default  
    const today = new Date().toISOString().split('T')[0];  
    document.getElementById('activityDate').value = today;  
}  
  
// Start the application  
init();
```