МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій і систем

Кафедра програмного забезпечення автоматизованих систем

**Пояснювальна записка**

до кваліфікаційної бакалаврської роботи

за напрямом підготовки 6.050103 «Програмна інженерія»

(освітньо-кваліфікаційний рівень)

на тему: «WEB орієнтований додаток футбольного трансферного ринку»

Виконав: студент 4 курсу,

групи ПЗ – 154

Напрям підготовки:

6.050103 «Програмна інженерія»

(шифр і назва напряму підготовки) Освітньо-кваліфікаційна програма

Інженерія програмного забезпечення

Студент \_\_\_\_\_Бондаренко А.В.\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_\_\_\_\_\_Катаєв Д.С.\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Рецензент \_\_\_\_\_\_Зеленько І.М.\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Черкаси, 2019 року

# **СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ**

|  |  |  |
| --- | --- | --- |
| **WEB** | — | система доступу до пов'язаних між собою документів на різних комп'ютерах, підключених до Інтернету |
| **HTML** | — | HyperText Markup Language — Мова розмітки гіпертекстових документів |
| **CSS** | — | Cascading Style Sheets (Каскадні таблиці стилів) |
| **БД** | — | база даних |
| **XML** | — | eXtensible Markup Language (розширювана мова розмітки) |
| **URL** | — | Uniform Resource Locator - єдиний вказівник на ресурс |
| **FTP** | — | File Transfer Protocol (Протокол передачі файлів) |
| **CMS**  **xG** | —  — | Content Management System (Система керування вмістом)  («expected goals») - це показник очікуваних голів. |
| **СКБД** | — | Система керування базами даних |

# **ВСТУП**

Футбол - це бізнес глобального рівня. Девід Бекхем, наприклад, на різних етапах своєї кар'єри грав за клуби Великої Британії, Іспанії, США, Італії та Франції. Футбольний трансфер — перехід гравця з одного футбольного клубу до іншого згідно з умовами, обумовленими у контракті (договорі) між цими клубами.

Професійні гравці підписують контракти з клубами на строк до п'яти років. Якщо футболіст переходить в інший клуб до закінчення контракту, новий клуб має виплатити попередньому компенсацію. При цьому футболіст підписує свій контракт, у якому вказаний строк, а також зарплата, преміальні, бонуси, права та обов'язки сторін. Клуб, якому належать права на футболіста, має право на отримання компенсації (зазвичай — грошової) за футболіста. Вартість гравця узгоджують представники клубів — учасників угоди. Окрім грошей інколи за гравця віддають ще одного футболіста як частину плати. Після завершення контракту футболіст має право перейти до іншого клубу без оплати трансферу останнім попередньому клубу.

Футбольні трансфери відіграють важливу роль у керуванні бюджетом клубу, формуванні складу, а також корисні для популяризації та реклами клубу. У випадку вдалої трансферної політики рівень гри та популярності клубу підвищуватиметься, у випадку невдалої — знижуватиметься.

Мета дипломного проекту – розробити WEB додаток, щоб з його допомогою можна було відслідковувати усі трансфери, які відбулися за останній трансферний період та детальну інформацію про них. Таким чином, скоротити час футбольних вболівальників на пошук інформації про той, чи інший трансфер на спортивних сайтах.

Створення такого WEB додатку доволі таки актуально, оскільки тема футболу зараз у світі дуже популярна. Футбольні матчі збирають не лише багатотисячні стадіони, а й мільйони глядачів перед телевізором. Але в той же час не у всіх вистачає часу на те, щоб слідкувати за трансферною активністю тої чи іншої команди. Тому дуже зручно було б переглядати трансферні придбання улюбленої команди чи будь якої іншої, маючи доступ до WEB додатку, на будь-якому девайсі.

**1 АКТУАЛЬНІСТЬ РОЗРОБКИ WEB ДОДАТКУ**

## **1.1 Опис предметної області розробленої системи**

З розвитком інформаційних технологій, широким розповсюдженням швидкісного інтернету і маркетингу в цілому багатьох речей, в тому числі і футболу, стало приділятися дуже багато уваги. Його можна легко дивитися в найвищій якості, розглядати до найдрібніших подробиць помилки агротехніків по вирощуванню газону, його можна читати зрозуміло, доступно і блищати знаннями в області xG і packing, його можна обговорювати на будь-який інтернет-платформі і формувати своє футбольне співтовариство. Все це дуже цікаво. Ну і фахівці в області піару дуже багато зробили для того, щоб популяризувати цей вид спорту в усьому світі. Взяти хоча б для прикладу футбольний клуб з Італії - Туринський Ювентус. Його недавній ребрендинг ознаменував собою нову віху в історії футболу. Тепер футбол - це ще й модно. Поло або футболки з принтом емблеми Ювентуса легко розійдуться не тільки серед уболівальників, але і рядових глядачів, тому що це вже бренд, і дуже стильний бренд. Тому на даний момент футбол визнаний найпопулярнішим видом спорту[2].

Розглянуті дані лише підтверджують очевидне: світовий трансферний ринок залишається долею клубів з провідних європейських асоціацій, які борються один з одним за молодих талантів і вже сформованих топових гравців. У цій сфері панує Англія, чиї багатющі клуби радше воліють купити готових футболістів, ніж вирощувати свою молодь (приємним винятком, мабуть, буде лише «Тоттенхем»). Основна ж проблема полягає в іншому: англійські, італійські і в меншій мірі іспанські клуби витрачають занадто багато грошей на придбання гравців. Тому останніми роками дуже складно слідкувати за трансферною активністю клубів.

**1.2 Дослідження програмних аналогів**

Багато з нас полюбляють футбол як вид спорту. І завжди хочеться бути в курсі всіх новин, в тому числі і трансферних, але не завжди вистачає часу, щоб моніторити перехід того чи іншого футболіста. Така «лінь» користувачів посприяла появі в мережі Інтернет ресурсів, що пропонують величезні бази даних трансферних переходів з одного клубу в інший, не лише цього сезону, а й попередніх. Найприємніше, що найчастіше дані ресурси дозволяють вільне та безкоштовне користування, а власники ресурсів отримують прибуток, наприклад, від реклами на сайті. В результаті всі у виграші. Інша справа, що останнім часом складно знайти такий ресурс. Щоб задовольняти власні потреби, в основному, ми все більше часу витрачаємо на пошук трансферних новин на футбольних сайтах, де складно відрізнити чутки від правдивих новин.

Але виділимо найпопулярніші з них:

**Footballhd.ru**

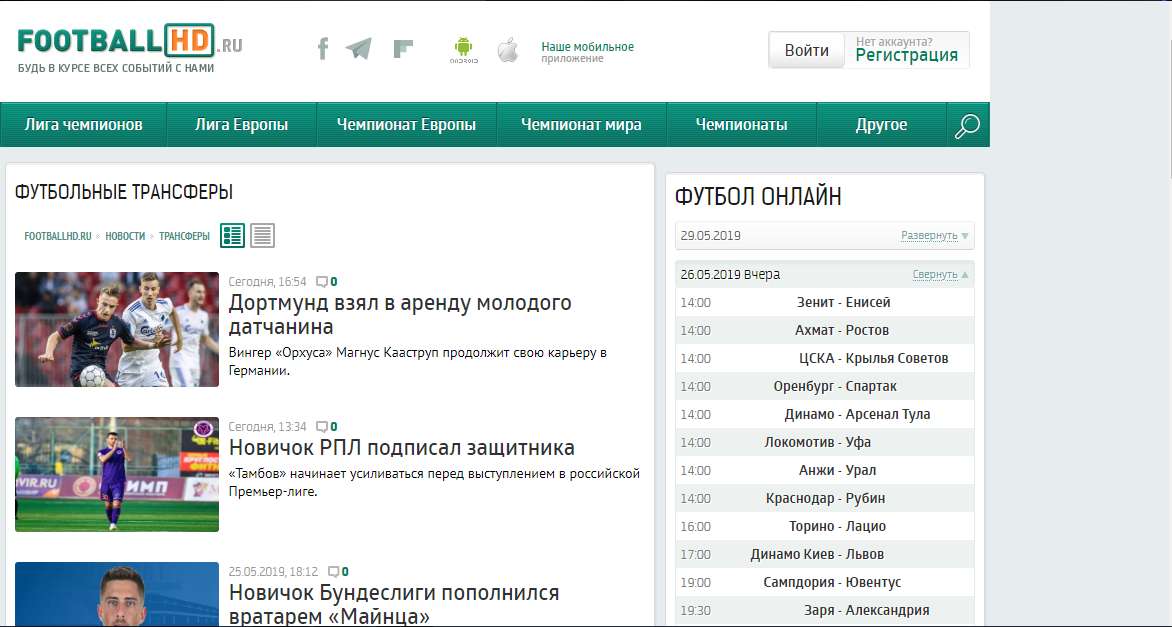


Рисунок 1.1– Головна сторінка Footballhd.ru.

Цей проект стартував у 2011 році[1]. Спочатку він позиціонувався як портал для перегляду онлайн трансляцій матчів. Але пізніше перекваліфікувався на портал трансферних новин та чуток. На даний момент також йде зміна формату. Розробники роблять більший акцент на новини футболу, а для трансферних новин залишили лише невеликий розділ(рис 1.1).

Переваги:

* наявність мобільної версії;
* безкоштовне користування;
* швидкодія;
* відсутність реклами.

Недоліки:

* + немає таблиці куплені/продані гравці;
  + відсутній фільтр сортування трансферних новин по лігах.

**Readfootball.com** 

Рисунок 1.2 – Головна сторінка readfootball.com.

Більшість сервісів дуже схожі один на одного, однак цей відрізняється в кращу сторону. Тут є можливість сортування трансферних новин по лігах та по датах. Крім того є корисна функція «Трансфери онлайн», де можна знайти свіжі, а головне перевірені трансферні новини(рис 1.2).

Розробники наділили ресурс особливою ​​фішкою – тут можна знайти інформацію не лише про ту чи іншу лігу або команду, тут також можна спостерігати за активністю гравця чи навіть тренера. Таким способом можна знайти конкретного гравця, переглянути його трансферну активність, результативність на футбольному полі, трофеї, а також порівняти його з іншими гравцями і багато всього іншого(рис. 1.3). Саме головне, що інтерфейс зручний, а сайт не перенасичений зайвою інформацію[3].

Переваги:

* зручний інтерфейс;
* різноманіття додаткової інформації;
* велика кількість фільтрів для сортування.

|  |
| --- |
|  |

Рисунок 1.3 – Статистика конкретного гравця

Недоліки:

* нестабільна робота;
* відсутність мобільної версії;
* неможливо побачити трансфери окремих клубів.

**Il-calcio.ru**

Хоча ресурс є досить таки непопулярний, має дуже скромний інтерфейс та мале різноманіття інформації, але сайт висвітлює події, які пов'язані з усіма футбольними клубами Серії А, тобто чемпіонату Італії(рис.1.4). Щодня моніторяться італійські ресурси, щоб знайти новини про кожен з клубів[4]. Тут є вичерпні відомості про все, що пов'язано з чемпіонатом Італії по футболу. Тут можна знайти таблицю Серії А , календар і розклад ігор , трансфери гравців і таблицю бомбардирів. Але найголовніше, нащо варто звернути увагу, це формат подачі інформації про трансфери (рис. 1.5).

Переваги:

* + зручна подача інформації;
  + наявність новин та додаткової інформації.

Недоліки:

* невелике різноманіття інформації;
* слабкий інтерфейс;
* відсутня підтримка для телефонів і планшетів.

|  |
| --- |
|  |

Рисунок 1.4 – Головна сторінка Il-calcio.ru.

|  |
| --- |
|  |

Рисунок 1.5 – Формат подачі інформації про трансфери

**UA-Футбол**

Старт проекту був даний більш десяти років тому. З тих пір UA-Футбол впевнено розвивається, знаходячи все нових і нових шанувальників. За роки роботи він став одним з найавторитетніших і читаємих футбольних ресурсів не тільки в Україні, а й інших країнах. Ось уже багато років UA-Футбол є оперативним і об'єктивним джерелом інформації про український та зарубіжний футбол. Особлива увага приділяється авторським матеріалам (звіти, інтерв'ю, аналітичні розбори, переклади та багато іншого), які давно стали візитною карткою сайту, вигідно відрізняючи UA-Футбол від конкурентів[5].

Також, велика увага приділяється мультимедіа. На UA-Футбол завжди можна знайти фоторепортажі з матчів, відеозвіти про ігри та багато іншого. Крім того, велика частину порталу приділена освітленню трансферним новинам та безпосередньо самим трансферам(рис. 1.6).

|  |
| --- |
|  |

Рисунок 1.6 – Розділ трансферів на порталі UA-Футбол.

Переваги:

* зручний інтерфейс;
* наявність додаткової інформації про трансфери;
* наявність фільтрів для сортування трансферів;
* мінімальна кількість реклами.

Недоліки:

* ресурс дуже часто не працює через проблеми із сервером;
* дрібні трансфери не відображаються;
* відсутня підтримка мобільних пристроїв та планшетів.

**Transfermarkt**

Transfermarkt – німецький WEB додаток футбольної тематики, який містить інформацію про футбол, рахунок матчів, результати турнірів і трансферні новини і

|  |
| --- |
|  |

Рисунок 1.7 – Головна сторінка порталу Transfermarkt.

багато чого іншого(рис. 1.7). За версією IVW, він входить в топ-25 найбільш відвідуваних німецьких сайтів і є одним з найбільших спортивних сайтів. Сайт був створений в травні 2000 року. Сайт містить інформацію про рахунок, результати матчів і турів(рис. 1.8), трансферні новини(рис. 1.9) і вартості гравців(рис. 1.10). Незважаючи на те, що вартість гравців, поряд з деякими іншими показниками, є оціночними, дослідники з Центру вивчення економічних показників виявили, що «чутки» про трансфери і вартості гравців в значній мірі точні. 2014 року відбувся перезапуск сайту, було введено так зване оновлення версії 4.0. В ході цього оновлення було вирішене ряд технічних, а також правових питань, оскільки приватні дані протягом невизначеного періоду часу було видно іншим користувачам. Найбільше критики користувачів було пов'язано зі складним новим дизайном. З одного боку це один з найавторитетніших ресурсів футбольних СМІ, а з іншого боку це яскравий приклад переповнення ресурсу великою кількістю зайвої інформації, яка ускладнює перегляд для користувача та дає велике навантаження на портал[6].

Також варто згадати такі ресурси як Sport.ua та Tribuna.com, які в кінці кожного міжсезоння роблять таблицю трансферних придбань та продажів клубів Української Прем’єр Ліги.

|  |
| --- |
|  |

Рисунок 1.8 – Результати матчів і турів.

|  |
| --- |
|  |

Рисунок 1.9 – Трансферні новини.

|  |
| --- |
|  |

Рисунок 1.10 – Трансферна вартість гравців.

**1.3 Постановка задач розробки**

Основним завданням розробки WEB додатку вважаємо:

* провести аналіз та систематизацію характеристик різноманітних WEB додатків футбольної трансферної тематики;
* визначити систему критеріїв для проведення порівняльного оцінювання основних характеристик і експлуатаційних показників WEB додатків футбольної трансферної тематики;
* визначити принципи реалізації динамічних WEB додатків;
* розробити структуру Front-end частини WEB додатку футбольного трансферного ринку;
* провести варіантний аналіз програмного забезпечення з метою вибору засобів реалізації WEB додатку;
* розробити засоби реєстрації і авторизації користувачів у WEB додатку;
* протестувати розроблений програмний продукт.

**Висновки**

В даному розділі було проведено детальний аналіз предметної області, визначено суть технічної проблеми, актуальність питання, проаналізовано сучасний стан та перспективи розвитку WEB додатків спрямованих на освітлення футбольних трансферів. Також було здійснено огляд існуючих сайтів на дану тематику, визначені їх переваги і недоліки, які будуть враховані при розробці WEB додатку футбольного трансферного ринку.

У заключному пункті даного розділу було здійснено постановку задач розробки.

## **2 ФУНКЦІОНАЛЬНІ ВИМОГИ ТА ОПИС СЕРЕДОВИЩА РОЗРОБКИ**

**2.1 Функціональні вимоги до системи**

Основним етапом розробки був вибір зручного середовища розробки, а також аналіз програмних засобів реалізації та систем керування базами даних. Обравши найбільш придатний варіант для реалізації поставлених завдань, потрібно було реалізувати основні модулі WEB додатку.

Оскільки програмний продукт є WEB додатком, не обов’язково мати конкретну оперативну систему. Для того, щоб людина могла користуватися додатком, на її девайсі повинен бути встановлений браузер (WEB оглядач)[7].

Найпоширеніші браузери:

* Google Chrome;
* Internet Explorer;
* Opera;
* Firefox;
* Safari.

Це дозволить гарантувати якісний досвід роботи із WEB додатком для більшості користувачів. Розробка програмного продукту спеціально під вище перечисленні браузери дозволить врахувати всі особливості середовища та гарантувати якість роботи продукту.

Технічні вимоги до продукту наступні:

* програмний продукт повинен функціонувати на персональних
* комп’ютерах із стандартним набором компонентів;
* забезпечувати цілодобовий доступ клієнтів до інформації WEB додатку;
* забезпечувати зручність і простоту взаємодії з користувачем;
* передбачати мінімальні витрати на впровадження програмного продукту.

**2.2 Середовище розробки**

На сьогодні існує величезна кількість систем створення WEB додатків, які підрозділяють на три різні категорії:

* текстові редактори HTML-коду (Alleycode HTML Editor, Aptana, Arachnophilia, BBEdit);
* редактори WYSIWYG («What You See Is What You Get» або «що ви бачите, то ви і отримаєте» – досить потужні і зручні засоби створення і обробки WEB додатків: Adobe Dreamweaver, Adobe GoLive, Amaya, Blockstar, Bluevoda, HotDog, iWeb, Media Lab);
* текстові редактори (Emacs, gedit, Notepad, TextEdit, UltraEdit).

Для реалізації WEB додатку було обрано інтегроване середовище розробки JetBrains WebStorm, так як воно є найпопулярнішою системою розробки WEB додатків. JetBrains WebStorm — вільне інтегроване середовище розробки (IDE) для мов програмування Java, JavaFX, C/C++, PHP, JavaScript, HTML5, Python, Groovy. Середовище може бути встановлене і для підтримки окремих мов, і у повній конфігурації.

Середовище розробки JetBrains WebStorm за умовчанням підтримує розробку для платформ J2SE і J2EE. JetBrains WebStorm доступна для платформ Microsoft Windows, GNU/Linux, FreeBSD, і Solaris (як SPARC, так x86). За якістю і можливостям останні версії JetBrains WebStorm змагається з найкращим інтегрованими середовищами розробки для мови Java Script, підтримуючи рефакторинг, профілювання, виділення синтаксичних конструкцій кольором, авто доповнення мовних конструкцій та шаблони коду[8].

**2.3 Обґрунтування вибору програмних засобів для реалізації WEB додатку**

Ще одним важливим підготовчим етапом є вибір мови програмування для розробки WEB додатку.[9] Для створення WEB додатків використовуються різноманітні технології і мови програмування, оскільки кожна з мов реалізує свою можливісті, що викликані способом реалізації програм на даній мові. Кожна з них має свої переваги і недоліки, тому вибір мови повинен залежати від поставленої задачі і в результаті прорахування навантаження. Найпопулярнішими мовами для задач подібного типу є:

* ASP;
* ASP.NET;
* JS;
* Perl;
* PHP.

**2.4 Аналіз можливостей JavaScript**

JavaScript – назва реалізації стандарту мови програмування ECMAScript компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови – написання сценаріїв для WEB сторінок, але, також, використовується для впровадження сценаріїв керування об'єктами вбудованими в інші програми[10].

Незважаючи на схожість назв, JavaScript та мова програмування Java мають дуже мало спільного. Права на назву JavaScript належать компанії Sun Microsystems.

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 рівня, об'єкти як списки, каррінг (currying), анонімні функції, замикання (closures), що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою С 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 та операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто в кінцевому рахунку 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 .

Отже, проаналізувавши WEB технології програмування було вирішено використовувати мову програмування Java Script для написання сценаріїв.

**2.5 Аналіз систем управління базами даних**

Робота з базами даних є одною з найважливіших складових програмування

WEB додатку динамічного типу. Бази даних для WEB додатку використаємо з метою зберігання різної інформації. Спрощено бази даних представляють собою певний набір взаємозалежних таблиць. Розміри таблиць в базах даних різні, їх кількість довільна. Саме в базах даних буде зберігатися на сервері необхідна для роботи сайту інформація, наприклад, інформація про клієнтів, каталог товарів, статистичні дані тощо.

До найпопулярніших баз даних, що використовують при програмуванні WEB додатків відносять SQL, MySQL, Oracle Database, PostgreSQL та ін. Для того, щоб обрати СУБД, найбільш підходящу для розробки WEB додатку, проаналізуємо переваги та недоліки кожної з них[11].

Для побудови запитів до баз даних широко застосовується SQL (Structured Query Language) ― мова структурованих запитів. За допомогою SQL може здійснюватися додавання, видалення, редагування записів в таблицях баз даних, вибірка даних у відповідності з різними умовами, сортування даних і багато іншого.

Основними перевагами SQL є:

* незалежність від конкретної СУБД. Незважаючи на наявність діалектів і відмінностей в синтаксисі, в більшості своїй тексти SQL-запитів, що містять, DDL і DML, можуть бути досить легко перенесені з однієї СУБД в іншу. Існують системи, розробники яких спочатку закладалися на застосування щонайменше кількох СУБД;
* наявність стандартів. Наявність стандартів і набору тестів для виявлення сумісності і відповідності конкретній реалізації SQL загальноприйнятому стандарту, що тільки сприяє «стабілізації» мови. Правда, варто звернути увагу, що сам по собі стандарт місцями занадто формалізований і роздутий в розмірах;
* декларативність. За допомогою SQL програміст описує тільки те, які дані потрібно витягнути або модифікувати. Те, яким чином це зробити, вирішує СУБД безпосередньо при обробці SQL-запиту. Проте не варто думати, що це повністю універсальний принцип – програміст описує набір даних для вибірки або модифікації, проте йому при цьому корисно уявляти, як СУБД розбиратиме текст його запиту. Особливо критичні такі моменти стають при роботі з великими базами даних і зі складними запитами – чим складніше сконструйований запит, тим більше він допускає варіантів написання, різних за швидкістю виконання, але тих самих за набором даних.

SQL має і недоліки:

* невідповідність реляційної моделі даних;
* складність;
* відступи від стандартів;
* складність роботи з ієрархічними структурами.

**2.6 Обґрунтування вибору систем управління базами даних**

Отже, для того щоб обрати СКБД, яка б як найкраще підходила саме для виконання задач при розробці WEB додатку програмного забезпечення, визначимо, які вимоги має задовольняти обрана СКБД. Система керування базою даних має відповідати таким вимогам:

* надійність;
* швидкодія;
* простота;
* зручність використання.

Проаналізувавши переваги та недоліки СКБД можна зробити висновок, що мова структурованих запитів SQL підходить для реалізації поставлених завдань, так як вона відповідає реляційній моделі даних та є відносно не складною у використанні.

Отже для реалізації WEB додатку було обрано скриптову мову програмування

Java Script та систему керування базами даних MySQL. MySQL складається з двох частин: серверної і клієнтської.

# **2.7 Система розробки інтерфейсу засобами Dreamweaver**

HTML (англ. HyperText Markup Language — мова розмітки гіпертекстових документів) — стандартна мова розмітки WEB сторінок в Інтернеті. Більшість WEB сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

У більшості випадків автор документа суворо визначає зовнішній вигляд документа. У разі HTML читач, ґрунтуючись на можливостях WEB браузера, може, певною мірою, керувати зовнішнім виглядом документа (але не його вмістом). HTML дозволяє відзначити, де в документі повинен бути заголовок або абзац за допомогою тега HTML, а потім надає WEB браузеру інтерпретувати ці теги. Наприклад, один WEB браузер може розпізнавати тег початку абзацу і представляти документ у потрібному вигляді, а інший не має такої можливості і 13 надає документ в один рядок. Користувачі деяких WEB браузерів мають, також, можливість налаштовувати розмір і вид шрифту, колір та інші параметри, що впливають на відображення документа[12].

HTML-теги можуть бути умовно розділені на дві категорії:

* теги, що визначають, як буде відображатися WEB браузером тіло документа в цілому.
* теги, що описують загальні властивості документа, такі як заголовок чи автор документа.

HTML-документи можуть бути створені за допомогою будь-якого текстового редактора або спеціалізованих HTML-редакторів і конвертерів. Вибір редактора, який буде використовуватися для створення HTML-документів, залежить виключно від поняття зручності і особистих пристрастей кожного автора.

Наприклад, HTML-редактори, такі, як "Netscape Navigator Gold" компанії Netscape, дозволяють створювати документи графічно з використанням технології WYSIWYG (What You See Is What You Get). З іншого боку, більшість традиційних засобів для створення документів мають конвертери, що дозволяють перетворювати документи до формату HTML. Основна перевага HTML полягає в тому, що ваш документ може бути переглянутий на WEB браузерах різних типів і на різних платформах.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Таблицю стилів CSS можна вмонтувати прямо в HTML сторінку – це внутрішня таблиця стилів. Або ж її можна створити в окремому файлі, і вже потім приєднати посилання на нього до потрібної HTML сторінки – це зовнішня таблиця стилів.

Зовнішню таблицю необхідно підключити до основного HTML документу за допомогою спеціальних тегів: < link rel="stylesheet" type="text/css" href="/style.css >,

style.css – це ім'я файлу, що містить таблицю CSS. Завдяки цьому, стиль, описаний у зовнішній таблиці CSS, можна використовувати повторно скільки завгодно разів.

Роблячи висновки з попередньої інформації можна зробити висновки, що використання HTML та CSS є зручним та не складним у використанні, але має ряд недоліків таких як:

* обмежені можливості при створені сайту ;
* відсутність динамічних елементів;
* велика кількість коду.

На сьогодні найпопулярнішою системою розробки WEB сайтів являється Dreamweaver — HTML-редактор. Розроблений і підтримувався компанією Macromedia у версіях до MX (6 версія, 2005 рік), а нині – Adobe. Багатий інструментарій, відвертість додатку для всіляких настройок, зручний інтерфейс і інші особливості зробили Dreamweaver одним з найпопулярніших HTML-редакторів в світі. Для деяких версій Dreamweaver виконувалися переклади довідкових систем на російську мову.

Dreamweaver включає багато інструментів і засобів для редагування і створення професійного сайту: HTML, CSS, JavaScript, відладчик JavaScript, редактори коду (перегляд коду і інспектор коду), що дозволяє вам, редагувати JavaScript, XML і інші текстові документи, які підтримуються в Dreamweaver.

Технологія Roundtrip HTML імпортує документи HTML без потреби переформатовувати код і ви можете налаштувати Dreamweaver, щоб він наводив лад і переформатовував HTML як ви цього хочете.

Можливість візуального редагування в Dreamweaver також дозволяє вам швидко створювати або міняти дизайн проекту без написання коду. Ви можете розглянути всі ваші централізовані елементи і перетягнути їх із зручної панелі безпосередньо в документ.

# **2.8 Використані бібліотеки Java Script**

Leaflet — JavaScript бібліотека з відкритим сирцевим кодом для відображення мап на HTML-сторінках. Самодостатня, сучасна, не велика за обсягом, з широким колом ліцензійної угоди, проста у використані.

Бібліотека реалізує підтримку шарів мап, які побудовані за технологією: WMS, GeoJSON, Тайли або векторного відображення поверхні. Інші типи проекцій мап підтримуються за допомогою додатків. Головне, щоб кожний формат мав однозначну функцію перетворення локальних координат в географічні координати в якої ці локальні координати завдані.

Була започаткована в 2011 році. Працює на більшості мобільних та десктопних платформ, які підтримують стандарти HTML5 та CSS3.

Leaflet не потребує від програміста досвіду роботу за картографічними сервісами, значно спрощуючи задачу вбудови мапи на HTML-сторінки або WEB додатки. Бібліотека дозволяє працювати із різними шарами, в якості джерел мапи використовувати будь-якій публічний WEB сервіс тайлів (порізані зображення мапи). Завантажувати дані та накладати дані з GeoJSON файлів, змінювати стилі, додавати інтерактивні маркери[13].

CartoDB – платформа для створення користувацьких динамічних карт, яка дозволяє користувачам наносити на карти, аналізувати і створювати додатки з використанням даних. Платформа CartoDB була створена для того, щоб спростити і прискорити роботу з великими обсягами геопросторових даних. CartoDB спочатку була призначена для розробників і творців мобільних додатків, але ця платформа також підходить для людей, у яких є геопросторові набори даних, які хочуть створити їх візуалізацію та отримати можливість реального глибокого розуміння цих даних[14].

CartoDB дозволяє перетягувати дані або завантажувати набір даних, створювати сучасний дизайн, використовуючи CartoCSS, виконувати SQL-запити даних і вставляти карту у WEB додаток або в блог на WordPress[15].

|  |
| --- |
|  |

Рисунок 2.1 – приклад використання CartoDB та Leaflet.

**3 РОЗРОБКА ІНТЕРФЕЙСУ КОРИСТУВАЧА**

**3.1 Структура макету WEB додатку**

В основному будь-який WEB додаток є сукупністю динамічних або статичних сторінок, які містять тексти, картинки, мультимедійні чи інші об’єкти. Для комфортного перебування відвідувача і для полегшення пошуку потрібної інформації, він повинен мати чітку і продуману структуру.

WEB додаток – інформаційний ресурс, доступний у мережі World Wide Web, який можна переглянути у браузері. Зазвичай ця інформація записана у форматі HTML або XHTML і може містити гіпертекст із навігаційними гіперпосиланнями на інші WEB сторінки. WEB додаток – сукупність WEB сторінок, доступних в Інтернеті, які об’єднані як за змістом, так і навігаційно.

Структура WEB додатку – це внутрішній устрій, його «кістяк», розташування

сторінок, розділів, підрозділів, додаткових матеріалів. І першочерговим завданням розробника є створення строгого порядку з хаотичного скупчення інформації.

Створення WEB додатку починається зі створення інформаційної моделі сайту. Будь-яку WEB сторінку можна оцінити за двома параметрами: зміст та зовнішній вигляд. Зовнішній вигляд кожного додатку є унікальним.

Загалом виділяють три типи структур WEB додатків – лінійну, деревоподібну та довільну.

Додаткові структури:

* стандартна;
* каскадна;
* хмарочос;
* павутина.

Створення WEB додатку умовно можна розділити на такі етапи:

* попередній етап розробки додатку. На цьому етапі розв’язуються питання
* загального характеру. Обговорюється загальна концепція, формулюються та фіксуються цілі створення додатку;
* етап проектування додатку. Визначення структури додатку: меню, посилання, розміщення модулів, побудова списку компонентів, що підключаються, тощо.
* етап розробки й тестування додатку;
* розміщення додатку на хостинг;
* розвиток ресурсу.

Вдалий дизайн в першу чергу має орієнтуватися на користувача, який без особливих зусиль може знайти і використати будь яку інформацію, що там міститься. При цьому повинні бути абсолютно зрозумілими логічна схема і логічний взаємозв'язок між окремими сторінками.

Хороша навігація означає, що користувачі точно знають, де знаходяться, де розташовані елементи додатку і як використовувати ці елементи.

Правильна структура інформації дозволяє користувачам без побоювання продовжувати ознайомлення, і бути впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

У вдало організованих WEB додатках відвідувачі можуть вільно переміщатися у пошуках потрібної інформації, не турбуючись про структуру. Піклування про структур є завданням розробника, а не відвідувачів. Логічне проектування передбачає організацію інформації, побудову його структури та навігацію по розділах.

Організація даних у вигляді впорядкованої структури повинна повідомляти відвідувача, яку інформацію він може знайти на, і де її шукати. Інформація повинна бути організована так, щоб відвідувач знав, що його чекає на наступній сторінці вже за назвою посилання, по якій він переходить.

Перед тим, як розділяти офіційну сторінку на окремі сторінки, необхідно розробити план. WEB додаток проектується таким чином, що потрібно добре продумати його загальну структуру, зміст інформації та посилання.

Нижче наведено чотири варіанти структури. Кожна з них має свої недоліки та переваги при проектуванні WEB додатку. Вибір конкретної структури визначається особливостями завдань, що вирішуються за допомогою WEB додатку[16].

Стандартна. Основна WEB сторінка містить посилання на інші документи, а документи містять посилання, відповідно, на основну WEB сторінку. Це найпростіший і найчастотніший спосіб організації. Таке подання інформації з однієї сторони полегшує процес взаємодії з відвідувачами, з іншої сторони подібна організація структури не залишає відвідувачам волі вибору, що може їм не сподобатися. Тому на практиці лінійна структура застосовується з деякими видозмінами. На рисунку 3.1.1 показана схема стандартного способу розділення Web-сторінки.

|  |
| --- |
|  |

Рисунок 3.1 – Стандартний спосіб організації WEB сторінок

Каскад. У цьому випадку посилання у документах задані таким чином, що існує тільки один шлях обходу сторінок WEB додатку. На рисунку 3.1.2 нижче показаний приклад каскадного сайту.

|  |
| --- |
| https://www.vuzlib.su/posibn_podatk/52.htm41.gif |

Рисунок 3.2 – Каскадний спосіб організації WEB сторінок

При каскадному способі організації сторінок відвідувачі можуть переміщуватись тільки в одному з напрямків, вперед або назад.

Хмарочос. У даній моделі відвідувачі можуть опинитися на деяких сторінках, тільки якщо вони йдуть вірним шляхом. Це нагадує підйом до потрібної кімнати у великому хмарочосі. Схема, яка наведена на малюнку нижче, дійсно нагадує план хмарочоса(рис.3.3)

|  |
| --- |
| https://www.vuzlib.su/posibn_podatk/52.htm42.gif |

Рисунок 3.3 – Структура сайту «хмарочос»

Павутина. У цьому випадку усі сторінки WEB додатку містять посилання на інші сторінки, і користувач може легко перейти з будь-якої сторінки практично на будь-яку іншу. Ця схема може перетворитися у лабіринт, якщо вийде з-під контролю, але вона популярна у тих випадках, коли посиланнями на документи користуються не надто часто. На рисунку 3.4 нижче представлений доволі простий приклад такої організації.

|  |
| --- |
|  |

Рисунок 3.4 – Структура WEB додаток «павутина»

При організації WEB додатку рекомендується користуватися гібридним методом, який представляє суміш стандартного методу і методу павутини. Ми можемо опинитись у будь-якому документі безпосередньо з основної офіційної сторінки, при цьому і самі документи також посилаються один на інший. Використання методу «хмарочосу» небажане тому, що воно дезорганізує навігацію користувача.

Інформацію необхідно подавати українською мовою, а певні розділи – дублювати іншими мовами, згідно з особливостями регіону (російською, англійською, угорською). Для цього головна сторінка має мати можливість переключення між цими мовами. Меню переключення мов складається із графічних зображень (у форматі GIF), а не з текстових надписів. Це дозволяє здійснювати вибір мови у середовищах, де не підтримується потрібне кодування літер, зокрема українських, наприклад, користувачами країн Європи або Америки.

Обсяги надання інформації в українському та іншомовному варіантах WEB сторінок можуть не співпадати тому, що WEB додаток призначений одночасно для різних верств користувачів в Україні і за кордоном.

Структура є першочерговим кроком у розробці. Вона представляє собою певний каркас, тобто базис, який згодом заповнюється графічним та інформаційним матеріалом.

Найбільш зручною та універсальною є блочна структура WEB сторінок, що передбачає розділення на логічні частини (блоки), до яких окремо застосовується певне графічне оформлення, яке відповідає загальному дизайну всього проекту. Вона є незалежною від різного програмного забезпечення, на якому в подальшому буде відображатися додаток, що надає їй значних переваг над іншими структурами, наприклад, табличною.

Логічну блочну структуру можна реалізувати у вигляді декількох моделей, а саме: лінійної, моделі «грати», «дерево» та «павутина». Використовуються також різні комбінації описаних варіантів.

Лінійна структура дозволяє відвідувачам переходити тільки від попередньої сторінки до наступної, ніяк інакше. Через таку жорсткість та відсутність вибору вона застосовується досить рідко, хоч і є легкою у реалізації. Модель «грати» має і горизонтальні, і вертикальні взаємозв’язки, тобто двонаправлену лінійну структуру, що дає можливість переглядати сторінки за обома критеріями. Структура «дерево» є ієрархічною, і дає можливість керувати глибиною відвідування, а простір вибору обмежується «шириною дерева». Ця структура є найбільш використовуваною та зручної, тому вона буде обрана для розробки WEB додатку. Щодо моделі «павутина», то така організація взаємодії WEB сторінок є заплутаною і важко зрозумілою, але надає користувачам весь обсяг інформації.

Головними елементами сторінки, на які потрапляє увага відвідувача, є система навігації та меню, назви категорій, гіперпосилання і структура інформації. Вдала навігація означає, що користувач точно знає, де знаходиться, елементи додатку і як використовувати ці елементи. Правильна структура інформації дозволяє вільно продовжувати дослідження додатку, залишаючись впевненими в тому, що завжди є можливість повернутися до раніше переглянутих сторінок. Значна зміна елементів навігації при переході від однієї WEB сторінки до іншої, відволікає увагу відвідувачів від вмісту, тому її не варто застосовувати. Меню потрібно робити зручним для використання та легко зрозумілим. Надмірне застосування ефектів, задля гарного вигляду, в даному випадку себе не виправдовує.

Розробка WEB додатка базується на описаних методиках, з використанням усіх технологій, принципів та основ побудови структур.

Проаналізувавши всі підходи та вибравши найбільш підходящі методи рішення задачі був вибраний найоптимальніший із варіантів побудови структури сайту – гібридний. Згідно з ними, структура WEB додатку з вказаними блоками та зв’язками зображена на Рисунку 3.5.

|  |
| --- |
|  |

Рисунок 3.5 – Структура WEB додатку

**3.2 Структура бази даних**

На рисунку 3.6 показана діаграма моделі.

|  |
| --- |
|  |

Рисунок 3.6– ER-діаграма таблиць бази даних

У відповідності до вибраної архітектури, необхідно реалізувати моделі, які будуть виступати у ролі посередника між скриптом і базою даних. База даних повинна відповідати усім поставленим вимогам і будуватись у відповідності до моделей. Побудуємо ER-діаграму таблиць бази даних, яка буде описувати структуру і відношення таблиці. Для побудови використаємо програму MySql Workbench, і після побудови моделі синхронізуємо її з базою даних.

Отже, після того, як створена база даних наповнимо її таблицями, які будуть містити інформацію зі сторінок WEB додатку.

На прикладі таблиць «Teams», «Players», «Match», «MatchResults» показано інформацію, яка буде знаходитись на динамічних WEB сторінках додатку.

Задамо назви полів таблиці «Teams» :

* + «ID», дане поле будемо використовувати як первинний ключ. Дане поле буде унікальним і буде зростати при додаванні записів;
  + «Name», поле зберігає назву команди;
  + «Country», поле, яке зберігає назву країни, яку представляє футбольний клуб;
  + «PlayerCount», поле, зберігає кількість футболістів в клубі;
  + «Logo», поле, яке зберігає логотип футбольного клубу;
  + «League\_ID», ідентифікатор ліги, яку представляє футбольний клуб;
  + «Staff\_ID», ідентифікатор обслуговуючого персоналу, який працює в системі футбольного клубу.

Задамо назви полів таблиці «Players» :

* + «ID», дане поле будемо використовувати як первинний ключ. Дане поле буде унікальним і буде зростати при додаванні записів;
  + «Name», поле зберігає ім’я гравця;
  + «MiddleName», поле зберігає по батькові гравця;
  + «Surname», поле, зберігає прізвище гравця;
  + «Birthday», поле, яке зберігає дату народження;
  + «Nationality», поле, яке зберігає національність гравця;
  + «Age», поле яке зберігає вік гравця;
  + «Photo», поле зберігає фото гравця.

Задамо назви полів таблиці «Players :

* + «ID», дане поле будемо використовувати як первинний ключ. Дане поле буде унікальним і буде зростати при додаванні записів;
  + «Name», поле зберігає назву матчу;
  + «MatchDate», поле зберігає дату мату;
  + «Country», поле, зберігає назву країни, в якій відбувся матч;
  + «Stadium», поле, яке зберігає назву стадіону.

Задамо назви полів таблиці «MatchResults»:

* + «ID», дане поле будемо використовувати як первинний ключ. Дане поле буде унікальним і буде зростати при додаванні записів;
  + «MatchDate», поле зберігає дату матчу;
  + «Value», поле, зберігає рахунок матчу, який відбувся;

**3.3 Інтерфейс користувача**

Користувацький інтерфейс забезпечує взаємодію між користувачем та WEB додатком. Це важлива складова програмного продукту, оскільки являється єдиним способом для користувачів взаємодіяти із системою[17].

Програмний продукт створюється для наступних користувачів:

* футбольні фанати;
* люди, які цікавляться лише результатами матчів;
* футбольні агенти;
* футбольні аналітики.

Описані вище користувачі мають різний досвід та навички використання WEB додатків. Саме тому інтерфейс повинен забезпечувати зручність користування системою для користувача із будь-яким рівнем комп’ютерної грамотності. Але разом із тим забезпечувати приємний вигляд та відповідати сучасним нормам та стандартам.

Графічні матеріали (фотографії та малюнки) створюються з обов’язковою оптимізацією для відображення за розміром і кількістю кольорів. Графічні матеріали повинні відображатись без спотворень.

Дизайн WEB додатку повинен відповідати усім сучасним нормам та напрямкам у WEB дизайні, бути простим й одночасно сучасним та інформативним. Дизайн необхідно виконати в спокійній кольоровій гамі. Дизайн повинен бути виконаний в стилі мінімалізму (нічого зайвого, відволікаючого увагу).

Інтерфейс користувача повинен забезпечувати інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації та швидкий і логічний перехід між розділами.

Навігаційні елементи повинні забезпечувати однозначне розуміння користувачами їх змісту: лінки на сторінки повинні містити заголовки, умовні позначення - відповідати загальноприйнятим. Графічні елементи навігації повинні містити альтернативний підпис.

Під час розробки було створено макет інтерфейсу користувача для головних компонентів.

Основне меню має надавати доступ користувачу до будь-якої частини WEB додатку для більш зручної навігації(рис3.7).

|  |
| --- |
|  |

Рисунок 3.7 – Основне меню WEB додатка

Додаткова навігаційна панель призначена для більш швидкого доступу до найбільш важливих сторінок WEB додатку(рис.3.8).

|  |
| --- |
|  |

Рисунок 3.8 – Макет додаткової навігаційної панелі

|  |
| --- |
|  |

Рисунок 3.9 – Макет головної сторінки з новинами

|  |
| --- |
|  |

Рисунок 3.10 – Макет компонента «Таблиця трансферів»

|  |
| --- |
|  |

Рисунок 3.10 – Макет компонента «Інформація про клуб»

|  |
| --- |
|  |

Рисунок 3.11 – Макет компонента «Контрольні матчі»

Відразу після реєстрації(рис.3.12), а потім авторизації(рис. 3.13), у користувачів з’являються додаткові можливості роботи у WEB додатку, а саме – коментувати будь-які новини та оцінювати коментарі інших користувачів(рис. 3.14). Всі компоненти були реалізовані з допомогою QuintaDB – онлайн конструктору WEB форм і баз даних[18].

|  |
| --- |
|  |

Рисунок 3.12 – Макет компонента «Реєстраційна форма»

|  |
| --- |
|  |

Рисунок 3.13 – Макет компонента «Авторизація»

|  |
| --- |
|  |

Рисунок 3.14 – Макет компонента «Коментар»

За допомогою бібліотек CartoDB та Leaflet було створено користувацьку динамічну карту, яка дозволяє користувачам побачити, який гравець і яким футбольним клубом був придбаний (рис. 3.15).

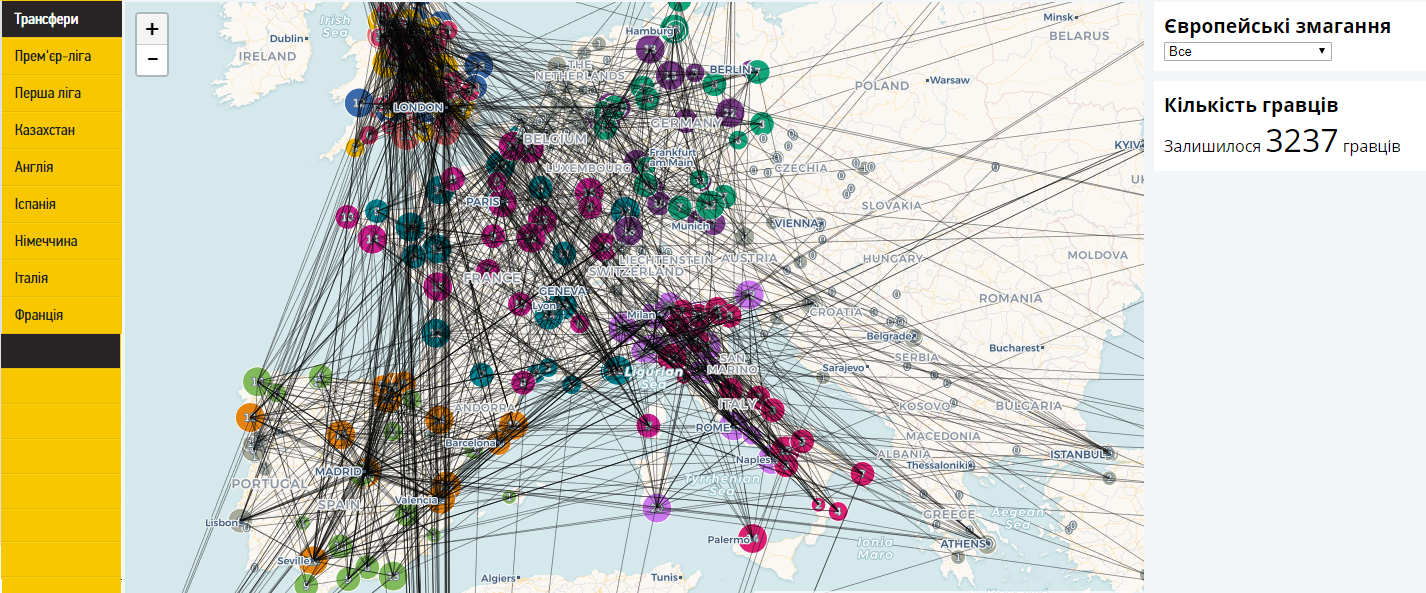


Рисунок 3.15 – макет компоненту «Трансфери»

**Висновки**

В даному розділі було проаналізовано принципи реалізації WEB додатків. Було визначено наступні структури: стандартна (лінійна), каскадна, хмарочосна, павутинна (комбінована). Для реалізації WEB додатку було обрано павутинну структуру. Також було визначено основні принципи, що будуть застосовуватись при проектуванні інформаційного наповнення WEB додатку і засобів навігації.

Було визначено типи інформаційного забезпечення та коротко охарактеризовано їх. Доцільним вважається розміщення загальної інформації про клуб, про історію клубу, досягнення, інформацію гравців, персоналу, а найголовніше футбольних трансферів.

Детально описаний і реалізований модуль новин, що забезпечує додавання, редагування та видалення новин, також він забезпечує виведення останніх новин на головній сторінці WEB додатку.

В розділі було розроблено функціональну структуру клієнтської та адміністративної частин сайту відповідно до вимог та принципів розробки WEB додатку.

Також було розроблено інтерфейс та логіку роботи WEB додатку, які повністю реалізують функції, що на них покладені.

Для більш змістовного наповнення було використано статичні графічні зображення, які надають контенту WEB додатку різноманітності.

Крім того було розглянуто основні принципи розробки динамічних WEB додатку. Проведений варіантний аналіз мов програмування та СКБД, що використовуються при розробці WEB додатків.

Також було розроблено засобами MySQL базу даних та таблиці, які використовувались для обробки за допомогою скриптів, реалізовані скрипти, що забезпечують повне функціонування бази даних і відповідає усім поставленим йому завданням, а саме додавання, редагування, видалення новин і окремих складових.

**4 ТЕСТУВАННЯ РОБОТИ WEB ДОДАТКУ**

У процесі створення WEB додатку, на етапі розробки, дуже важливо перевірити працездатність перед його розміщенням на хостингу. Для перевірки працездатності WEB додатку проводиться спеціальне тестування. Процес тестування стосується як програмної частини, так і частини користувацького інтерфейсу. Даний етап роботи є надзвичайно важливим, так як допомагає визначити всі помилки та недоробки, що виникли у процесі розробки WEB додатку.

**4.1 Тестування адмін-панелі WEB додатку**

В процесі тестування було здійснено такі перевірки:

* додавання, редагування та видалення новин;
* додавання та видалення користувачів WEB додатку;
* формування динамічних сторінок;
* додавання зображень до контенту сторінок.

**4.2 Тестування інтерфейсу користувача**

Під час тестування WEB додаток перевіряється на відповідність технічному завданню, перевіряються його технічні характеристики.

В процесі тестування було здійснено такі перевірки:

* переглянуто WEB додаток на різних моніторах. Переглядаючи WEB додаток на моніторах різних розмірів і при різному дозволі, дизайн WEB додаток на комп’ютерах потенційних відвідувачів не змінився, не з’явилась горизонтальна смуга прокручування на маленьких екранах, елементи дизайну при різному дозволі не змістились;
* переглянуто WEB додаток в різних браузерах. Переглядаючи WEB додаток в різних інтернет-браузерах і їхніх версіях, перевірено кросбраузерність WEB додатку. Для перевірки кросбраузерності було обрано чотири найпоширеніших браузера: Internet Explorer, Firefox, Opera та Google Chrome. Переглядаючи WEB додаток на різних браузерах впевнились у коректності відображення інформації різними браузерами;
* перевірено швидкість завантаження WEB додатку. WEB додаток завантажуються досить швидко, що робить його зручним у використанні;
* перевірено працездатність гіперпосилань WEB додатку. Під час перевірки не було виявлено посилань, які не працюють або ведуть на іншу сторінку.
* перевірено коректність відображення кольорів WEB додатку при різних налаштуваннях колірної палітри монітора, а також відображення шрифтів, анімації та графічних зображень;
* перевірено властивості кожної сторінки WEB додатку: заголовки, ключові слова, описи та інші мета-теги;
* перевірено зміст кожної сторінки на відповідність вихідним матеріалам й правопису на кожній сторінці.

Отже в ході тестування інтерфейсу кристувача WEB додатку було виявлено і усунено певні недоліки WEB додатку, які могли б вплинути на коректність роботи та зручність. Тепер WEB додаток працює коректно, однаково виглядає на моніторах різних розмірів, однаково відображається в найпопулярніших браузерах, зберігаючи при цьому правильне розміщення елементів дизайну (кольори, шрифти, анімації та графічні зображення тощо). Всі гіперпосилання WEB додатку робочі, працюють правильно. Висока швидкість завантаження сторінки.

**4.3 Тестування роботи бази даних**

Перевіримо правильність налаштування зв’язку з базою даних. Для цього розглянемо елементарний приклад роботи скрипта відправлення зі сторінки «Реєстрація», за допомогою якого дані про зареєстрованих користувачів зберігаються в онлайн базі даних. Приклад використання скрипта наведено на рисунку 4.1.

|  |
| --- |
|  |

Рисунок 4.1 – Робота скрипта збереження даних в базу даних «Користувачі»

Як бачимо, дані збережені в базу даних ― «Користувачі» відображаються коректно, що свідчить про правильне налаштування з’єднання з базою даних.

**Висновки**

Отже в даному розділі була здійснена перевірка працездатності WEB додатку перед його розміщенням на хостингу. Тестування проведено у двох напрямках: тестуванні користувацького інтерфейсу та адміністративної частини.

В ході тестування користувацького інтерфейсу було виявлено і усунуто певні невеличкі недоліки. Тепер WEB додаток працює коректно, однаково виглядає на моніторах різних розмірів, однаково відображається в найпопулярніших браузерах, зберігаючи при цьому правильне розміщення елементів дизайну (кольори, шрифти, анімації та графічні зображення тощо). Всі гіперпосилання робочі, працюють правильно. Швидкість завантаження сторінки висока.

При тестуванні адміністративної частини, було перевірено правильність роботи з базою даних, протестовано модулі реєстрації, управління користувачами, динамічними сторінками та новинами. В ході тестування не було виявлено збоїв роботи розроблених модулів. Після проведеного тестування можна зробити висновок, що WEB додаток відлагоджений і готовий до розміщення на хостингу.

Тестування бази даних коректно відобразило дані, що засвідчує правильне налаштування з’єднання з базою даних.

# **ВИСНОВКИ**

В ході виконання завдання на випускну кваліфікаційну роботу було досліджено галузь. На основі отриманої інформації було спроектовано WEB додаток, що відповідає потребам галузі.

Протягом виконання роботи було закріплено отримані під час навчання знання з таких предметів як розробка бази даних, web-програмування, архітектура програмного забезпечення, конструювання програмного забезпечення та програмування WEB додатків. Проведено детальний аналіз предметної області та визначені особливості архітектури програмного продукту.

Для розробки бази даних був використаний сервер MySQL і клієнт HeidiSQL. База складається з 7 таблиць, що зв’язані між собою таким чином, що на їх основі можна було побудувати реальну, працюючу модель.

Також було побудовано клієнтський додаток з використанням мов, технологій, фреймворків, плагінів таких як: Java Script, MySQL, XAMPP Control Panel.

Створений WEB додаток відповідає усім поставленим вимогам та задовольняє потреби користувача.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1 footballhd.ru [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://footballhd.ru/transfers/ (дата звернення 26.05.2019).

2 bbc.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.bbc.com/ukrainian/vert-cap-41097797 (дата звернення 25.05.2019).

3 readfootball.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: http://www.readfootball.com/transfer.html (дата звернення 26.05.2019).

4 il-calcio.ru [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://il-calcio.ru/serie-a-winter-transfers-2018/ (дата звернення 26.05.2019).

5 ua-football.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.ua-football.com/ua/offseason/transfers/premier (дата звернення 26.05.2019).

6 transfermarkt.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.transfermarkt.com/statistik/neuestetransfers (дата звернення 26.05.2019).

7 easy-code.com.ua [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: http://easy-code.com.ua/2012/07/sposib-opisu-funkcionalnix-vimog-do-sistemi-ta-%D1%97%D1%97-funkcij-z-vikoristannyam-standartiv-i-universalno%D1%97-movi-modelyuvannya-case-zasobi-modelyuvannya-programuvannya-statti/ (дата звернення 27.05.2019).

8 jetbrains.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://www.jetbrains.com/webstorm/ (дата звернення 27.05.2019).

9 uk.wikipedia.org [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://uk.wikipedia.org/wiki/Веб-застосунок (дата звернення 27.05.2019).

10 uk.wikipedia.org [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://uk.wikipedia.org/wiki/JavaScript (дата звернення 27.05.2019).

11 uk.wikipedia.org [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: <https://uk.wikipedia.org/wiki/Система_керування_базами> даних (дата звернення 27.05.2019).

12 helpx.adobe.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://helpx.adobe.com/ru/dreamweaver/using/general-information-coding-dreamweaver.html (дата звернення 27.05.2019).

13 uk.wikipedia.org [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: uk.wikipedia.org/wiki/Leaflet (дата звернення 28.05.2019).

14 habr.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://habr.com/ru/company/ruvds/blog/423983/ (дата звернення 28.05.2019).

15 helpx.adobe.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://helpx.adobe.com/ru/dreamweaver/using/general-information-coding-dreamweaver.html (дата звернення 27.05.2019).

16 github.com [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: https://github.com/benoitblanc/Football-transfers (дата звернення 28.10.2019).

17 Ітан Маркот Отзывчивый веб-сайт / Итан Маркотт – М:Манн, Иванов и Фербер, 2012. – С. 176

18 quintadb.ru [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – Режим доступу: <https://quintadb.ru/apps/c2kSoRWQPdP4k_W6eCW6SE> (дата звернення 29.05.2019).