**Чернівецький національний університет імені Юрія Федьковича**

**Факультет математики та інформатики**

**Кафедра прикладної математики та інформаційних технологій**

**КУРСОВА РОБОТА**

на тему: **Система трекінгу робочих завдань в ІТ компанії**

Студент IV курсу 402 групи

галузь знань 0403 – системні науки та кібернетика.

напрям підготовки 6.040301–Прикладна математика

Собіль Роман Богданович

(прізвище, ім’я, по-батькові)

Керівник: к.ф. − м.н., асистент Краснокутська Інесса Володимирівна

.

.

Чернівці–2018

|  |  |
| --- | --- |
|  |  |

**АНОТАЦІЯ**

У курсовій роботі розглядається розробка системи тайм-трекінгу працівників ТоВ "Юкон Софтваре". Спроектовано та імплементовано інтерфейс створення груп працівників, додавання до цих груп користувачів і менеджерів з бази даних. Дана система написана за допомогою мов програмування PHP, HTML5, MySQL, а також за допомогою CSS та Smarty.

**Зміст**

[ВСТУП 4](#_Toc510725805)

[РОЗДІЛ 1. ЗАСТОСУВАННЯ "ТАЙМ-ТРЕКІНГУ" В IT-СФЕРІ 6](#_Toc510725806)

[1.1. Що таке "time tracker" і для чого він використовується 6](#_Toc510725807)

[1.2. Програмне забезпечення для відстеження часу / запису 6](#_Toc510725808)

[1.3. Можливості системи трекінгу 7](#_Toc510725809)

[РОЗДІЛ 2. ПРОГРАМНІ ЗАСОБИ, ЯКІ ВИКОРИСТАНІ ДЛЯ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКУ 8](#_Toc510725810)

[2.1. Проектування бази даних на SQL 8](#_Toc510725811)

[2.2. Мова програмування PHP 10](#_Toc510725812)

[2.3. Дизайн веб-додатку за допомогою HTML, CSS, Smarty, Javascript 13](#_Toc510725813)

[РОЗДІЛ 3. ОПИС ПРОЕКТУ 20](#_Toc510725814)

[3.1. Основна ідея проекту 20](#_Toc510725815)

[3.2. Робота з базою даних 21](#_Toc510725816)

[3.3. Опис функціоналу веб-додатку 25](#_Toc510725817)

[ВИСНОВОК 30](#_Toc510725818)

[СПИСОК ПОСИЛАНЬ 31](#_Toc510725819)

[ДОДАТОК (КОД ПРОГРАМИ) 32](#_Toc510725820)

ВСТУП

В IT-сфері працюють багато інженерів програмного забезпечення і менша кількість керівників. У керівників постає задача слідкувати за роботою програмістів. В таких випадках в нагоді стають певні додатки (програми), які допоможуть відстежувати роботу програмістів і полегшити роботу керівників та менеджерів.

**Мета курсової роботи** – створення веб-застосунку для моніторингу роботи програмістів, а також наповнення її інформацією про працівників підприємства, згрупованою за технологіями програмування.

**Об’єкт дослідження:** програмне забезпечення в IT-сфері "Юкон-Софтваре".

**Предмет дослідження**: веб-застосунок для контролю роботи працівників ІТ-компанії.

**Основні етапи розробки програмного продукту:**

1. Підключення та налагодження бази даних MySQL.
2. Розробка функціоналу веб-застосунку:

* Створення в базі даних нових таблиць.
* Написання за допомогою HTML нової сторінки для сайту.
* За допомогою CSS, Smarty гарно оформити сторінки та вивести інформацію з бази даних.
* Можливість додавання,редагування груп.
* Додавання, видалення користувачів та менеджерів з відповідних груп.
* Система трекінгу повинна сповіщати менеджерів, якщо програміст не повідомив, що він виконав завдання.

При написанні веб-застосунку були використані такі технології:

1. Локальний веб-сервер XAMPP – використовувався у процесі розробки програмного додатку.
2. СКБД MySQL та веб-застосунок PHPMyAdmin – для створення та адміністрування бази даних.
3. Мова програмування PHP.
4. Для дизайну були використані HTML, CSS, Smarty.
5. Для розробки використовувалось середовище Atom.

**Практичне значення дослідження.**

Створений веб-застосунок буде використаний ІТ-компанією ТоВ "Юкон Софтваре". За допомогою системи трекінгу менеджери груп зможуть контролювати кількість потраченого часу програмістом, роботу програміста тощо.

РОЗДІЛ 1. ЗАСТОСУВАННЯ "ТАЙМ-ТРЕКІНГУ" В IT-СФЕРІ

1.1. Що таке "time tracker" і для чого він використовується

Програмне забезпечення для відстеження часу — це категорія комп'ютерного програмного забезпечення, яка дозволяє співробітникам записувати час, що витрачається на виконання завдань або проектів. Такого роду програмне програмне забезпечення використовується в багатьох галузях, включаючи тих, хто працює фрілансерами і погодинними працівниками. Воно також використовується професіоналами, які працюють погодинно. Інструмент може використовуватися автономно або бути інтегрованим з іншими додатками, такими як програмне забезпечення для управління проектами, підтримка клієнтів та бухгалтерський облік, щоб назвати лише деякі з них. Програмне забезпечення для відстеження часу — це електронна версія традиційного паперового розкладу. Час відстеження може збільшити продуктивність, оскільки компанії можуть відстежувати час, що витрачається на виконання завдань, і проаналізувати які причини змушують співробітників витрачати час. Програмне забезпечення для відстеження часу підвищує прозорість за рахунок документування часу, необхідного для завершення завдань. Дані збираються в базі даних і можуть використовуватися для подальшого аналізу відділами керування персоналом.

1.2. Програмне забезпечення для відстеження часу / запису

Програмне забезпечення для відстеження часу автоматизує процес відстеження часу, записуючи дії, що виконуються на комп'ютері, і час, що витрачається на кожну з них. Це програмне забезпечення призначене для покращення керування електронним розкладом. Його мета — запропонувати загальну картину використання комп'ютерів. Автоматичне відстеження часу програмного забезпечення записує і показує використання додатків, документів, ігор, веб-сайтів і т.д.

При використанні всередині компаній це програмне забезпечення дозволяє контролювати продуктивність співробітників, записуючи завдання, які вони виконують на своїх комп'ютерах. Його можна використовувати для заповнення розкладів.

American Payroll Association вважає, що компанії можуть заощадити 2% від загальної суми витрат на оплату праці щороку за рахунок автоматизації відстеження часу.

При використанні фрілансерами це програмне забезпечення допомагає створювати звіти для клієнтів (наприклад, розклади і рахунки-фактури) або у якості доказу та звіту про виконану роботу.

1.3. Можливості системи трекінгу

Після того як ми розібрались що таке система трекінгу, можна виділити основні можливості та чим сисетма допомагає керівнику.

Трекер допомагає:

1) оцінювати продуктивність роботи за комп'ютером програміста;

2) контролювати запізнення та інші порушення;

3) виявляти кращого або гіршого співробітника;

4) контролювати співробітників з гнучким графіком роботи;

5) вести облік часу нарад і перерв;

6) оцінити реальне навантаження персоналу;

7) вести віддалений контроль роботи персоналу;

8) розрахувати заробітну плату співробітників з погодинною оплатою праці;

9) оперативно розсилати повідомлення з допомогою email, якщо програміст не заповнив інформацію про виконану роботу;

РОЗДІЛ 2. ПРОГРАМНІ ЗАСОБИ, ЯКІ ВИКОРИСТАНІ ДЛЯ СТВОРЕННЯ ВЕБ-ЗАСТОСУНКУ

2.1. Проектування бази даних на SQL

Для підключення, адміністрування, редагування бази даних було використано СКБД MySQL та веб-застосунок phpMyAdmin.

MySQL — вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL  – компактний [багатопотоковий](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%96%D1%81%D1%82%D1%8C) [сервер баз даних](https://uk.wikipedia.org/w/index.php?title=%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85&action=edit&redlink=1). Характеризується високою швидкістю, стійкістю і простотою використання. MySQL вважається гарним рішенням для малих і середніх застосувань.  Найповніше можливості сервера виявляються в [UNIX](https://uk.wikipedia.org/wiki/UNIX)-системах, де є підтримка [багатопоточності](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2%D1%96%D1%81%D1%82%D1%8C), що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

* простота у встановленні та використанні;
* підтримка необмеженої кількості користувачів, що одночасно працюють із БД;
* кількість рядків у таблицях може досягати 50 млн;
* висока швидкість виконання команд;
* наявність простої і ефективної системи безпеки.

phpMyAdmin — це безкоштовний програмний інструмент, написаний на PHP, призначений для управління адмініструванням MySQL через Інтернет. phpMyAdmin підтримує широкий спектр операцій над MySQL і MariaDB. Часто використовувані операції (управління базами даних, таблиці, стовпці, відносини, індекси, користувачі, дозволи і т. д.) можуть виконуватися через призначений для користувача інтерфейс, в той час як у вас все ще є можливість безпосередньо виконувати будь-яку інструкцію SQL.

Щоб полегшити використання для широкого кола людей, phpMyAdmin перекладається на 72 мови і підтримує мови LTR і RTL.

phpMyAdmin — це зрілий проект зі стабільною і гнучкою базою коду.

Проект phpMyAdmin є членом програми Software Freedom Conservancy. SFC — це некомерційна організація, яка допомагає просувати, покращувати, розробляти і захищати проекти Free, Libre і Open Source Software (FLOSS).

SQL ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Structured query language – мова структурованих запитів) –[декларативна мова програмування](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) для взаємодії користувача з [базами даних](https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85), що застосовується для формування запитів, оновлення і керування [реляційними БД](https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85), створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування ([C](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_C) або С++), SQL може формувати інтерактивні запити або, бувши вбудованою в прикладні програми, виступати в якості інструкцій керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

SQL – це діалогова мова програмування для здійснення [запиту](https://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D0%B7%D0%B0%D0%BF%D0%B8%D1%82) і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і вилучення даних за допомогою використання системи керування і адміністративних функцій.

2.2. Мова програмування PHP

PHP (Hypertext Preprocessor, Препроцесор гіпертексту) — це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом.

Синтаксис мови бере початок з C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання web-розробникам можливості швидкого створення динамічних web-сторінок.

Навідміну від мови програмування JavaScript, який є клієнтським кодом, PHP-скрипти виконуються на стороні серверу. Навіть є можливість конфігурувати свій сервер таким чином, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта.

Можна виділити три основні області, де використовується PHP:

* **Створення скриптів для виконання на стороні сервера**. PHP найбільш широко використовується саме таким чином. Все, що знадобиться, це парсер PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Для перегляду результатів виконання PHP-скриптів в браузері, потрібен працюючий вебсервер і встановлений PHP.
* **Створення скриптів для виконання в командному рядку**. Можна створити PHP-скрипт, здатний запускатися незалежно від веб-сервера та браузера. Все, що буде потрібно — парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою утиліти планування задач cron (на платформах Unix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформі Windows. Ці скрипти також можуть бути використані в задачах простої обробки текстів.
* **Створення додатків GUI (графічних інтерфейсів), що виконуються на стороні клієнта**. Можливо, PHP є не найкращим для створення подібних додатків, але, якщо дуже добре знати PHP і мати бажання використовувати деякі його можливості у своїх клієнт-додатках, можна використовувати PHP-GTK для створення таких додатків.

PHP доступний для більшості операційних систем, включаючи Linux, багато модифікацій Unix (такі, як HP-UX, Solaris і OpenBSD), Microsoft Windows, Mac OS X, RISC OS, та багатьох інших. Існує навіть версія PHP для OS/2. Про це мова йде в [3].

Також в PHP включена підтримка більшості сучасних веб-серверів, таких, як Apache, Microsoft Internet Information Server (IIS), Personal Web Server, серверів Netscape і iPlanet, Oreilly Website Pro, Caudium, Xitami, OmniHTTPd та багатьох інших. Для більшості серверів PHP поставляється в якості модуля, для інших, що підтримують стандарт CGI, PHP може функціонувати як процесор CGI.

Таким чином PHP надає свободу вибору операційної системи і веб-сервера. Крім того з"являється можливість обрати між використанням процедурного або об'єктно-орієнтованого програмування або поєднати їх.

В даний час PHP підтримує наступні бази даних:

Adabas D, Ingres, Oracle (OCI7 і OCI8), dBase, InterBase, Ovrimos, Empress, FrontBase, PostgreSQL, FilePro (тільки читання) , mSQL, Solid, Hyperwave, Direct MS-SQL, Sybase, IBM DB2, MySQL, Velocis, Informix, ODBC, Unix, dbm

PHP також підтримує "спілкування" з іншими сервісами з використанням таких протоколів як LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) і багатьох інших.

PHP містить ряд готових бібліотек для роботи із популярними базами даних.

**Засоби безпеки мови PHP**:

1. Засоби безпеки системного рівня. РНР можна налаштувати такщоб була забезпечена максимальною свободою дій і безпекою. РНР може працювати в безпечному режимі (safe mode), який обмежує можливості застосування РНР користувачами.

2. Засоби безпеки рівня програми. РНР включає надійні механізми шифрування. РНР також сумісний з багатьма додатками інших розробників, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції. Вихідний код РНР не можна переглянути у браузері, оскільки він виконується на сервері.

РНР використовується не лише у поєднанні з HTML, але й із JavaScript, WML, XML та іншими мовами програмування. РНР-код може передаватися будь-яким браузерам і пристроям, в тому числі стільниковим телефонам, портативним комп’ютерам. РНР-код можна виконувати в режимі командного рядка.

РНР працює на різних web-серверах (Apache, Netscape Enterprise Server, Microsoft IIS, Stronghold, Zeus) і платформах (UNIX, Solaris, FreeBSD, Windows 95/98/NT/2000/XP/2003/VISTA/7/8/8.1/10).

2.3. Дизайн веб-додатку з використанням HTML, CSS, Smarty, Javascript

**HTML** (HyperText Markup Language, мова розмітки  [гіпертекстових](https://uk.wikipedia.org/wiki/%D0%93%D1%96%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) документів) – стандартна [мова розмітки](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D1%80%D0%BE%D0%B7%D0%BC%D1%96%D1%82%D0%BA%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0) в [Інтернеті](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82). Більшість [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0) створюються за допомогою мови HTML (або [XHTML](https://uk.wikipedia.org/wiki/XHTML)). Документ HTML обробляється [браузером](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) та відтворюється на екрані у звичному для людини вигляді.

Незважаючи на те, що HTML – штучна комп'ютерна мова, вона не є [мовою програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). HTML разом із [каскадними таблицями стилів](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D1%81%D0%BA%D0%B0%D0%B4%D0%BD%D1%96_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D1%96_%D1%81%D1%82%D0%B8%D0%BB%D1%96%D0%B2) (CSS) та [вбудованими скриптами](https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) — це три основні технології побудови [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0).

HTML впроваджує засоби для:

* створення структурованого документа шляхом позначення структурного складу тексту: заголовків, абзаців, списків, таблиць, цитат та іншого;
* отримання інформації із Всесвітньої мережі через [гіперпосилання](https://uk.wikipedia.org/wiki/%D0%93%D1%96%D0%BF%D0%B5%D1%80%D0%BF%D0%BE%D1%81%D0%B8%D0%BB%D0%B0%D0%BD%D0%BD%D1%8F);
* створення інтерактивних форм;
* включення зображень, звуку, відео, та інших об'єктів до тексту.

Документ HTML 4.01 складається з трьох частин:

* Декларація типу документа (Document type declaration, Doctype) на початку документа, в якій визначається тип документа ([DTD](https://uk.wikipedia.org/wiki/Document_Type_Definition)).
* Шапка документа (знаходиться в межах елемента head), в якій записано загальні технічні відомості або додаткова інформація про документ, яка не відтворюється безпосередньо в [браузері](https://uk.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80).
* Тіло документа (може знаходитися в елементах body або frameset), в якому міститься основна інформація документа.

Веб-форми призначені для обміну даними між користувачем і сервером. До того ж, за допомогою клієнтських скриптів можна отримати доступ до будь-якого елементу форми (наприклад змінити його).

Для вказівки браузеру де починається і закінчується форма, використовується контейнер. Між відкриваючими і закриваючими тегами <FORM> і </ FORM> можна поміщати будь-які необхідні теги HTML. Це дозволяє додати елементи форми в осередку таблиці для їх форматування, а також використовувати зображення. Документ може містити декілька форм, але вони не повинні бути вкладені одна в іншу.

Параметри форми:

1. Стандартні поля для введення інформації.

2. Кнопку відправки даних форми на сервер (SUBMIT).

3. Адреса програми на веб-сервері, яка буде обробляти вміст даних форми.

Перед відправкою даних браузер готує інформацію у вигляді пари «ім'я – значення», де ім'я визначається параметром name тега, допустимого в формі, а значення (параметр value) введено користувачем або встановлено в поле форми за замовчуванням. Після натискання користувачем кнопки SUBMIT, відбувається запуск обробника форми, який отримує і обробляє введену в формі інформацію. В якості обробника форми зазвичай виступає CGI-програма, задана параметром action тега <FORM>. Абревіатурою CGI (Common Gateway Interface, загальний шлюзовий інтерфейс) позначається протокол, за допомогою якого програми взаємодіють з веб-сервером. За допомогою CGI на сервері можна виконувати програми на будь-якій мові програмування і результат їх дії виводити у вигляді веб-сторінки. Найбільш популярні PHP, Perl, С ++.

Атрибут method вказує браузеру, який вид HTTP запиту необхідно використовувати для відправки форми; можливі значення POST і GET.

Атрибут name визначає унікальне ім'я форми. Як правило, ім'я форми використовується для доступу до її елементів через скрипти. Як ім'я використовується набір символів, включаючи числа і букви. Java Script? чутливий до регістру, тому при зверненні до форми на ім'я через скрипти дотримуйтесь той же написання, що і в параметрі name.

У своєму проекті я використовував наступний код, який видно в лістингу 1:

*Лістинг 1*

*<form method="post" action="group\_edit.php?id={$group\_id}">*

*<label for = "name"> Назва групи: </label>*

*<input type="text" name="name" id="name" value="{$group\_info.name}">*

*{$error\_text}*

*<input type="submit" value="SAVE">*

*<br>*

*<br>*

*<input type="button" value="back" onClick='location.href="./groups.php"'>*

*</form>*

**Каскадні таблиці стилів** ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) Cascading Style Sheets або скорочено CSS) — спеціальна [мова](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F), що використовується для опису сторінок, написаних [мовами розмітки даних](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D1%80%D0%BE%D0%B7%D0%BC%D1%96%D1%82%D0%BA%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85).

Найчастіше CSS використовують для візуальної презентації сторінок, написаних [HTML](https://uk.wikipedia.org/wiki/HTML) та [XHTML](https://uk.wikipedia.org/wiki/XHTML), але формат CSS може застосовуватися до інших видів [XML](https://uk.wikipedia.org/wiki/XML)-документів.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0). Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

CSS використовується авторами та відвідувачами [веб-сторінок](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B0), щоб визначити кольори, [шрифти](https://uk.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), [верстку](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D1%80%D1%81%D1%82%D0%BA%D0%B0_%D0%B2%D0%B5%D0%B1-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BA%D0%B8) та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай [HTML](https://uk.wikipedia.org/wiki/HTML), [XML](https://uk.wikipedia.org/wiki/XML) або подібна мова розмітки) від вигляду документу (що описується в CSS).

Стилі складаються зі списку правил. Кожне правило має один або більше селектор ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) selector) та блок визначення ([англ.](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) declaration block). Блок визначення складається із оточеного фігурними дужками списку властивостей.

Властивості в списку оформлюються у вигляді назва властивості, двокрапка (:), значення, крапка з комою (;).

**Smarty** — це компілюючий обробник шаблонів до PHP. Надає інструменти, які дозволяють домогтися відділення прикладної логіки й даних від відображення. Це дуже зручно в ситуаціях, коли програміст і верстальник шаблону працюють окремо.

**Можливості**

Smarty дозволяє:

* Створювати функції та модифікатори користувача.
* Використовувати роздільники тегів шаблону.
* Включати PHP-код прямо в шаблон.
* Користувальницькі функції кешування.
* Використання компонентної архітектури.

Для спрощення створення масштабованих веб-застосунків в Smarty вбудований механізм кешування.

**Механізм дії**

Smarty читає файли шаблонів та створює PHP-код на їх основі. Код створюється один раз і потім лише виконується тому немає необхідності обробляти файл шаблона для кожного запиту та кожний шаблон може користуватися всіма перевагами розширень PHP для кешування байт-коду, таких як eAccelerator або PHP Accelerator. Smarty став, ймовірно, першою шаблонною бібліотекою, яка використала такий механізм. Пізніше з'явилося безліч аналогів, що використовують як smarty-подібний синтаксис шаблонів (Twig або Quicky), так й інші підходи до синтаксису шаблонів.

Компіляція шаблонів дозволяє досягти продуктивності, порівнянної з продуктивністю PHP-шаблонів, написаних вручну[2]. Конструкції if/elseif/else/e передаються обробникові PHP, так що синтаксис виразу {if… } може бути натільки простим або складним, наскільки це потрібно. Можливо необмежене вкладення секцій, умов і т. д.

Приклад коду з проекту системи трекінгу можна побачити в лістингу 2:

*Лістинг 2*

*$smarty->assign("forms",array($sform->getName()=>$sform->toArray()) );*

*$smarty->assign("programers", $programers);*

*$smarty->assign("error\_text", $error\_text);*

*$smarty->assign\_by\_ref("userdet\_string",UserHelper::getUserDetailsString($user,$GLOBALS["I18N"]));*

*$smarty->assign\_by\_ref("title\_page",$i18n->getKey("groups.page\_title"));*

*$smarty->assign\_by\_ref("group\_name",$i18n->getKey("groups.name"));*

*$smarty->assign\_by\_ref("group\_id",$i18n->getKey("groups.id"));*

*$smarty->assign\_by\_ref("edit\_group",$i18n->getKey("groups.th.edit"));*

*$smarty->assign\_by\_ref("delete\_group",$i18n->getKey("groups.th.del"));*

*$smarty->assign\_by\_ref("confirm\_delete\_group",$i18n->getKey("groups.th.confirm.del"));*

*$smarty->assign\_by\_ref("add\_group",$i18n->getKey("groups.add"));*

*$smarty->assign\_by\_ref("group\_field\_name",$i18n->getKey("groups.field.name"));*

*$smarty->assign\_by\_ref("button\_add\_group",$i18n->getKey("groups.button.add"));*

*$smarty->assign\_by\_ref("add\_group\_users",$i18n->getKey("group.users.add"));*

*$smarty->assign\_by\_ref("add\_group\_managers",$i18n->getKey("group.managers.add"));*

*$smarty->assign("content\_page\_name","groups.tpl");*

*$smarty->display(INDEX\_TEMPLATE);*

**JavaScript (JS)** — динамічна, об'єктно-орієнтована прототипна мова програмування. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

* написання сценаріїв веб-сторінок для надання їм інтерактивності;
* створення односторінкових веб-застосунків (ReactJS, AngularJS, Vue.js);
* програмування на стороні сервера (Node.js);
* стаціонарних застосунків (Electron, NW.js);
* мобільних застосунків (React Native, Cordova);
* сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
* всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманний «у спадок» від мови С, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам, — функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою Сі має такі корінні відмінності:

* об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів
* функції як об'єкти першого класу
* обробка винятків
* автоматичне приведення типів
* автоматичне прибирання сміття
* анонімні функції

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, грубо кажучи, не обов'язково є функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто, зрештою, успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї.

В лістингу 3 показаний приклад оголошення і використання класу в JavaScript (клас є одночасно функцією, оскільки функції — це об'єкти першого рівня):

*Лістинг 3*

*function MyClass() {*

*this.myValue1 = 1;*

*this.myValue2 = 2;*

*}*

*var mc = new MyClass();*

*mc.myValue1 = mc.myValue2 \* 2;*

Використання в HTML

При використанні в рамках технології HTML JavaScript код включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер. Код JavaScript вставляється в теги *<script></script>* з обов'язковим по специфікації HTML 4.01 атрибутом type="*text/javascript*", хоча в більшості браузерів мова сценаріїв за умовчанням саме JavaScript.

Скрипт, що виводить модальне вікно з класичним написом «Hello, World!» усередині браузера продемонстровано в лістингу 4:

*Лістинг 4*

*<script type="text/javascript">*

*alert('Hello, World!');*

*</script>*

РОЗДІЛ 3. ОПИС ПРОЕКТУ

3.1. Основна ідея проекту

Роботу виконано на замовлення менеджера ТоВ "Юкон-Софтваре"– Вінтонаіка Івана Борисовича. На фірмі є більше 100 програмістів і порядка 20-ти менеджерів. Для полегшення роботи менеджерів, мені була поставлена задача — написати веб-додаток: систему трекінгу та контролю за часом. За допомогою трекеру менеджери будуть щодня бачити, що саме зробили девелопери, скільки часу потратили на роботу тощо. Також моїм завданням було вдосконалити внутрішній сайт фірми, дописавши окрім трекера ще декілька веб сторінок, в яких буде можливість створювати групи з мовами програмування та записувати чи видаляти програмістів та менеджерів з відповідної групи.

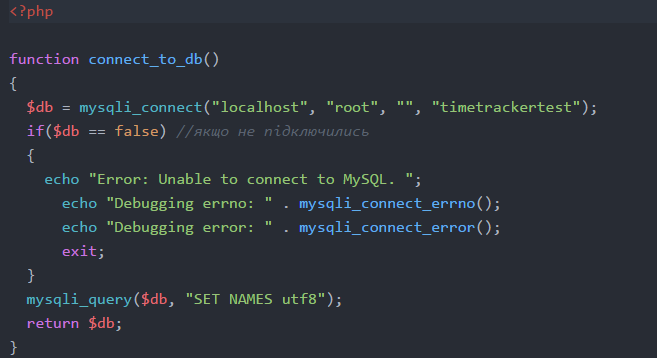
Маючи інформацію про роботу програмістів, менеджери/керівники зможуть оцінювати хто впорався найкраще, а хто — найгірше, а також начисляти відповідно премії чи штрафи.

Під час виробничої практики я уточнював вимоги до проекту, спілкуючись з менеджерами фірми "Юкон-Софтваре". Основні вимоги були такі:

* Можливість створювати групи та редагувати їх.
* Можливість додавати користувачів до вибраної групи або видаляти їх.
* Можливість додавати менеджерів до вибраної групи або видаляти їх.
* Система трекінгу повинна відправляти сповіщення керівникам/менеджерам про те, що програміст незаповнив інформацію про виконану роботу за день.

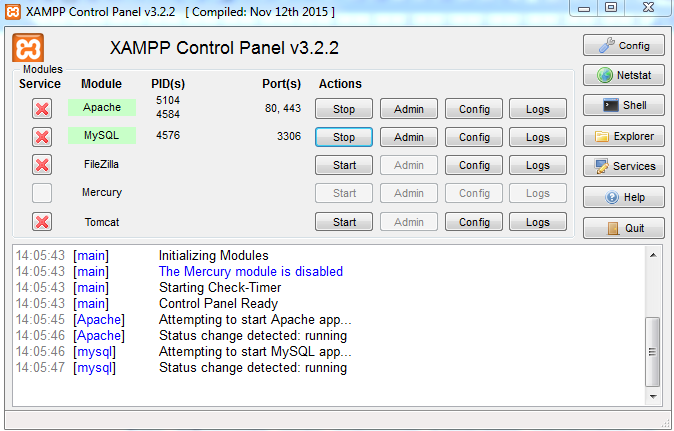
3.2. Робота з базою даних

Щоб підключитись до бази даних, я використовую мову php (рис.3.1).



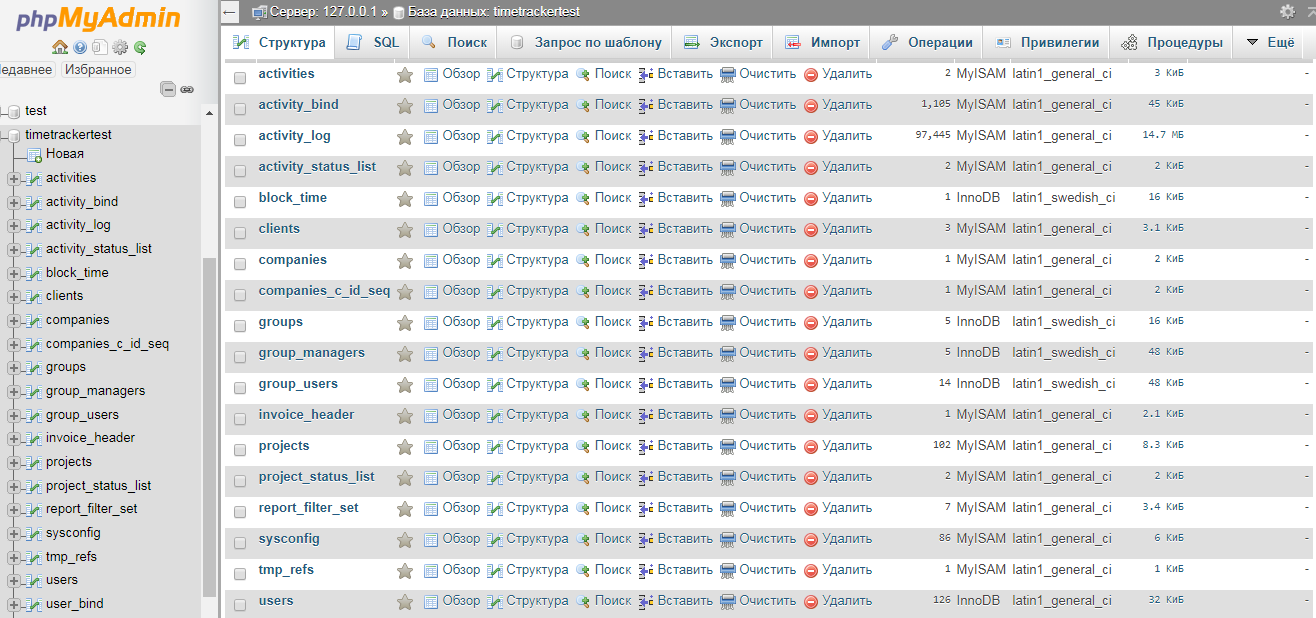
*Рис.3.1. Підключення до бази даних*

Щоб працювати з сервером та базою даних далі, я буду використовувати програму XAMPP (рис 3.2).



*Рис.3.2. Робота програми xampp*

Після запуску сервера apache і MySQL, можна приступити до створення таблиць в базі даних в phpmyadmin (рис 3.3).

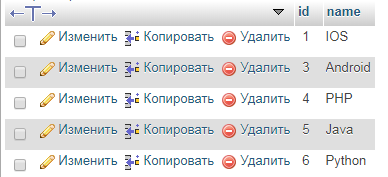


*Рис.3.3. Таблиці в phpmyadmin*

Спочатку задача полягала в тому, щоб створити 3 нові таблиці в базі даних:

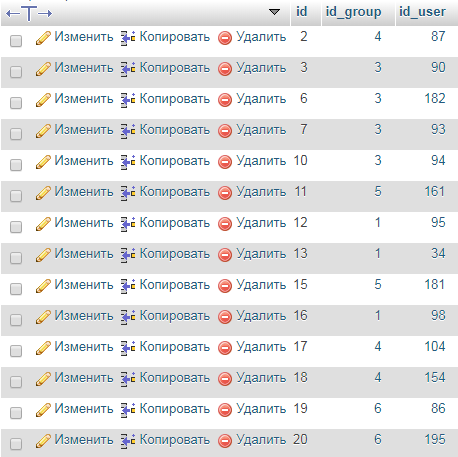
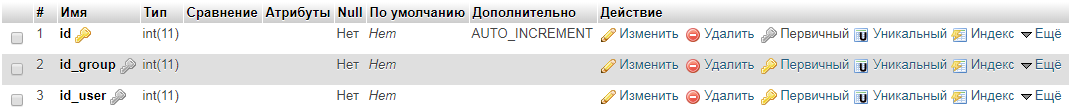
* groups
* group\_users
* group\_managers

Таблиця groups містить id групи та назву: (рис 3.4)



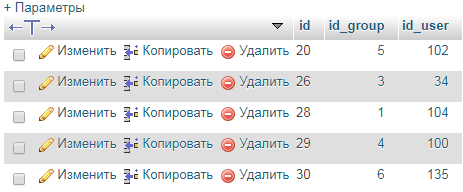
*Рис.3.4. Таблиця groups*

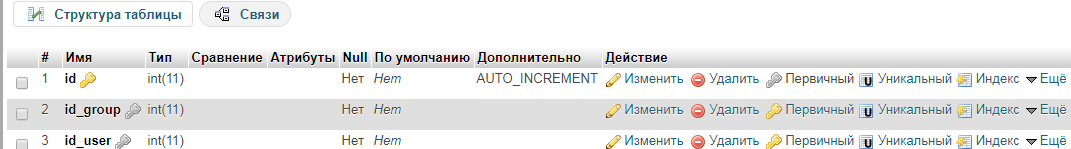
Таблиця group\_ussers містить id, id\_group, id\_user. Оскільки я зв'язав таблиці за допомогою первинного та вторинного ключів, таблиці можуть користуватись інформацією з інших пов'язаних таблиць: (рис 3.5)

*Рис.3.5. Таблиця group\_users*

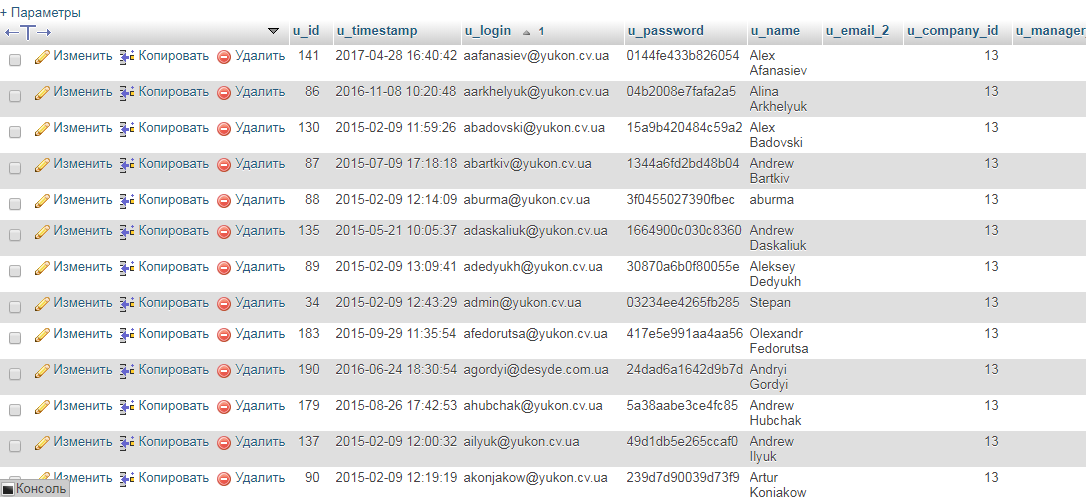
Подібним чином зроблена і таблиця group\_managers (рис. 3.6).





*Рис.3.6. Таблиця group\_managers*

Таблиця users вже існувала. З цієї таблиці в подальшому буде використовуватись інформація для заповнення груп користувачами (рис 3.7).

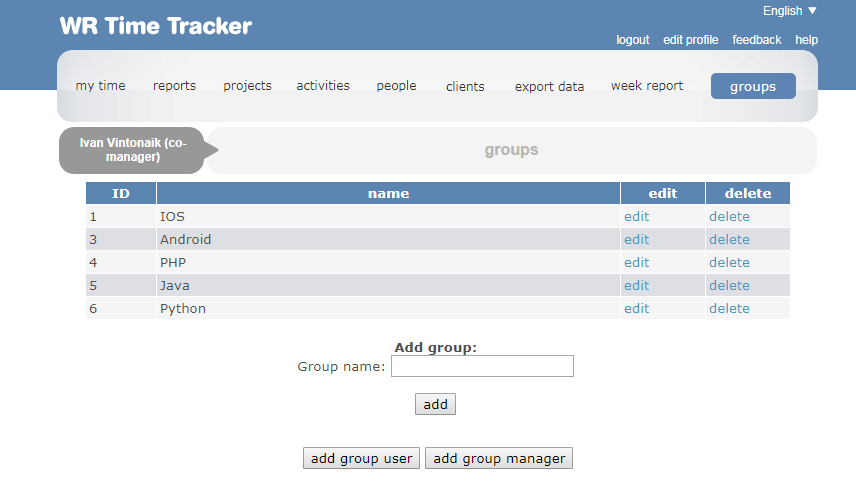


*Рис.3.7. Таблиця users*

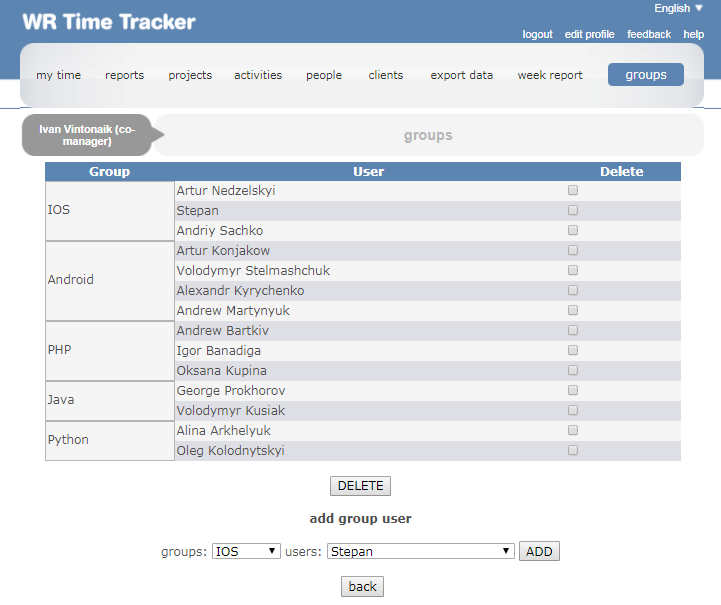
3.3. Опис функціоналу веб-додатку

Після створення всіх необхідних таблиць в базі даних та успішного підключення бд до серверу можна переходити до веб-інтерфейсу. Було створено багато нових веб-сторінок. А саме:

Сторінка, де можна створити нову групу, редагувати або видалити існуючі має наступний вигляд (рис 3.8)

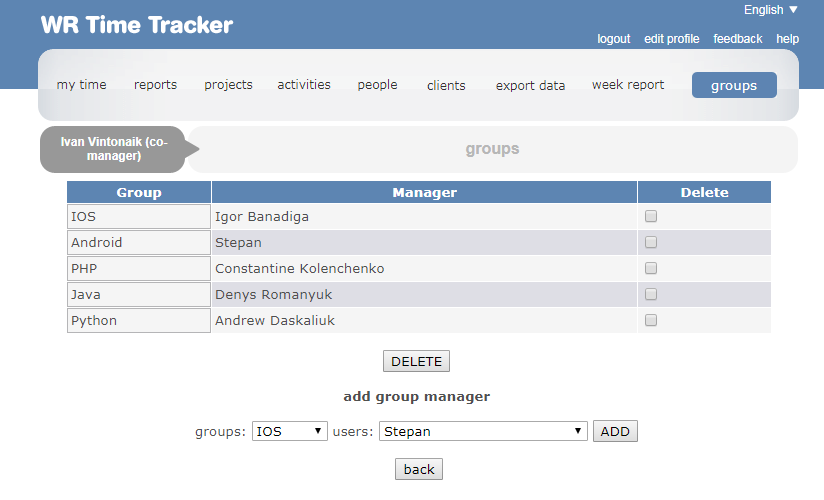
 *Рис.3.8. Загальний вигляд сторінки "Групи", яка надає можливості створення, редагування та знищення груп користувачів*

Якщо натиснути на кнопку "add group user", то ми перейдемо на нову сторінку, де можемо додавати/видаляти користувачів до/з вибраної групи (рис 3.9).



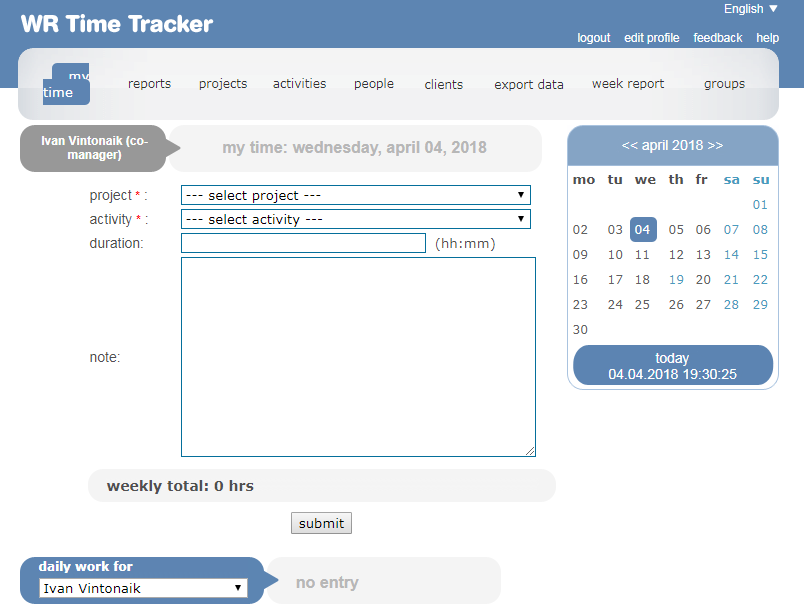
*Рис.3.9. Загальний вигляд сторінки "Додавання користувачів", яка надає можливості додавання та знищення користувачів в групах*

Якщо натиснути на кнопку "add group manager", то ми перейдемо на сторінку, де можемо додавати/видаляти менеджерів до/з вибраної групи (рис 3.10).



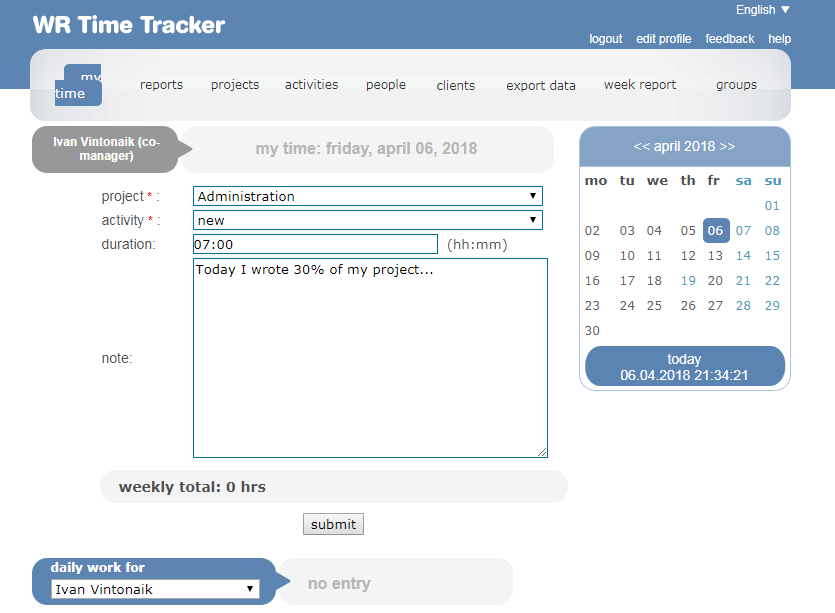
*Рис.3.10. Загальний вигляд сторінки "Додавання менеджерів", яка надає можливості додавання та знищення менеджерів в групах*

Основною задачею проекту було зробити сторінку, де програміст вписуває все, що він зробив і скільки часу він на це потратив. Цю інформацію девелопер повинен писати щодня. Якщо ж програміст не написав нічого, то рівно в 20.00 менеджеру групи приходите сповіщення на емейл зі списком тих девелоперів, які не заповнили інформацію про свою денну роботу. Сторінка трекінгу має наступний вигляд (рис 3.11).



*Рис.3.11. Загальний вигляд сторінки "Додавання Тайм-трекеру", яка надає можливість користувачам заповнювати інформацію про виконану роботу*

Першим ділом потрібно вибрати дату, за який час буде звітуватись програміст. Це можна зробити справа в календарі. Далі у випадаючому списку потрібно вибрати назву проекту та який саме захід. Далі вказується кількість потрачених на роботу годин. Ну і детальний опис виконаної роботи. Ці всі дії може виконувати програміст, який працює у фірмі "Юкон=Софтваре" і є в базі даних в таблиці "users". Перед тим як натиснути кнопку "submit", потрібно вибрати ім'я із списку існуючих людей в базі даних (рис 3.12).



*Рис.3.12. Приклад заповнення інформації користувачеи*

ВИСНОВОК

Було створено веб-застосунок для моніторингу роботи програмістів, а також наповнено базу даних інформацією про працівників підприємства, згрупованою за технологіями програмування.

Після написання проекту системи трекінгу та створення груп з користувачами та менеджерами, ТоВ "Юкон-Софтваре" відразу почала використовувати цю систему. Кожен програміст повинен відтепер відчитуватись за свою роботу. Через тиждень роботи системи трекінгу, керівники та директор побачив, хто з програмістів дійсно працюють, а хто несерйозно ставиться до роботи. Цим самим він має змогу комусь підняти зарплату чи виписати премію, а а комусь виписати штраф. Після декількох штрафіф, всі співробітники стали відноситись до роботи серйозніше.

Отже, система трекінгу дійсно оптимізувала та покращила роботу фірми.

СПИСОК ПОСИЛАНЬ

1. Навчальні програми і приклади для вивчення HTML, CSS, SQL, PHP та багатьох інших технологій веб-розробки [Електронний  ресурс].**–**Режим доступу**:** <https://www.w3schools.com>
2. Офіційний сайт MySQL [Електронний ресурс].**–**Режим доступу**:** <http://dev.mysql.com>
3. Лінн Бейлі "Изучаем SQL" − Пітер − 2000. − 490с.
4. Інформація про смарти [Електронний ресурс]. **–** Режим доступу: https://uk.wikipedia.org/wiki/Smarty
5. Інформація про javascript [Електронний ресурс]. **–** Режим доступу: https://uk.wikipedia.org/wiki/JavaScript
6. Створення форм в мові HTML [Електронний ресурс]. **–** Режим доступу: http://sites.znu.edu.ua/webprog/lect/1215.ukr.html
7. Програмування по ураїнськи на php [Електронний ресурс]. **–** Режим доступу: http://programming.in.ua/web-design/allphp/30-about-php.html

ДОДАТОК (КОД ПРОГРАМИ)

Програмний код реалізовано на мові php, з використанням додатково мов MySQL, HTML, Javascript, а також CSS, Smarty. Код написано в текстовому редакторі Atom.

Код розділено на різні файли.

**timetracker\_cron.php:**

<?php

require "./config.php";

require "../user\_groups/db.php";

$sqlSelect = "select al\_user\_id from activity\_log as a"; //витягаєм лише ід юзерів які щось зробили за останню добу

if (date('G',time())>= '14' ){

$sqlWhere = " where a.al\_date = DATE\_FORMAT(NOW(), '%Y-%m-%d')";

}else{

$sqlWhere = " where a.al\_date = DATE\_FORMAT(NOW() — interval 1 day, '%Y-%m-%d')";

}

$headers = 'From: timetracker@yukon.cv.ua' . "\r\n" .

'X-Mailer: PHP/' . phpversion();

$link = connect\_to\_db();

$managers = get\_group\_managers($link);

foreach ($managers as $manager)//для кожного менеджера кожної групи

{

$strList='';//список людей які нічого не зробили

$sql = "SELECT group\_users.id\_user, users.u\_login as u\_email, users.u\_name FROM group\_users INNER JOIN users ON group\_users.id\_user = users.u\_id WHERE id\_group = '$manager[group\_id]' AND group\_users.id\_user NOT IN ($sqlSelect $sqlWhere)";//витягуємо юзерів хто нічого не зробив у поточній групі

$result = mysqli\_query($link, $sql) or die('Query error ' . mysqli\_error());

while ($row = mysqli\_fetch\_assoc($result))

{

$strList.= "$row[u\_name] ($row[u\_email])\n"; //додаємо до списку імя боржника

}

if ($strList != '')//якщо є боржники то повідомляємо менеджера

{

mail ($manager['email'],'Somebody from your team didn\'t fill in timetracker',$strList,$headers);

}

}

mysqli\_free\_result($result);

mysqli\_close($link);

?>

**db.php**

<?php

function connect\_to\_db()

{

$db = mysqli\_connect("localhost", "root", "", "timetrackertest");

if($db == false) //якщо не підключились

{

echo "Error: Unable to connect to MySQL. ";

echo "Debugging errno: " . mysqli\_connect\_errno();

echo "Debugging error: " . mysqli\_connect\_error();

exit;

}

mysqli\_query($db, "SET NAMES utf8");

return $db;

}

function Groups\_info($db) //зчитування

{

$query = "SELECT \* FROM groups";

$res = mysqli\_query($db, $query);

$programers = [];

while ($row = mysqli\_fetch\_assoc($res))

{

$programers[] = $row;

}

return $programers;

}

function add\_group($db, $group\_info) //створюєм нову групу

{

$res = mysqli\_query($db, "INSERT INTO groups (name)

VALUES ('$group\_info[name]')");

if($res == false)

{

echo mysqli\_error($db);

}

}

function get\_groups($db)

{

$res = mysqli\_query($db, "SELECT \* FROM groups");

$groups = [];

while ($row = mysqli\_fetch\_assoc($res))

{

$groups[] = $row;

}

return $groups;

}

function get\_users($db) //зчитує імена та емейли з таблиці юзер

{

$res = mysqli\_query($db, "SELECT `u\_id`, `u\_name`, `u\_login` FROM users");

$users = [];

while ($row = mysqli\_fetch\_assoc($res))

{

$users[] = $row;

}

return $users;

}

function get\_group\_users($db)//зчитує з бд користувачів груп

{

//SELECT \* FROM group\_users, groups WHERE group\_users.id\_group = groups.id

$res = mysqli\_query($db, "SELECT group\_users.id, groups.name, users.u\_name FROM group\_users INNER JOIN groups INNER JOIN users ON group\_users.id\_group = groups.id AND group\_users.id\_user = users.u\_id");

$users = [];

while ($row = mysqli\_fetch\_assoc($res))

{

$users[] = $row;

}

return $users;

}

function get\_group\_managers($db)//зчитує з бд менеджерів груп

{

$res = mysqli\_query($db, "SELECT group\_managers.id, groups.name, groups.id as group\_id, users.u\_name, users.u\_login as email FROM group\_managers INNER JOIN groups INNER JOIN users ON group\_managers.id\_group = groups.id AND group\_managers.id\_user = users.u\_id");

$users = [];

while ($row = mysqli\_fetch\_assoc($res))

{

$users[] = $row;

}

return $users;

}

function add\_group\_user($db, $group\_user\_info) //додаємо користувача до групи

{

$res = mysqli\_query($db, "INSERT INTO group\_users(id\_group, id\_user)

VALUES ('$group\_user\_info[group]', '$group\_user\_info[user]')");

if($res == false)

{

echo mysqli\_error($db);

}

}

function add\_group\_manager($db, $group\_manager\_info) //додаємо менеджера групи

{

$res = mysqli\_query($db, "INSERT INTO group\_managers(id\_group, id\_user)

VALUES ('$group\_manager\_info[group]', '$group\_manager\_info[user]')");

if($res == false)

{

echo mysqli\_error($db);

}

}

function get\_group\_by\_id($db, $id) //витягує 1 користувача з поточним ід

{

$id = mysqli\_real\_escape\_string($db, $id);

$query = "SELECT \* FROM groups WHERE id = '$id'";

$res = mysqli\_query($db, $query);

$programers = [];

while ($row = mysqli\_fetch\_assoc($res))

{

$programers[] = $row;

}

return $programers;

}

function update\_group($db, $id, $new\_data)//змінюємо назву групи

{

$id = mysqli\_real\_escape\_string($db, $id);

$query = "UPDATE groups SET name = '$new\_data[name]' WHERE id = '$id'";

$res = mysqli\_query($db, $query);

if($res == false)

{

echo mysqli\_error($db);

}

}

function delete\_group($db, $id)//видалення групи

{

$id = mysqli\_real\_escape\_string($db, $id);

$query = "DELETE FROM groups WHERE id = '$id'";

$res = mysqli\_query($db, $query);

if($res == false)

{

echo mysqli\_error($db);

}

}

function mass\_delete\_group\_user($db, $row\_ids) //масове видалення користувачів з групи

{

$res = mysqli\_query($db, "DELETE FROM group\_users WHERE id IN(" . implode(",", $row\_ids) . ")");

if($res == false)

{

echo mysqli\_error($db);

}

}

function mass\_delete\_group\_manager($db, $row\_ids) //масове видалення менеджерів з групи

{

$res = mysqli\_query($db, "DELETE FROM group\_managers WHERE id IN(" . implode(",", $row\_ids) . ")");

if($res == false)

{

echo mysqli\_error($db);

}

}

?>

**group\_add\_user.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('SysConfig');

$db = DB::connect(DSN);

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

} else {

Header("Location: login.php");

exit();

}

include ("./user\_groups/db.php");

$link = connect\_to\_db();

if (!empty($\_POST))//якщо натиснули кнопку отправить

{

add\_group\_user($link, $\_POST);

}

$gr\_users = get\_group\_users($link);

//групування людей по групам, обєднуєм клітинки

$groupped = [];

foreach ($gr\_users as $row)

{

$groupped[$row['name']][$row['id']] = $row['u\_name'];

}

$rows = "";

foreach ($groupped as $key => $row) //всі люди згруповані

{

$person\_count = count($row);

$k=1;

foreach ($row as $person\_id => $person\_name)

{

$rows .= "<tr>";

if ($k == 1)

{

$rows .= "<td rowspan = '$person\_count' class = 'name'> $key </td>";

}

$rows .= "<td class = 'info'>$person\_name</td><td class = 'info'><input type = 'checkbox' name='id[]' value='$person\_id'></td></tr>";

$k++;

}

} //key — назва групи, row — користувач в ній

$groups = get\_groups($link);

$users = get\_users($link);

$smarty->assign\_by\_ref("group\_members",$rows);

$smarty->assign\_by\_ref("groups",$groups);

$smarty->assign\_by\_ref("users",$users);

$smarty->assign\_by\_ref("userdet\_string",UserHelper::getUserDetailsString($user,$GLOBALS["I18N"]));

$smarty->assign\_by\_ref("title\_page",$i18n->getKey("groups.page\_title"));

$smarty->assign\_by\_ref("group\_users\_add",$i18n->getKey("group.users.add"));

$smarty->assign\_by\_ref("groups\_list",$i18n->getKey("groups.list"));

$smarty->assign\_by\_ref("groups\_users\_list",$i18n->getKey("groups.users.list"));

$smarty->assign("content\_page\_name","group\_add\_user.tpl");

$smarty->display(INDEX\_TEMPLATE);

?>

<script>

document.getElementById('groups').className = 'selectedScreen';

</script>

**group\_add\_managers.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('SysConfig');

$db = DB::connect(DSN);

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

} else {

Header("Location: login.php");

exit();

}

include ("./user\_groups/db.php");

$link = connect\_to\_db();

if (!empty($\_POST))//якщо натиснули кнопку отправить

{

add\_group\_manager($link, $\_POST);

}

$gr\_users = get\_group\_managers($link);

//групування людей по групам, обєднуєм клітинки

$groupped = [];

foreach ($gr\_users as $row)

{

$groupped[$row['name']][$row['id']] = $row['u\_name'];

}

$rows = "";

foreach ($groupped as $key => $row) //всі люди згруповані

{

$person\_count = count($row);

$k=1;

foreach ($row as $person\_id => $person\_name)

{

$rows .= "<tr>";

if ($k == 1)

{

$rows .= "<td rowspan = '$person\_count' class = 'name'> $key </td>";

}

$rows .= "<td class = 'info'>$person\_name</td><td class = 'info'><input type = 'checkbox' name='id[]' value='$person\_id'></td></tr>";

$k++;

}

} //key — назва групи, row — користувач в ній

$groups = get\_groups($link);

$users = get\_users($link);

$smarty->assign\_by\_ref("group\_managers",$rows);

$smarty->assign\_by\_ref("groups",$groups);

$smarty->assign\_by\_ref("users",$users);

$smarty->assign\_by\_ref("userdet\_string",UserHelper::getUserDetailsString($user,$GLOBALS["I18N"]));

$smarty->assign\_by\_ref("title\_page",$i18n->getKey("groups.page\_title"));

$smarty->assign\_by\_ref("groups\_list",$i18n->getKey("groups.list"));

$smarty->assign\_by\_ref("groups\_users\_list",$i18n->getKey("groups.users.list"));

$smarty->assign\_by\_ref("group\_managers\_add",$i18n->getKey("group.managers.add"));

$smarty->assign("content\_page\_name","group\_add\_manager.tpl");

$smarty->display(INDEX\_TEMPLATE);

?>

<script>

document.getElementById('groups').className = 'selectedScreen';

</script>

**group\_delete\_manager.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('SysConfig');

$db = DB::connect(DSN);

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

} else {

Header("Location: login.php");

exit();

}

include ("./user\_groups/db.php");

$link = connect\_to\_db();

if (!empty($\_GET['id']))

{

mass\_delete\_group\_manager($link, $\_GET['id']);

}

header("Location: ./group\_add\_manager.php");

**group\_delete\_user.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('SysConfig');

$db = DB::connect(DSN);

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

} else {

Header("Location: login.php");

exit();

}

include ("./user\_groups/db.php");

$link = connect\_to\_db();

if (!empty($\_GET['id']))

{

mass\_delete\_group\_user($link, $\_GET['id']);

}

header("Location: ./group\_add\_user.php");

**group\_delete.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('SysConfig');

$db = DB::connect(DSN);

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

} else {

Header("Location: login.php");

exit();

}

include ("./user\_groups/db.php");

$link = connect\_to\_db();

if (!empty($\_GET['id']))

{

delete\_group($link, $\_GET["id"]);

}

header("Location: ./groups.php");

**group\_edit.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('SysConfig');

$db = DB::connect(DSN);

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

} else {

Header("Location: login.php");

exit();

}

include ("./user\_groups/db.php");

$link = connect\_to\_db();

$group\_info = get\_group\_by\_id($link, $\_GET["id"]);

$error\_text = "";

if (!empty($\_POST))//якщо натиснули кнопку отправить

{

if (empty($\_POST["name"]))

{

$error\_text = "Заповніть поле \"Назва групи\"";

}

else

{

update\_group($link, $\_GET["id"], $\_POST);

header("Location: ./groups.php");

}

}

$smarty->assign("error\_text", $error\_text);

$smarty->assign("group\_id", $\_GET["id"]);

$smarty->assign\_by\_ref("group\_info",$group\_info[0]);

$smarty->assign\_by\_ref("userdet\_string",UserHelper::getUserDetailsString($user,$GLOBALS["I18N"]));

$smarty->assign\_by\_ref("title\_page",$i18n->getKey("groups.page\_title"));

$smarty->assign("content\_page\_name","group\_edit.tpl");

$smarty->display(INDEX\_TEMPLATE);

?>

<script>

document.getElementById('groups').className = 'selectedScreen';

</script>

**groups.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('TimeHelper');

import('MyDateTime');

import('SysConfig');

$db = DB::connect(DSN);

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

} else {

Header("Location: login.php");

exit();

}

// elements of form 'behalfForm'

$sform = new Form('behalfForm');

if ($user->isManager() || $user->isCoManager()) {

$user\_list = UserHelper::findAllUsers($user);

$sform->addInput(array("type"=>"combobox",

"onchange"=>"if(this.form) this.form.submit();",

"name"=>"behalfUser",

"value"=>$in\_behalf\_id,

"data"=>$user\_list,

"datakeys"=>array("u\_id","u\_name"),

));

$sform->addInput(array("type"=>"submit","name"=>"bhvsubmit","value"=>$i18n->getKey('button.behalf\_set')));

//$sform->addInput(array("type"=>"hidden", "name"=>"date", "value"=>$cl\_date));

}

include ("./user\_groups/db.php");

$link = connect\_to\_db();

if (!empty($\_POST))//якщо натиснули кнопку отправить

{

if (empty($\_POST["name"]))

{

$error\_text = $i18n->getKey("groups.empty.field.name");

}

else

{

add\_group($link, $\_POST);

}

}

$programers = Groups\_info($link);

$smarty->assign("forms",array($sform->getName()=>$sform->toArray()) );

$smarty->assign("programers", $programers);

$smarty->assign("error\_text", $error\_text);

$smarty->assign\_by\_ref("userdet\_string",UserHelper::getUserDetailsString($user,$GLOBALS["I18N"]));

$smarty->assign\_by\_ref("title\_page",$i18n->getKey("groups.page\_title"));

$smarty->assign\_by\_ref("group\_name",$i18n->getKey("groups.name"));

$smarty->assign\_by\_ref("group\_id",$i18n->getKey("groups.id"));

$smarty->assign\_by\_ref("edit\_group",$i18n->getKey("groups.th.edit"));

$smarty->assign\_by\_ref("delete\_group",$i18n->getKey("groups.th.del"));

$smarty->assign\_by\_ref("confirm\_delete\_group",$i18n->getKey("groups.th.confirm.del"));

$smarty->assign\_by\_ref("add\_group",$i18n->getKey("groups.add"));

$smarty->assign\_by\_ref("group\_field\_name",$i18n->getKey("groups.field.name"));

$smarty->assign\_by\_ref("button\_add\_group",$i18n->getKey("groups.button.add"));

$smarty->assign\_by\_ref("add\_group\_users",$i18n->getKey("group.users.add"));

$smarty->assign\_by\_ref("add\_group\_managers",$i18n->getKey("group.managers.add"));

$smarty->assign("content\_page\_name","groups.tpl");

$smarty->display(INDEX\_TEMPLATE);

?>

<script>

document.getElementById('groups').className = 'selectedScreen';

</script>

**mytime.php**

<?php

require\_once('initialize.php');

import('form.Form');

import('UserHelper');

import('TimeHelper');

import('MyDateTime');

import('SysConfig');

$db = DB::connect(DSN);

// init and store date in session

$cl\_date = $request->getParameter('date',@$\_SESSION['date']);

$crdate = new MyDateTime(SYS\_DATEFORMAT, $cl\_date);

if($crdate->isError())

$crdate = new MyDateTime(SYS\_DATEFORMAT);

if(!$cl\_date)

$cl\_date = $crdate->toString(SYS\_DATEFORMAT);

$\_SESSION['date'] = $cl\_date;

if ($auth->isAuthenticated()) {

$user = new User($auth->getUserId());

if ($user->isAdministrator()) {

Header("Location: admin.php");

exit();

}

} else {

Header("Location: login.php");

exit();

}

$cl\_start = $request->getParameter('start');

$cl\_finish = $request->getParameter('finish');

$cl\_duration = $request->getParameter('duration');

$cl\_note = $request->getParameter('note');

if ($request->getMethod()=="POST") {

$cl\_billable = $request->getParameter('billable');

} else {

$cl\_billable = 1;

}

// init and store behalf\_id in session

$in\_behalf\_id = $request->getParameter('behalfUser',(isset($\_SESSION['behalf\_id'])?$\_SESSION['behalf\_id']:$user->getUserId()));

$\_SESSION['behalf\_id'] = $in\_behalf\_id;

$cl\_project = $request->getParameter('project',($request->getMethod()!="POST"?@$\_SESSION['project']:null));

$\_SESSION['project'] = $cl\_project;

$cl\_activity = $request->getParameter('activity',($request->getMethod()!="POST"?@$\_SESSION['activity']:null));

$\_SESSION['activity'] = $cl\_activity;

$cl\_chperiod = $request->getParameter('chPeriod',($request->getMethod()!="POST"?@$\_SESSION['chPeriod']:null));

if ($cl\_chperiod==null) {

$sc = new SysConfig($user);

$cl\_chperiod = $sc->getValue(SYSC\_CHART\_PERIOD);

}

if (!$cl\_chperiod) $cl\_chperiod = 5;

$\_SESSION['chPeriod'] = $cl\_chperiod;

// elements of form 'behalfForm'

$sform = new Form('behalfForm');

if ($user->isManager() || $user->isCoManager()) {

$user\_list = UserHelper::findAllUsers($user);

$sform->addInput(array("type"=>"combobox",

"onchange"=>"if(this.form) this.form.submit();",

"name"=>"behalfUser",

"value"=>$in\_behalf\_id,

"data"=>$user\_list,

"datakeys"=>array("u\_id","u\_name"),

));

$sform->addInput(array("type"=>"submit","name"=>"bhvsubmit","value"=>$i18n->getKey('button.behalf\_set')));

//$sform->addInput(array("type"=>"hidden", "name"=>"date", "value"=>$cl\_date));

}

// elements of form 'chartForm'

$chperiod\_data = array();

$chperiod\_data["1"]=$i18n->getKey('controls.per\_td');

$chperiod\_data["2"]=$i18n->getKey('controls.per\_tw');

$chperiod\_data["3"]=$i18n->getKey('controls.per\_tm');

$chperiod\_data["4"]=$i18n->getKey('controls.per\_ty');

$chperiod\_data["5"]=$i18n->getKey('controls.per\_at');

$form3 = new Form('chartForm');

$form3->addInput(array("type"=>"combobox",

"onchange"=>"if(this.form) this.form.submit();",

"name"=>"chPeriod",

"value"=>$cl\_chperiod,

"data"=>$chperiod\_data

));

$form3->addInput(array("type"=>"submit","name"=>"chsubmit","value"=>$i18n->getKey('button.save')));

// elements of form 'mytimeForm'

import('ProjectHelper');

$project\_list = ProjectHelper::findAllProjects($user, true);

$form = new Form('mytimeForm');

$form->addInput(array("type"=>"combobox",

"onchange"=>"fillActivityDir();",

"name"=>"project",

"style"=>"width: 350; border: 1px solid #056F9B",

"value"=>$cl\_project,

"data"=>$project\_list,

"datakeys"=>array("p\_id","p\_name"),

"empty"=>array(""=>$i18n->getKey('controls.select.project')),

"tabindex"=>"1"

));

import('ActivityHelper');

$activity\_list = ActivityHelper::findAllActivity($user);

$form->addInput(array("type"=>"combobox",

"name"=>"activity",

"style"=>"width: 350; border: 1px solid #056F9B",

"value"=>$cl\_activity,

"data"=>$activity\_list,

"datakeys"=>array("a\_id","a\_name"),

"empty"=>array(""=>$i18n->getKey('controls.select.activity')),

"tabindex"=>"2"

));

$form->addInput(array("type"=>"text","name"=>"start","size"=>"28","style"=>"border: 1px solid #056F9B","value"=>$cl\_start,"onchange"=>"formDisable('start');", "onblur"=>"strValidTime(this.id);" ));

$form->addInput(array("type"=>"text","name"=>"finish","size"=>"28","style"=>"border: 1px solid #056F9B","value"=>$cl\_finish,"onchange"=>"formDisable('finish');", "onblur"=>"strValidTime(this.id);"));

$form->addInput(array("type"=>"text","name"=>"duration","size"=>"28","style"=>"border: 1px solid #056F9B","value"=>$cl\_duration,"onchange"=>"formDisable('duration');", "onblur"=>"strValidTime(this.id);", "tabindex"=>"3"));

$form->addInput(array("type"=>"textarea","name"=>"note","style"=>"width: 355; height: 200; border: 1px solid #056F9B","value"=>$cl\_note, "tabindex"=>"4"));

$form->addInput(array("type"=>"calendar", "name"=>"date", "value"=>$cl\_date, "sysdateformat"=>SYS\_DATEFORMAT, "user"=>$user->getActiveUser() )); // calendar

$form->addInput(array("type"=>"submit","name"=>"btmytime","value"=>$i18n->getKey('button.submit'), "onclick"=>"return MandatoryField(this.id);", "tabindex"=>"5" ));

$locktime = $user->getLocktime();

$lockdate = 0;

if ($locktime<0 || $locktime==null || $locktime=="") {

$sc = new SysConfig(new User($user->getOwnerId(), false));

$locktime = $sc->getValue(SYSC\_LOCK\_DAYS);

}

if ($locktime>0) {

$cl\_date\_now = $request->getParameter('date\_now');

if ($cl\_date\_now) {

$lockdate = new MyDateTime($i18n->getDateTimeFormat(), $cl\_date\_now);

//$lockdate = new MyDateTime();

$lockdate->decDay($locktime);

}

}

$date = strtotime("+ 14 hours +1 day",strtotime($\_SESSION['date']));

$now=mktime(date("H"), date("i"), 0, date("m") , date("d"), date("Y"));

$date= date("Y-m-d H:i:s",$date);

$now= date("Y-m-d H:i:s",$now);

$q ="SELECT ALL\_PERS, CO\_M from block\_time";

$sth = $db->prepare($q);

$rc = $db->execute($sth);

if (DB::isError($rc) == 0) {

while ($val = $rc->fetchRow()) {

$check\_all=$val[0];

$ckeck\_co=$val[1];

}

}

if ($request->getMethod()=="POST") {

$flag=false;

$q ="SELECT ALL\_PERS, CO\_M from block\_time";

$sth = $db->prepare($q);

$rc = $db->execute($sth);

if (DB::isError($rc) == 0) {

while ($val = $rc->fetchRow()) {

$check\_all=$val[0];

$ckeck\_co=$val[1];

}

}

if($check\_all==1) $flag=true;

if($check\_all==0 && $ckeck\_co==1){

if($user->isCoManager()){

$flag=true;

}

else

{

$flag=false;

}

}

if($date>$now || $flag==true ){

if ($request->getParameter("btmytime")) {

// check input data

if (!$cl\_project || $cl\_project<=0) {

$errors->add("project",$i18n->getKey("errors.wr\_project"));

}

if (!$cl\_activity || $cl\_activity<=0) {

$errors->add("activity",$i18n->getKey("errors.wr\_activity"));

}

if (!$cl\_duration) {

if (isset($cl\_start) || isset($cl\_finish)) {

if (!TimeHelper::isValidTime($cl\_start)) {

$errors->add("start",$i18n->getKey("errors.wrong"),$i18n->getKey("form.mytime.start"));

}

if ($cl\_finish)

if (!TimeHelper::isValidTime($cl\_finish)) {

$errors->add("finish",$i18n->getKey("errors.wrong"),$i18n->getKey("form.mytime.finish"));

}

/\*if (!TimeHelper::isCorrectTimeInterval($cl\_start, $cl\_finish)) {

$errors->add("interval",$i18n->getKey("errors.wr\_interval"));

}\*/

} else {

$errors->add("start",$i18n->getKey("errors.empty"),$i18n->getKey("form.mytime.start"));

$errors->add("finish",$i18n->getKey("errors.empty"),$i18n->getKey("form.mytime.finish"));

$errors->add("duration",$i18n->getKey("errors.empty"),$i18n->getKey("form.mytime.duration"));

}

} else {

if(strstr($cl\_duration, "."))

$cl\_duration = decimal\_to\_time($cl\_duration);

if (!TimeHelper::isValidDuration($cl\_duration)) {

$errors->add("duration",$i18n->getKey("errors.wrong"),$i18n->getKey("form.mytime.duration"));

}

}

// finish check input data

if($lockdate && $crdate->before($lockdate)) {

$errors->add("lock\_period",$i18n->getKey("errors.period\_lock"));

}

if (ProjectHelper::isProjectExistsStrict($cl\_project, $user) &&

ActivityHelper::isActivityExistsStrict($cl\_activity, $user) &&

$errors->isEmpty()) {

if($no\_finished\_rec = TimeHelper::checkPresentTimeWithEmptyDuration($user->getUserId()) AND ($cl\_finish=="" AND $cl\_duration=="")) {

$errors->add("insert", "uncompleted entry already exists, close previous one. <A HREF='mytime\_edit.php?ts=".$no\_finished\_rec['al\_timestamp']."&date=".date("m/d/Y", strtotime($no\_finished\_rec['al\_date']))."'>Go to record</A>");

}

else {

if ($r = TimeHelper::insert($crdate->toString(DB\_DATEFORMAT), $user->getActiveUser(), $cl\_project, $cl\_activity, $cl\_start, $cl\_finish, $cl\_duration, $cl\_note, "", $cl\_billable)) {

if (count($r)>1) {

$d = new MyDateTime(DB\_DATEFORMAT, $r[1]["r\_date"]);

header("Location: mytime.php?date=".$d->toString(SYS\_DATEFORMAT));

}

else {

header("Location: mytime.php");

}

exit();

}

$errors->add("insert",$i18n->getKey("errors.mt\_insert"));

}

}

}

if ($request->getParameter("chPeriod")) {//echo SYSC\_CHART\_PERIOD." = $cl\_chperiod";

$sc = new SysConfig($user);

$sc->setValue(SYSC\_CHART\_PERIOD, $cl\_chperiod);

header("Location: mytime.php");

exit;

}

}

else

{

$errors->add("insert", "Access Denied");

}

if ($request->getParameter("behalfUser")) {

// if user is manager

if($user->isManager() || $user->isCoManager()) {

$user->setBehalfId($in\_behalf\_id);

$user->reload();

header("Location: mytime.php");

exit();

}

}

}

//End if//

$ud = UserHelper::findUserById($user->getOwnerId(), new User($user->getOwnerId(),false));

$week\_time = TimeHelper::getTimePerWeek($user, $crdate);

$smarty->assign\_by\_ref("errors", $errors);

$smarty->assign\_by\_ref("messages", $messages);

$smarty->assign\_by\_ref("time\_list", TimeHelper::findAllTime($crdate->toString(DB\_DATEFORMAT),$user->getActiveUser()) );

$smarty->assign\_by\_ref("total\_time", TimeHelper::getTotalTime($crdate->toString(DB\_DATEFORMAT),$user->getActiveUser()) );

$smarty->assign\_by\_ref("curr\_date", $crdate->toString() );

$smarty->assign\_by\_ref("project\_list", $project\_list );

$smarty->assign\_by\_ref("activity\_list", $activity\_list );

$smarty->assign("forms",array($form->getName()=>$form->toArray(),$sform->getName()=>$sform->toArray(),$form3->getName()=>$form3->toArray()) );

$smarty->assign("onload","onLoad = \"document.mytimeForm.project.focus();fillActivityDir();\"");

$smarty->assign\_by\_ref("timestring",TimeHelper::parseTimeString($crdate));

$smarty->assign\_by\_ref("userdet\_string",UserHelper::getUserDetailsString($user,$GLOBALS["I18N"]));

$smarty->assign\_by\_ref("week\_time", $week\_time);

if($ud['u\_show\_pie']!=0) {

if($ud['u\_pie\_mode']==2)

$smarty->assign("chart\_href", "chart.php?r=".md5(uniqid("random"))."&period=".$cl\_chperiod."&pie\_mode=project");

else

$smarty->assign("chart\_href", "chart.php?r=".md5(uniqid("random"))."&period=".$cl\_chperiod);

}

$mt = "mytime";

$smarty->assign\_by\_ref("mytime", $mt);

$smarty->assign\_by\_ref("title\_page",$i18n->getKey("form.mytime.title"));

$smarty->assign("content\_page\_name","mytime.tpl");

$smarty->display(INDEX\_TEMPLATE);

?>

<script>

document.getElementById('mytime').className = 'selectedScreen';

</script>

**groups.tpl**

<table cellspacing="0" cellpadding="7" border="0" width="720">

<tbody><tr>

<td valign="top" align="center">

<table cellspacing="1" bordercolordark="#ffffff" cellpadding="3" bordercolorlight="#cccccc" border="0" width="100%">

<tr>

<th width="10%" class="tableHeader">{$group\_id}</th>

<th class="tableHeader">{$group\_name}</th>

<th width="12%" class="tableHeader">{$edit\_group}</th>

<th width="12%" class="tableHeader">{$delete\_group}</th>

</tr>

{foreach from=$programers item=row}

<tr bgcolor="{cycle values="#f5f5f5,#dedee5"}">

{foreach from=$row item=value}

<td>{$value}</td>

{/foreach}

<td><a href = './group\_edit.php?id={$row.id}'> {$edit\_group} </a></td>

<td><a href = './group\_delete.php?id={$row.id}' onClick='return confirm("{$confirm\_delete\_group}");'> {$delete\_group} </a></td>

</tr>

{/foreach}

</table>

<table width="100%">

<tbody><tr>

<td align="center" height="50">

<br>

<b>{$add\_group}:</b>

<br>

<form method="post" action="groups.php">

<label for = "name"> {$group\_field\_name}: </label>

<input type="text" name="name" id="name">

<br>

{$error\_text}

<br>

<input type="submit" value="{$button\_add\_group}">

<br>

<br>

<br>

<input type="button" value="{$add\_group\_users}" onClick='location.href= "./group\_add\_user.php"'>

<input type="button" value="{$add\_group\_managers}" onClick='location.href= "./group\_add\_manager.php"'>

</form>

</td>

</tr>

</tbody>

</table>

</td>

</tr>

</tbody>

</table>

**group\_edit.tpl**

<form method="post" action="group\_edit.php?id={$group\_id}">

<label for = "name"> Назва групи: </label>

<input type="text" name="name" id="name" value="{$group\_info.name}">

{$error\_text}

<input type="submit" value="SAVE">

<br>

<br>

<input type="button" value="back" onClick='location.href="./groups.php"'>

</form>

**group\_add\_user.tpl**

{literal}

<style>

table.list tr td.name {border: 1px solid #b5b5b7; background: #f5f5f5;}

table.list tr:nth-child(even) td.info {background: #f5f5f5;}

table.list tr:nth-child(odd) td.info {background: #dedee5;}

</style>

{/literal}

<form method="get" action="./group\_delete\_user.php">

<table cellspacing="0" cellpadding="7" border="0" width="720">

<tbody><tr>

<td valign="top" align="center">

<table class="list" cellspacing="1" bordercolordark="#ffffff" cellpadding="3" bordercolorlight="#cccccc" border="0" width="100%">

<tr>

<th class="tableHeader">Group</th>

<th class="tableHeader">User</th>

<th class="tableHeader">Delete</th>

</tr>

{$group\_members}

</table>

<br>

<input type="submit" value="DELETE" onClick='return confirm("Підтвердіть видалення користувачів");'>

</form>

<br>

<br>

<b>{$group\_users\_add}</b>

<br>

<br>

<form method="post" action="">

<label for = "group"> {$groups\_list}: </label>

<select name="group" id="group">

{foreach from=$groups item=group}

<option value={$group.id}>{$group.name}</option>

{/foreach}

</select>

<label for = "user"> {$groups\_users\_list}: </label>

<select name="user" id="user">

{foreach from=$users item=user}

<option value={$user.u\_id}> {$user.u\_name} </option>

{/foreach}

</select>

<input type="submit" value="ADD">

<br>

<br>

<input type="button" value="back" onClick='location.href="./groups.php"'>

</form>