

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра інформатики, програмної інженерії та економічної
кібернетики

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ REACT
ДЛЯ СТВОРЕННЯ WEB ТА МОБІЛЬНОГО ІНТЕРФЕЙСІВ
НА ПРИКЛАДІ СИСТЕМИ БРОНЮВАННЯ

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти “магістр”

Виконав: студент 2 курсу

Спеціальності 121 Інженерія програмного
забезпечення

Освітньо-професійної програми

«Інженерія програмного забезпечення»
другого (магістерського) рівня вищої
освіти

Краковський Олександр Олександрович

Керівник доктор педагогічних наук,
професор Круглик Владислав Сергійович

Рецензент кандидат педагогічних наук,
доцент Таточенко Володимир Іванович

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Принципи створення інтерфейсу додатків	7
1.1. Принципи проектування дизайну додатків.....	7
1.2. Особливості розробки web інтерфейсів.....	11
1.3. Особливості розробки мобільних інтерфейсів	17
РОЗДІЛ 2. React – JavaScript-бібліотека для розробки інтерфейсів ..	24
2.1. Огляд технології React.....	24
2.2. Переваги React для розробки інтерактивних інтерфейсів	26
РОЗДІЛ 3. Розробка інтерфейсу системи бронювання з використанням технологій React	31
3.1. Загальні вимоги до інтерфейсу систем бронювання.....	31
3.2. Реалізація web інтерфейсу системи бронювання	34
3.3. Реалізація мобільного інтерфейсу системи бронювання.....	39
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44
ДОДАТКИ	48
Додаток А.....	48

ВСТУП

Актуальність дослідження. У сучасному світі активно використовуються планшети, смартфони та інші мобільні пристрої. Із розвитком нових технологій і зростанням аудиторії користувачів, які стабільно використовують портативні пристрої щоб отримати доступ до мережі Інтернет, виникає потреба у розробці таких сайтів, які будуть адаптуватися до середовища залежно від можливостей мобільного пристрою. Під час розробки таких систем дуже важливо приділяти увагу проєтуванню інтерфейсу взаємодії користувача із системою [6].

Адаптивний вебдизайн є сучасним підходом у розробці вебсайтів, який дозволяє вебсайту реагувати на поведінку користувача і навколишнього середовища залежно від розмірів екрану, платформи та орієнтації пристрою.

Оскільки технології розробки адаптивних веб-сайтів дозволяють створити декілька проєкцій сайту для різних пристроїв (девайсів), найважливішим етапом постає створення інтерфейсу, який зміг би забезпечити якомога більш комфортну та ефективну взаємодію користувача із сайтом [5].

Сучасні вебсайти практично завжди є системами, що являють собою «інструмент», який використовується користувачем самостійно. Дуже часто вебсайти не мають докладної інструкції щодо його використання, яку можна було б прочитати завчасно. При першому користуванні вебресурсом користувач самостійно вивчає механізм роботи вебсайту, принципи його функціонування, спираючись на власний досвід і кмітливість [8].

Надмірно розширена функціональність лише ускладнює вебсайт, що робить його незручним при роботі. Наприклад, коли певна форма замовлення надміру насичена функціональними можливостями, вона навряд чи матиме успіх, оскільки буде складною для розуміння

користувачам. Тобто, широкий набір функціональності обмежує ефективність вебсайту. Оскільки саме досвід користувацької взаємодії формує думку про вебсайт, то за умови такого негативного досвіду наслідком буде втрата аудиторії. Такий досвід взаємодії з ресурсом є надзвичайно важливим для успіху вебсайту. Саме тому, стикаючись з дуже широким асортиментом функціональних можливостей, користувач змушений самотійно з'ясовувати, що ж саме відповідає його потребам [9].

Відмінності між десктопною (настільною, для ПК) та мобільною версіями сайту обумовлюються різними моделями поведінки кінцевих користувачів. Під час перегляду сайтів у web версії, призначеній для ноутбуків і широкоформатних моніторів, зазвичай характерною для користувачів є статична поведінка. Мобільні ж версії, в першу чергу, є джерелами інформації, з якими працюють динамічно, «на ходу», фокусуючись на найголовнішому.

Розглядаючи мобільну версію веб-ресурсу, слід мати на увазі, що навряд чи користувач, який вперше потрапив на сайт, повною мірою використає всі його функціональні можливості. Саме тому під час створення мобільного інтерфейсу варто не зменшувати перелік можливостей сайту, тобто закривати доступ до функціоналу, а зменшувати чи реорганізовувати інтерфейс. За таких умов функціонал, який використовується найчастіше, буде доступним, «на виду», а той, до якого користувач звертається рідше, може бути розміщеним у нижній частині сторінки або ж тимчасово прихованим [9].

Об'єкт дослідження – інтерфейс сайту.

Предмет дослідження – технології створення web та мобільного інтерфейсів.

Мета дослідження – розробити інтерфейс системи бронювання з використанням технологій React.

У зв'язку з поставленою метою було визначено **завдання**

дослідження:

1. Визначити основні принципи створення інтерфейсу додатків.
2. Проаналізувати особливості і вимоги до розробки web та мобільного інтерфейсів.
3. Розглянути бібліотеку React як інструмент розробки інтерактивного інтерфейсу.
4. Виконати аналіз вимог до інтерактивного інтерфейсу системи бронювання.
5. Реалізувати web та мобільний інтерфейси системи бронювання за допомогою технології React.

Методи дослідження. Для вирішення поставлених завдань були використані наступні методи:

- теоретичні: аналіз, синтез, систематизація, зіставлення, класифікація наукових джерел інформації з веб-програмування;
- практичні: проектування та створення додатку здійснювалось за допомогою: HTML5, CSS3 (SCSS), JavaScript (React.js).

Зв'язок роботи з науковими програмами, планами, темами. Кваліфікаційна робота пов'язана з напрямом наукових досліджень кафедри інформатики, програмної інженерії та економічної кібернетики факультету комп'ютерних наук, фізики та математики Херсонського державного університету.

Наукова новизна одержаних результатів. Результати кваліфікаційної роботи є новими і полягають у використанні технології React до реалізації web та мобільного інтерфейсів системи бронювання басейну університету.

Практична цінність роботи полягає у тому, що представлені результати дозволяють застосовувати розроблений інтерфейс системи бронювання для підвищення ефективності роботи окремого структурного підрозділу «Басейн» Херсонського державного університету.

Апробація результатів роботи. Основні положення та результати

роботи були обговорені на засіданні кафедри інформатики, програмної інженерії та економічної кібернетики. Також здійснено їх оприлюднення шляхом публікації у фаховому збірнику наукових праць.

Структура роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1

ПРИНЦИПИ СТВОРЕННЯ ІНТЕРФЕЙСУ ДОДАТКІВ

1.1 Принципи проєктування дизайну додатків

Функціональність системи часто розглядається розробниками окремо від її користувальницького інтерфейсу. Однак за такого підходу вважається, що користувачий інтерфейс – це своєрідне доповнення до функціональності системи. З позиції користувача програмних засобів, функціональність та користувачий інтерфейс, як правило, не розрізняються, тобто користувачі сприймають саме інтефейс як програму: якщо інтерфейс задовольняє їх потреби, то і сама програма гарна і зручна.

Користувальницький інтерфейс – це цілісна система правил і засобів, якою регламентується і забезпечується взаємодія між програмою і користувачем [30].

Програмне забезпечення має розроблятися з урахуванням вимог і побажань користувача, тобто система повинна підлаштовуватися до користувача. Ось чому принципи проєктування настільки важливі.

Принципи розробки інтерфейсу – це високорівневі концепції і уявлення, які можуть використовуватися при проєктуванні програмного забезпечення.

Інтерфейс – це засіб, що забезпечує зручну взаємодію користувача з інформаційною системою, тобто сукупність засобів, призначених для обробки та відображення інформації, що якнайкраще пристосовані для зручності використання [30].

Взаємодію людини і сайтів, мобільних додатків, програм повинна бути спроектована дизайнерами інтерфейсів (UI/UX-дизайнерами). Мета їх роботи полягає у розробці логічної і дружньої до користувача взаємодії людини і програми [40].

Поняття UX дизайну.

UX дизайн (UX design). «User Experience Design» в перекладі означає «досвід взаємодії» і позначає весь досвід, який отримує користувач при взаємодії із сайтом або додатком. Він включає в себе різні UX -компоненти:

- інформаційну архітектуру,
- проектування взаємодії на конкретні дії користувача,
- візуальний та інтерактивний дизайн,
- юзабіліті,
- контент [45].

UX-дизайн відповідає за функції, адаптивність продукту і те, які емоції він викликає у користувачів. Чим зрозуміліше інтерфейс, тим легше користувачеві отримати результат і зробити цільову дію.

Отже, UX-дизайн — це проектування інтерфейсу на основі досліджень користувацького досвіду та поведінки [48].

В цілому, UX дизайн зосереджений на комплексному підході до взаємодії користувача з інтерфейсом веб-сайту, мобільного додатку чи будь-якої іншої програми. Іншими словами, UX дизайном називають процес створення корисних, простих, а також приємних у використанні продуктів (цифрових або ж фізичних). Основною ціллю UX дизайну є покращення досвіду взаємодії з продуктом таким чином, щоб користувачі знаходили у ньому додаткову цінність.

Фахівець, який займається UX дизайном – *UX-дизайнер* (UX-архітектор, UX -інженер) – у процесі розробки інтерфейсу має (за можливості) якомога більше враховувати такі аспекти, як середовище користувача і тип електронного пристрою, аж до способів вводу та відображення інформації [40].

UX дизайн зосереджений на розв’язанні ряду питань, зокрема таких, як:

- Постановка мети і завдань.

- Здійснення підбору відповідних UX інструментів для реалізації цілей.
- Розробка продукту, якомога більш зручного та легкого для сприйняття цільової аудиторії.
- Аналіз кінцевого результату – встановлення, чи є відповідність між продуктом і очікуваннями замовника, а також рівню вдовolenості користувачів.

Саме продуманість усіх наведених аспектів на цих етапах дозволить збільшити кількість клієнтів вашої компанії та користувачів продукту [19].

UI дизайн (UI design).

User Interface Design або користувацький інтерфейс – це вужче за UX поняття, до якого входить деякий набір графічно оформлених технічних елементів (таких як кнопки, чекбокси, селектори тощо). Завдання UI design полягає у наданні допомоги користувачу та організації взаємодії із програмою / сайтом [48].

До UI включаються:

- візуальний дизайн,
- баланс кольорів,
- дизайн інтерфейсу,
- оформлення тексту,
- ілюстрації,
- графіка.

Наразі до основних правил UI дизайну відносять:

1. *Організованість елементів інтерфейсу.* Таким чином, елементи мають бути логічно структурованими і взаємопов'язаними.
2. *Угрупування елементів інтерфейсу.* Мається на увазі групування елементів, що мають логічний зв'язок (наприклад, меню та форми).
3. *Вирівнювання елементів інтерфейсу.* Якщо вирівнювання елементів відсутнє, це негативно впливає на зручність його

використання.

4. *Єдиний стиль елементів інтерфейсу.* Оформлення інтерфейсу у певному стилі покликане залишити емоційний слід у пам'яті користувача.
5. *Наявність вільного простору.* Він розмежовує інформаційні блоки між собою, при цьому увага користувача зосереджується на певному елементі.

Розроблений за всіма правилами користувацький інтерфейс значною мірою підвищує ефективність використання ресурсу і надає йому конкурентні переваги.

З огляду на вищезазначене, можна виділити спільні риси UI та UX:

- візуальний дизайн,
- дизайн інтерфейсу,
- оформлення тексту,
- ілюстрації, графіка [48].

Комфорт користувача залежить від UX дизайну. UX охоплює усе, що вміщає UI, і навіть більше. Бути UX дизайнером означає поряд з глибоким розумінням потреб та бажань користувачів, брати участь у аналізі та дослідженні зворотнього зв'язку від споживача та тестувати процес дизайну. Загалом, UX включає весь процес взаємодії між клієнтом та компанією.

Окремо слід виділити інформаційну архітектуру (Information Architecture, IA). Її діяльність зосереджена на організації даних, тобто на тому, якою мірою інформація структурована з позиції користувача, а не технічних чи системних правил. Нею визначається розміщення елементів на сторінці та їх взаємозв'язок. Компетенція інформаційної архітектури – це швидше меню й навігація, а також їх грамотне використання [19].

Зручність та краса інтерфейсів часто проєктуються однією й тією ж самою людиною, однак інтерфейси стають все складнішими, тому професію поділяють на дві. User Interface Designer (UI) – це той, хто

відповідає за красу та задоволення, тобто за те, як саме продукт представлено користувачам. Він працює з кольорами, іконками, типографією, навігацією, у тому числі з меню, вікнами та кнопками, анімацією, сповіщеннями. UI-фахівець створює дизайн, базуючись на даних, які отримує від UX-фахівця [24].

User Experience Designer (UX) – це той, хто відповідає за зручність та відповідність ресурсу бізнес-задачам. Він здійснює проектування дизайну сайту, додатку таким чином, щоб користувачам було зручно і зрозуміло, і вони могли отримати від сайту необхідне з докладанням мінімальних зусиль [17].

UX-дизайнер вивчає користувацькі проблеми, розбирається у їх поведінці, досліджує досвід. Він повинен впевнитися, що продукт працює логічно та вирішує конкретні проблеми. До ключових обов'язків UX-фахівця відносять: дослідження аудиторії та продукту, проектування користувацьких сценаріїв. UX-дизайнер займається «щастям» користувача: задоволенням і продуктивністю від роботи з інтерфейсом, загальним розумінням і легкістю вирішення проблем [45].

Ролі обох дизайнерів перетинаються, тому неможливо займатися виключно UI без знань UX – і навпаки. Дуже часто обидва види робіт виконує одна людина – UI/UX-дизайнер.

Підсумовуючи, UI та UX – це інструмент пошуку рішень, які відносяться до візуального дизайну та дизайну, що пов'язаний користувацькою взаємодією.

1.2 Особливості розробки web інтерфейсів

Вибір конкретних рішень під час розробки веб-інтерфейсів залежить від особливостей самої інформації та потреб роботи з нею.

Відмінності між десктопною (настільною, для ПК) та мобільною версіями сайту обумовлюються різними моделями поведінки кінцевих користувачів.

Основні вимоги до дизайн-макетів сайту:

1. Сітка, розміри, ширина контенту [1]:

- Елементи бажано розставляти по сітці, наприклад, «960 grid» або подібні. Якщо використовується інша сітка, то необхідно вказати її назву.
- Ширина макета повинна бути 1920px.
- Мінімальна ширина тематичної (інформаційної) області - 1000 px (для десктопа).
- Максимальна висота для області контенту (якщо блоки сайту робляться екранами) 700 px, включаючи відступи.
- Якщо потрібно адаптивна (респонс) верстка, потрібні макети для реперних точок і коментарі по поведінці верстки між цими точками, наприклад:
 - 320-480 px,
 - 480-1024 px,
 - 1024-1920 px.

2. Зображення та іконки:

- Намагатися не робити великих растрових зображень, вони обтяжують сторінку.
- Зображення, які можна зробити в векторі - робити в векторі.
- Іконки повинні бути в векторному форматі (.svg).
- Для коректного відображення картинок і елементів на дисплеях Retina все, що можна - необхідно робити в .svg форматі, якщо для зображення такої можливості немає - робити іконки і картинки в два рази більше, ніж вони є. Наприклад, якщо в макеті іконка 100x100 px, то для ретини повинен бути розмір 200x200 px. Великі картинки необхідно зашивати в вихідні відразу.
- Потрібно свідомо обирати пропорції і розміри об'єктів і робити їх кратними 10, наприклад, 1000px, а не 998px.

3. Тексти і шрифти [1; 4; 32]:

- Використовувати «безпечні» шрифти встановлені в Windows. При необхідності використовувати нестандартні, вказувати запасні безпечні (fallback font).
- Розмір шрифту повинен бути без дробів. Ніколи не потрібно розтягувати шрифт примусово.
- Не допускається растрування тексту або розміщення його в smart-об'єкті.
- Використовувані шрифти, за винятком Tahoma, Arial, Verdana, Times New Roman, Courier, необхідно прикладати до переданих матеріалів. Передані шрифти повинні бути тільки форматів ttf і otf.
- Припустимо і навіть вітається використання вільних кирилических шрифтів від Google web fonts.
- Необхідно описати поведінку посилань в дизайні («неактивна», «при наведенні», «відвідана») в меню / модулі, те ж саме стосується посилань, що відрізняються від дефолтного стилю (наприклад, посилання в меню, pathway і т.д.)
- Шрифти можна вибрати на сайті. Їх можна з легкістю використовувати і підключати на сайтах.
- Якщо використовуються обрані шрифти - скоротити їх кількість і кількість накреслень до мінімально необхідного.
- Шрифт повинен бути безкоштовним або повинна бути придбана ліцензія на його використання.
- Згладжування шрифтів завжди має бути відключено.

4. *Styleguide* або *UI KIT* [5]:

- Намалювати активні стани елементів (меню, кнопки, поля форм і тд):
 - hover (наведення)
 - focus / active (клік)
 - preloader (для аjax-кнопок)

- Намалювати як будуть виглядати стандартні елементи форматування тексту: параграф, абзац, заголовки з 1 по 4 рівень, маркований список, нумерований список, підкреслений текст (посилання), жирний, курсив і ін.
- Обов'язково зобразити всі спливаючі вікна, прелоадери, спливаючі меню, інформаційні повідомлення, валідації в формах і інші ефекти.
- Використовувати кінцеве число кольорів у макеті і описати їх в styleguide.

Компоненти сайту мають відповідати наступним вимогам [8]:

Посилання. Оформлення посилань повинно бути вказано в трьох статусах: звичайне посилання (link), посилання при наведенні курсору (hover link), відвідане посилання (visited link). Колір посилання зазвичай блакитний, а сам текст посилання підкреслено [18].

Форма. Елементи форми повинні мати два стани: стандартний, обраний (активний). При фокусі користувача на елементі потрібно його виділити, наприклад рамку другим кольором або змінити фоновий колір.

Кнопки. Повинні бути в 2-3 станах, тобто, стандартний, при наведенні курсору та активний. Зазвичай кнопки додають внутрішній відступ, для того щоб, збільшити її та акцентувати увагу користувача на ній.

Навігація. Навігація сайту - це можливість переходу між його сторінками. Чим простіше і зрозуміліше навігаційна система, тим краще для відвідувачів, оскільки вона дозволяє знаходити необхідну інформацію і швидко орієнтуватися в розділах сайту. Перерахуємо критерії якісної навігації [17]:

- Простота - всі елементи меню або призначеного для користувача інтерфейсу повинні бути добре видимими і інтуїтивно зрозумілими для будь-якого користувача. В ідеалі, щоб пройти до будь-якій сторінці або розділу сайту, користувачеві повинно

знадобитися не більше 3 кліків.

- Доступність на будь-якій сторінці сайту - якісно опрацьовані елементи навігації повинні бути присутніми на кожній сторінці сайту, щоб у користувача була можливість перейти з будь-якого розділу в необхідний.
- Графічне оформлення – елементи навігації повинні бути контрастні фону і відрізнятися від основного тексту, але при цьому гармоніювати із загальним дизайном ресурсу.

Заголовок. Зазвичай заголовок на сторінці роблять більшого розміру ніж основний текст, другого кольору, використовують інший шрифт або виділяють жирним стилем.

Параграф. Параграф зазвичай має розмір шрифту від 14 до 16 пікселів та має невеликий, нижній відступ.

Пагінація. В пагінації повинні бути елементи управління сторінками, тобто стрілки або слова вперед/назад. Відображення кількості наступних або попередніх сторінок на поточній сторінці повинно бути від 3 до 5 елементів.

Банер. Банер або слайдер може мати елементи керування (стрілки зазвичай) або «точки» на які можливо нажати та перейти до наступного вмісту слайду. Банер повинен містити посилання на інформацію, яка в ньому описується.

Картка товару. Містить в собі: зображення продукту, заголовок, короткий опис та кнопку для повного перегляду. Зображення, заголовок та кнопка повинні містити в собі посилання на сторінку повної інформації про товар.

Підвал сайту. Підвал сайту повинен бути після всієї інформації на сторінці та включати в себе наступні елементи: копірайт, логотип сайту, навігацію, контакти .

Модальне вікно. Зазвичай такі вікна роблять для отримання даних від користувача. Така форма повинна мати одне або декілька полів введення

інформації та кнопку «Надіслати». Також модальна форма повинна мати хрестик для її закриття. Зазвичай він знаходиться в верхньому кутку самого вікна.

Картка статті. Вона схожа з карткою продукту, але сюди додають дату публікації. Тобто карта статті містить в собі: зображення, дату публікації, заголовок, опис статті та посилання на повний опис.

Сайдбар. Зазвичай сайдбар називають бічну колонку сайту, яка розміщується праворуч або ліворуч від області основного контенту. В сайдбарі зазвичай знаходяться: банери, допоміжні меню, форми, віджети тощо.

Список. Список має внутрішній лівий відступ в 10-15 пікселів для того щоб виділити його вміст та нижній зовнішній відступ в 10-15 пікселів [23].

Отже, під час перегляду сайтів у web версії, призначеній для ноутбуків і широкоформатних моніторів, зазвичай характерною для користувачів є статична поведінка, коли вони мають зручний маніпулятор для роботи з ресурсом, тобто мишку або тачпад, трекпад.

З урахуванням цього, під час розробки десктопної версії сайту можна:

- розміщувати досить велику кількість видимого контенту на екрані;
- організувати зручну для користувача навігацію у вигляді вкладеного меню, списку категорій;
- горизонтально розташовувати тематичні блоки (використовуючи декілька колонок);
- надавати користувачам широкі можливості для роботи з фільтрами і пошуком по сайту;
- розміщувати різноманітні мультимедійні інтерактивні елементи та дизайнерські ефекти.

1.3 Особливості розробки мобільних інтерфейсів

Мобільні версії ресурсів, в першу чергу, є джерелами інформації, з якими працюють динамічно, «на ходу», фокусуючись на найголовнішому. Для перегляду сайтів в такому ритмі важливі наступні параметри [47]:

- висока швидкість завантаження вебсторінки;
- відповідь сайту на певні користувацькі дії повинна бути однозначною і точно відповідати запиту;
- на головній сторінці вебресурсу у мобільній версії користувачі повинні отримати максимум корисної та конкретної інформації;
- елементи, які можуть відволікати увагу, мають бути відсутні;
- навігація повинна бути зрозумілою і простою, тобто користувачі повинні мати можливість дістатися до будь-якої інформації лише за кілька кроків (без використання різних списків і фільтрів);
- текст ресурсу повинен бути лаконічним і зручним для прочитання (при цьому дизайн відходить на другий план);
- Не рекомендується використовувати на сайті більше 4 шрифтів;
- мультимедійні дані повинні бути розміщені в такому форматі, який підтримується більшістю сучасних мобільних пристроїв (слід уникати flash-технологій);
- зміст сайту має бути лінійним (найважливіший контент повинен розташовуватися ближче до початку сторінки);
- для досягнення цілей користувачі повинні здійснювати мінімум дій (наприклад, здійснення дзвінка після натискання кнопки, покупка товару за один клік тощо).

Створити мобільну версію сайту «з нуля» можна різними способами. Найпопулярніші з них – це самостійна розробка, використання CMS або ж конструкторів сайтів [35, 36].

Самостійна розробка найчастіше застосовується, якщо у вас достатньо ресурсів, і проєкт досить великий. Крім того, розробка

унікальної мобільної версії дозволяє вносити тонкі зміни, недоступні при використанні CMS, а також конструкторів.

Використання CMS (система керування вмістом) – це найпоширеніший і простий спосіб будування сайтів. Найпопулярнішими CMS є WordPress, Webasyst, Joomla та Drupal.

Системи управління контентом, які створені компаніями-розробниками, бувають як платними, так і безкоштовними.

Безкоштовні CMS обмежені набором функцій і технічної підтримки, зазвичай використовуються для створення особистих або невеликих корпоративних сайтів. Нерідко мають проблеми з безпекою, тому для створення серйозних проєктів не розглядаються.

Комерційні CMS – це потужні, універсальні системи для розробки різних типів сайтів із захистом і техпідтримкою.

CMS використовують, коли:

- не потрібні специфічні функції.
- проєкт не передбачає високого навантаження.
- обмежений бюджет.
- стислі часові рамки.
- потрібен простий і зрозумілий інструмент.

На масовій CMS можна створити всі основні види сайтів. Більшість систем складається з модулів. Набори таких модулів складають редакції сайтів: корпоративний сайт, візитка, каталог, інтернет-магазин, блог, портал і так далі. Це найпоширеніший і простий спосіб сайтобудування.

Застосування CMS має свої переваги та недоліки. До переваг можна віднести:

- Легко вивчити і налаштувати стандартний сайт.
- Модулі. Можна розширювати можливості за рахунок плагінів.
- Швидка швидкість розробки. Необхідні лише контент, дизайн та налаштування.
- Технічна підтримка.

- Повноцінна документація. Для масових комерційних CMS легко знайти всю необхідну документацію.
- Готові платформи мають інтерфейс прикладного програмування API, який дозволяє інтегрувати ресурс з іншими сервісами.

До недоліків CMS належать:

- Обмежена функціональність. Як правило, у кожної CMS є своя спеціалізація.
- Невисока продуктивність.
- Надмірність деяких модулів. Більшість можливостей може взагалі не використовуватися.
- Уразливість. Оскільки основна маса комерційних сайтів зроблена на популярних версіях CMS, саме на них спрямовані атаки.

Конструктори сайтів призначені для непрофесіоналів і для тих проектів, для яких немає сенсу застосовувати більш дорогі технології.

Найпопулярніші і потужні конструктори на ринку:

- Wix;
- Tilda;
- uKit.

З їх допомогою буквально за кілька годин збирається цілий сайт або прототип, по дизайну і оформлення не поступається професійним дорогим ресурсам.

У конструкторах ви можете використовувати прості і наочні технології:

- Drag-n-drop - перетягування елементів.
- WYSIWYG - «Що бачите, те й отримуєте». Підхід, при якому дизайн при розробці максимально відповідає фінального.
- Сотні шаблонних тем і блоків.
- Елементи структури - кнопки, форми, списки, інфографіка,

ілюстрації.

- Інтеграції з CRM, сервісами email-розсилок, аналітики, збору даних, комунікації з відвідувачами.

Методи адаптації сайту для мобільних пристроїв

Існують три принципово різних підходи до адаптації сайту під мобільні пристрої. Кожен з методів має свої переваги і свої недоліки. Власники сайтів та / або розробники повинні розуміти різницю в даних підходах і самостійно приймати рішення щодо адаптації сайту на основі цих методів [23].

1. Мобільна версія

Мобільна версія передбачає розробку окремої версії сайту, на яку відвідувач буде автоматично перенаправлений в разі використання мобільного пристрою. Як правило, така версія розміщується на новому домені (наприклад, m.site.ua).

Необхідно розуміти, що дана технологія дозволяє створити міні-версію основного сайту з найбільш важливою інформацією. Мобільна версія підходить для всіх типів сайтів від інтернет-магазинів до сервісів розважального характеру: блоги, новини, соціальні мережі.

Переваги мобільної версії сайту:

- мобільна версія є зручною для користувача, видає найбільш пріоритетну інформацію, а також дозволяє здійснювати замовлення або покупку товару за мінімальну кількість кроків;
- в мобільну версію просто вносити істотні зміни, тому що вона є окремою версією;
- швидкість завантаження сторінок збільшується, тому що всі необхідні елементи мають менший інформаційний обсяг;
- завжди залишається можливість перейти на повну десктопну версію сайту.

Недоліки мобільної версії сайту:

- висока вартість розробки: розробку мобільної версії сайту можна

порівняти зі створенням повноцінного сайту або програми;

- необхідно враховувати і вартість обслуговування: доведеться підтримувати працездатність, як основної версії сайту, так і мобільної;
- мобільна версія сайту має на увазі відмову від частини інформації і можливостей сайту [8].

2. Динамічний показ

Технологія динамічного показу має на увазі адаптацію сайту для конкретних розширень екранів мобільних гаджетів. При використанні даної технології використовуються різні варіанти коду для різних мобільних пристроїв, доступні по одному URL. При виборі потрібного варіанти показу сервер орієнтується на браузер користувача. Розташування елементів при цьому є фіксованим для різних пристроїв.

Переваги динамічного показу:

- простота розробки адаптованих версій сайту;
- відсутність потреби вносити зміни окремо в мобільну версію сайту;
- розробникам набагато простіше контролювати зовнішнє представлення сайту - дизайн і контент - на різних пристроях.

Як *недолік динамічного показу* можна виділити різноманітні проблеми при відображенні сайтів на тих мобільних пристроях, розміри екранів яких не збігаються з заявленими виробниками [5].

3. Адаптивний дизайн

Технологія адаптивного дизайну означає, що за рахунок певних css-стилів коду, сайт автоматично адаптується під розміри екрану пристрою, на якому він відкритий. Розробники сайтів верстають їх не під певний розмір екрану, а створюють сторінки сайту з таких елементів, які автоматично можуть підлаштуватися під ширину дисплея. Блоки, з яких складається сайт, змінюють своє місце розташування, а деякі взагалі виключаються на смартфонах.

При розробці адаптивного дизайну необхідно використовувати мета-

тег *meta name="viewport"*, який вказує браузеру, яким саме чином слід корегувати зміст вебсторінки. Синтаксис мета-тегу наступний:

```
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

Сторінку без мета-тега браузер користувача приймає за сторінку для настільних комп'ютерів і масштабує сайт по ширині екрану мобільного пристрою. В результаті зміст читається досить непросто.

Сторінка з мета-тегом *viewport*, який вказує, що область перегляду дорівнює ширині екрану пристрою, не масштабується браузером мобільного пристрою, і її зміст читається легко.

Важливо, що якщо в якості значення елемента *width* в мета-тегу *viewport* вказується *device-width*, це значення оновлюється, якщо користувач змінює орієнтацію екрану планшета або смартфона.

Переваги адаптивного дизайну:

- однакове відображення web-сторінок на будь-яких мобільних пристроях за рахунок адаптації стилів до браузерів;
- відповідність вимогам Google до юзабіліті сайту на мобільних пристроях;
- вартість розробки нижче вартості створення мобільної версії сайту;
- використання однієї url-адреси для сайту, завдяки чому не буде потреби перенаправляти користувачів;
- можливість зберегти індивідуально розроблений дизайн сайту [5].

Недоліки адаптивного дизайну:

- при адаптивному дизайні збільшується інформаційний обсяг сайту, тому що елементи дизайну не змінюються, отже, з'являється необхідність додатково оптимізувати швидкість завантаження;
- у відвідувача немає можливості перейти на звичайну версію сайту;
- складність розробки адаптивної версії для вже існуючих сайтів - цей параметр важливий в тому випадку, якщо бізнес цілком залежить від сайту і від його роботи;

- можливе виникнення проблем з адаптацією блоків на сайтах з досить складним дизайном [5].

Висновки до розділу 1

Отже, нами були розглянуті принципи створення інтерфейсу додатків. Користувальницький інтерфейс – це сукупність засобів, призначених для обробки та відображення інформації, що якнайкраще пристосовані для зручності використання користувачем.

Здійснено аналіз поняття UI/UX дизайну сайтів та вимоги до нього, виконано огляд основних принципів проектування дизайну додатків з позиції UI/UX-дизайнерів (дизайнерів інтерфейсів).

Розглянуто особливості розробки веб інтерфейсів, наведено основні вимоги до дизайну web версії сайту, зокрема, для таких елементів, як сітка, розміри, ширина контенту, зображення та іконки, тексти і шрифти.

Виокремлено важливі параметри, які враховуються при розробці мобільних інтерфейсів з огляду на динамічність роботи з ними. Розглянуто способи створення мобільної версії сайту «з нуля», такі як самостійна розробка, використання CMS або ж конструкторів сайтів. Наведено найпопулярніші системи, проаналізовано та узагальнено переваги і недоліки кожного методу.

Також було виконано огляд різних підходів до адаптації вже існуючого сайту під мобільні пристрої, їх сильних та слабких сторін: створення мобільної версії, застосування технології динамічного показу, технології адаптивного дизайну.

РОЗДІЛ 2

РЕАКТ – JAVASCRIPT-БІБЛІОТЕКА ДЛЯ РОЗРОБКИ ІНТЕРФЕЙСІВ

2.1 Огляд технології React

Мова програмування JavaScript є однією з найпопулярніших у веб-розробці, і причиною тому є не тільки її універсальність, але і необмежені можливості за переказами проекту інтерактивності.

Будь-який веб-сайт, десктопних або мобільний додаток складається з Front-end частини, яка виконується в браузері, і Back-end частини, що виконується на сервері. JavaScript – єдина мова програмування, який здатний виконуватися на стороні клієнтської частини (в браузері). Спочатку він розроблявся для виконання невеликих скриптів в браузері, які надавали статичним HTML сторінкам динаміку [33].

За більш, ніж 20 років існування, JavaScript пережив безліч технологічних проривів і тепер мова виконується не тільки на клієнтській, але і на серверній частини, як на десктопних додатках, так і на мобільних платформах. Мова постійно розвивається, щодня за допомогою нього пишеться величезна кількість бібліотек, які прискорюють розробку і роблять її більш зручною [33].

React – JavaScript-бібліотека для роботи з користувацькими інтерфейсами (UI), яку створили розробники Facebook. Бібліотеку почали використовувати на сайті цієї соціальної мережі в 2011 році. А в 2013 році Facebook відкрив вихідний код React [14].

За допомогою React розробники створюють веб-додатки, які змінюють відображення без перезавантаження сторінки. Завдяки цьому додатки швидко реагують на дії користувача, наприклад, заповнення форм, застосування фільтрів, додавання товарів в корзину і так далі.

React застосовують для відтворення компонентів для користувацького інтерфейсу. Також бібліотека може повністю управляти

фронтедом. В цьому випадку React використовують з бібліотеками для управління станом і роутінга, наприклад, Redux і React Router [16].

Ключові особливості React:

- декларативність,
- універсальність,
- компонентний підхід,
- віртуальний DOM,
- JSX.

Одна з ключових особливостей React - універсальність. Цю бібліотеку можна використовувати на сервері і на мобільних платформах за допомогою React Native. Це принцип “Learn Once, Write Anywhere” або «Навчіться один раз, пишiть де завгодно» [25].

Ще одна важлива особливість бібліотеки - декларативність. За допомогою React розробник описує, як компоненти інтерфейсу виглядають в різних станах. Декларативний підхід скорочує код і робить його зрозумілим.

React заснований на компонентах, це ще одна ключова особливість бібліотеки. Кожен компонент повертає частину призначеного для користувача інтерфейсу зі своїм станом. Об'єднуючи компоненти, програміст створює завершений інтерфейс веб-додатку.

Важлива особливість React – використання JSX. Це розширення синтаксису JavaScript, яке зручно використовувати для опису інтерфейсу. JSX дозволяє писати JavaScript-код за допомогою готових компонентів, які практично повністю повторюють HTML. Це спрощує розробку.

До важливих особливостей React відноситься використання віртуального DOM (Virtual DOM). Віртуальний DOM – об'єкт, в якому зберігається інформація про стан інтерфейсу. При зміні стану, наприклад, після відправки форми або натискання кнопки, React розраховує різницю між старим і новим станом. Після цього бібліотека отрисовує нове стан. Використання віртуального DOM дозволяє бібліотеці ефективно

оновлювати реальний DOM [14, 16].

Бібліотека React підходить для використання під час командної розробки, оскільки у ній суворо дотримується UI і шаблон робочого процесу розробки. Крім того, UI код читабельний і простий в супроводі.

2.2 Переваги React для розробки інтерактивних інтерфейсів

В даний час JavaScript-фреймворки є дуже популярними серед розробників. Осягнути переваги певної технології можна, порівнявши її з інструментами аналогічного або спорідненого призначення.

Angular вперше був випущений у 2010 році, розроблений компанією Google. Це JavaScript-фреймворк, заснований на TypeScript.

React спочатку був випущений у 2013 році Facebook. Крім Facebook, він використовується Instagram та WhatsApp.

Vue (або Vue.js) – наймолодша із всіх перерахованих технологій, розроблена у 2014 році.

Компоненти є невід'ємною частиною всіх фреймворків. Загальний робочий процес компонента подібний до того, як він приймає вхідні дані і відповідно реагує на них [22].

У Angular компоненти називаються директивами, які використовуються в якості маркерів для елементів об'єктної моделі документа (DOM), вони можуть відслідковувати і визначати конкретну поведінку кожного окремого компонента.

З тієї чи іншої причини Angular зазвичай виділяє частину компонентів для користувача інтерфейсу, як атрибути тегів HTML, а їх поведінку - у вигляді коду JavaScript.

React зазвичай об'єднує призначений для користувача інтерфейс і поведінку компонентів. Наприклад, код, який може створити в React компонент «hello world», і та ж частина коду, відповідає за розробку елемента користувальницького інтерфейсу, а також відстеження його поведінки [14].

Vue надає можливість налаштування, що дозволяє комбінувати призначений для користувача інтерфейс і поведінку компонентів в одному скрипті. Крім того, він дозволяє використовувати препроцесори, а не CSS, що в даний час полегшує роботу розробників. Vue може інтегрувати інші бібліотеки, такі як Bootstrap.

Вважається, що просунутий і професійний Angular вимагає від вас вивчення пов'язаних понять, таких як TypeScript і Model View Controller (MVC). Хоча для вивчення Angular потрібен якийсь час, але у ньому є чітке уявлення про те, як працює інтерфейс.

Стосовно вивчення React – по-перше, React не є повноцінним фреймворком, і для розширених функцій йому потрібні сторонні бібліотеки, що може спростити процедури навчання. Більш того, в документації по React наводяться рішення поширених проблем.

Vue легше вивчати, ніж Angular або React, оскільки він надає більш широкі можливості налаштувань. Цікаво, що функціональні можливості Vue частково збігаються з Angular і React, наприклад, існують компоненти, що забезпечують перехід до Vue з будь-якого з двох інших.

Розмір бібліотек є важливою характеристикою, яку слід враховувати, так як іноді час виконання залежить від розміру файлів.

Середній розмір файлу:

- Angular: 500+ KB
- React: 100 KB
- Vue: 80 KB

Незважаючи на те, що відмінності між цими файлами значні з точки зору розміру, вони все ж є невеликими в порівнянні із середнім розміром веб-сторінки, який зазвичай перевищує 2 МБ. Використання Мережі доставки контенту (CDN) прискорює завантаження цих файлів.

Однією з важливих характеристик для обраних технологій є міграція між версіями.

Хоча для переходу з однієї версії на іншу не потрібно багато змін,

важливо відстежувати ці моменти, так як іноді оновлення можуть бути більш значними і необхідно приділити увагу сумісності.

Зазвичай оновлення Angular виходять раз на півроку. Перед випуском будь-яких основних API-інтерфейсів витримується пауза ще в шість місяців, що дає вам два циклу релізів - приблизно один рік для внесення необхідних змін.

У React поновлення між версіями більш зручні, ніж в Angular і Vue [14]. Більш того, такі скрипти, як react codemod, допоможуть вам мігрувати.

Нещодавно Facebook заявив, що стабільність є їх головним завданням, оскільки React використовують такі великі компанії, як Twitter і Airbnb.

Відносно міграції Vue слід почати з того, що 90% API залишаються тими ж, якщо ви переходите з 1.x на 2. Крім того, існує допоміжний інструмент міграції, який працює з консолі.

Наведемо приклад кращих сайтів, які використовують вказані технології:

- *Vue*: 9gag, GitLab, Laravel, Font Awesome.
- *React*: Airbnb, Instagram, Tesla, Dropbox.
- *Angular*: Netflix, PayPal, Lego, The Guardian.

Отже, React – це популярна JavaScript-бібліотека для роботи з користувацькими інтерфейсами. До її основних особливостей відноситься декларативність, компонентний підхід, універсальність, використання віртуального DOM і JSX.

Існує безліч факторів, які впливають на вибір платформи для реалізації конкретного проєкту програми, оскільки кожна з порівнянних технологій має свої переваги і недоліки. Щоб вибрати будь-яку з них, розробник повинен мати чітке уявлення, яким повинен бути кінцевий продукт.

Порівняння Angular, React, Vue

	Angular	React	Vue
Рівень складності	Високий	Середній	Низький
Розмір файлу	500 Кб	100 Кб	80 Кб
Час завантаження	довго	середнє	Швидко
Швидкість розробки	Швидка розробка	Більш просте масштабування архітектури	Швидка розробка
Самодостатність	Комплексне середовище	Установка додаткових бібліотек обов'язкова	Застосовуються сторонні бібліотеки
Випущений у	2010	2013	2014
Використовується	Google, Wix	Facebook, Uber	Alibaba.com, GitLab
Офіційний сайт	angular.io	reactjs.org	vuejs.org
Поточна версія	v. 1.8.0	v. 17.0.1	v. 3.0

Висновки до розділу 2

React – популярна JavaScript-бібліотека для роботи з користувацькими інтерфейсами (UI). За допомогою неї розробники створюють веб-додатки, які змінюють відображення без перезавантаження сторінки. До ключових особливостей React належать декларативність, універсальність, компонентний підхід, віртуальний DOM, JSX.

Для визначення переваг React для розробки інтерактивних інтерфейсів було здійснене її порівняння з інструментами аналогічного

або спорідненого призначення, а саме: JavaScript-фреймворками Angular та Vue (або Vue.js).

Порівняння технологій проводилося за такими характеристиками, як рівень складності, розмір файлу, час завантаження, швидкість розробки, самодостатність при розробці, рік випуску, на яких популярних сайтах використовується. Також наведено адреси офіційних сайтів та версію останнього (поточного на момент написання роботи) релізу.

РОЗДІЛ 3

РОЗРОБКА ІНТЕРФЕЙСУ СИСТЕМИ БРОНЮВАННЯ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ REACT

3.1 Загальні вимоги до інтерфейсу систем бронювання

Використання системи бронювання дозволяє значною мірою покращити якість обслуговування клієнтів, оскільки, забезпечуючи їх бронювання в автоматичному режимі, скорочуються затрати часу на оформлення квитків чи абонементу, а також підвищується якість та ефективність роботи персоналу.

Сьогодні більшість людей віддають перевагу бронюванню послуг у мережі Інтернет, оскільки такий спосіб є більш зручним для клієнтів та має ряд переваг.

З-поміж переваг упровадження системи бронювання в мережі Інтернет для клієнтів можна виділити:

- швидке отримання підтвердження резервування;
- необхідно затратити мінімум часу, щоб оформити резерв;
- є можливість отримання повної інформації стосовно своєї заяви у будь-який момент часу.

Компанії також отримують переваги при впровадженні онлайн-системи бронювання [41]:

- в першу чергу, це скорочення витрат через автоматизацію робочого процесу;
- можливість отримати актуальну інформацію про поточний стан завантаженості системи у будь-який момент часу;
- поліпшується якість обслуговування клієнтів, а саме – заявки клієнтів обробляються більш швидко.

Одним із факторів, який визначає популярність компанії на ринку, виступає час, витрачений на обслуговування клієнтів. У цьому аспекті

беззаперечна перевага віддається обслуговуванню в онлайн-режимі. Саме тому бронювання виступає центральною ланкою у загальній схемі обслуговування клієнтів.

На практиці велика кількість сайтів компаній надають можливість здійснювати резервування по запиту, тобто в оф-лайн режимі. Також таку схему роботи називають псевдоонлайн-бронюванням. Власне, такий принцип роботи – це форма попередньої заявки. Тобто, користувач обирає час, кількість місць, і при цьому вказує контактну інформацію.

Заява, відправлена користувачем, підлягає обробці (тобто затвердженню) з боку співробітника компанії. Як є така потреба, далі він зв'язується з клієнтом, уточнює деталі та здійснює фактичне розміщення броні [29].

Системи бронювання такого типу з-поміж основних своїх переваг мають легкість (простоту) у реалізації та низьку вартість розробки та втілення. Однак при цьому варто враховувати, що процес оформлення резерву за таких умов багато в чому нагадує прийом заявок по телефону.

Автоматизована система бронювання дозволяє вирішити ряд завдань, які зазвичай виникають у процесі резервування:

- уникнення «людського фактору», тобто найрозповсюдженіших помилок персоналу;
- перенавантаження чи, навпаки, простій ресурсів;
- зниження навантаження на співробітників компанії;
- полегшення процесу відслідковування роботи, формування звітів за певний проміжок часу,
- можливість надавати або позбавляти повноважень співробітників для роботи з системою, за необхідності.

Отже, для підтримки популярності сервісу, варто враховувати, що досвід взаємодії з ресурсом є надзвичайно важливим для успіху компанії.

Тобто, щоб клієнти із задоволенням користувалися послугами, які надаються сайтом компанії, зокрема, функцією бронювання як основною,

необхідно, щоб сервіс був простий у використанні, а інтерфейс сайту – зручним та інтуїтивно зрозумілим.

Система бронювання повинна відображати реальну інформацію про:

- поточні тарифи,
- наявність вільних місць,
- можливість здійснення резервування.

З огляду на загальні вимоги до інтерфейсу систем бронювання, можна сформулювати наступні твердження [29; 36; 39]:

1. Інтерфейс системи не повинен бути перевантаженим, оскільки, стикаючись з дуже широким асортиментом функціональних можливостей, користувач змушений самостійно з'ясувати, що ж саме відповідає його потребам. Це негативно впливає на враження від сайту та компанії в цілому.

2. Всі елементи меню або призначеного для користувача інтерфейсу повинні бути добре видимими і інтуїтивно зрозумілими для будь-якого користувача. Організувати зручну для користувача навігацію можна у вигляді вкладеного меню, списку категорій.

3. Якісно опрацьовані елементи навігації повинні бути присутніми на кожній сторінці сайту, щоб у користувача була можливість перейти з будь-якого розділу в необхідний, не здійснюючи при цьому занадто багато зайвих дій. Наприклад, здійснення дзвінка після натискання кнопки, перехід до сторінки компанії в соціальних мережах після натискання відповідної іконки.

4. Елементи навігації повинні бути контрастні фону і відрізнятися від основного тексту, але при цьому гармоніювати із загальним дизайном ресурсу.

5. Мультимедійні дані повинні бути розміщені в такому форматі, який підтримується більшістю сучасних мобільних пристроїв (тобто, слід уникати flash-технологій).

3.2 Реалізація web інтерфейсу системи бронювання

Для реалізації як web, так і мобільної версії інтерфейсу системи бронювання басейну були використані такі технології, як:

- *React* – JavaScript-бібліотека для розробки користувацьких інтерфейсів (UI) [14].
- *Redux* – бібліотека для JavaScript з відкритим вихідним кодом, призначена для управління станом додатку.
- *Redux-Form* – бібліотека Redux, яка була використана для побудови форм.
- *SCSS* – CSS препроцесор для забезпечення модульності, вкладеності та змінних у CSS;
- *Moment.js* – бібліотека для JavaScript для роботи з різними форматами дати і часу.
- *Ant Design* – бібліотека готових React компонентів для розробки користувацьких інтерфейсів [3].
- *Axios* – бібліотека для створення http-запитів на сервер.
- *Webpack* – це інструмент, що дозволяє скомпілювати JavaScript модулі в єдиний JS-файл.

Реалізація web інтерфейсу системи бронювання басейну Херсонського державного університету здійснювалася з урахуванням:

- загальних вимог до системи бронювання;
- вимог до розробки веб-інтерфейсів;
- недоліків попередньої версії.

Основне призначення цього веб-ресурсу – надання інформації про басейн ХДУ у доступній та зрозумілій для користувачів формі та забезпечення можливості здійснювати бронювання в режимі онлайн.

Отже, основна частина сайту працює зі 100% масштабом екрану.

У веб-версії додатку було задано розмір контенту в 1440 пікселів, із внутрішніми відступами у 15 пікселів.

В якості шрифту був використаний Roboto – шрифтова гарнітура без засічок від Google, з базовим розміром у 16 пікселів [15].

Щоб не відволікати увагу користувача від контенту, при створенні дизайну додатку було використано світлі відтінки [психологія цвета]

Меню сайту розміщене горизонтально і лишається незмінним незалежно від сторінки, на якій перебуває користувач. Меню однорівневе, доступне. Пункти мають зрозумілі користувачу назви.

Інформаційна структура сайту – проста і логічна, має лінійний вигляд.

Всього на сайті є такі сторінки:

- Головна,
- Про басейн,
- Розклад,
- Галерея,
- Контакти.

Головна сторінка сайту виглядає наступним чином (Рисунок. 3.1):

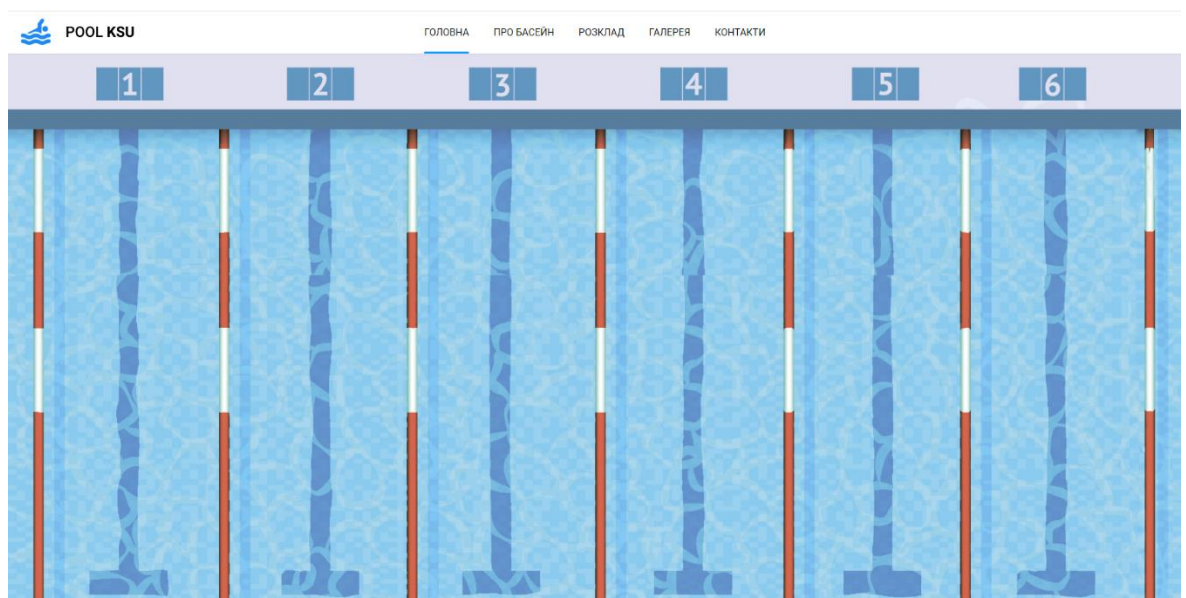


Рисунок. 3.1 - Головна сторінка сайту

Варто зазначити, що функція бронювання реалізована вже на головній сторінці. Доріжки басейну – це інтерактивний елемент, з допомогою якого можна отримати доступ до форми бронювання.

При натисканні на одну з доріжок, з'являється форма бронювання (рис. 3.2). Для її заповнення необхідно вказати наступну інформацію:

- дату відвідування басейну;
- прізвище та ім'я;
- контактний телефон.

ДАТА	ФІО
Оберіть дату	Ім'я та прізвище
ТЕЛЕФОН	ДОРІЖКА
+380 99 99-99-999	4

ЗАБРОНІЮВАТИ

Рисунок. 3.2 - Форма для бронювання доріжки

Номер доріжки у формі пропоставляється автоматично. Це залежить від того, на яку доріжку натиснув користувач.

Для зручності заповнення форми, у полях наведено підказки для користувача – які саме дані та в якому форматі необхідно вносити для коректної роботи системи.

Усі поля є обов'язковими для заповнення. Після внесення всіх даних для продовження роботи варто натиснути кнопку «Забронювати».

Перелік уже зареєстрованих користувачів та їх резервів доступні для перегляду та корекції за допомогою панелі інструментів адміністратора.

На сторінці «Про басейн» передбачене розміщення мультимедійних файлів (фото та відео) про басейн та невеликого обсягу текстової інформації.

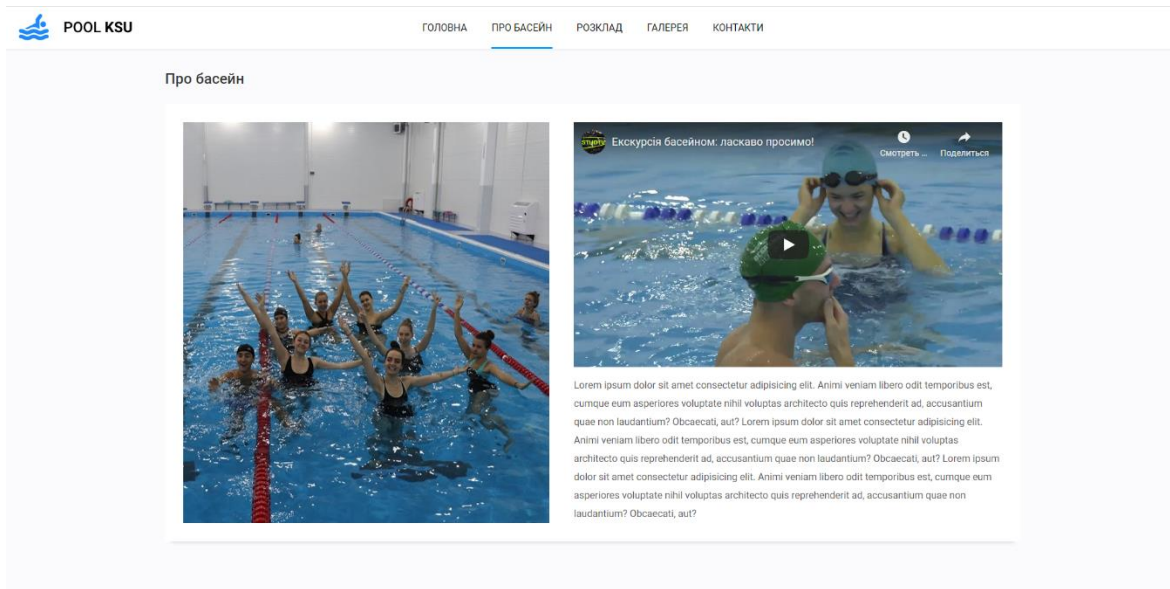


Рисунок. 3.3 - Сторінка «Про басейн»

На сторінці «Галерея» можна переглянути фото з басейну (Рисунок. 3.4).

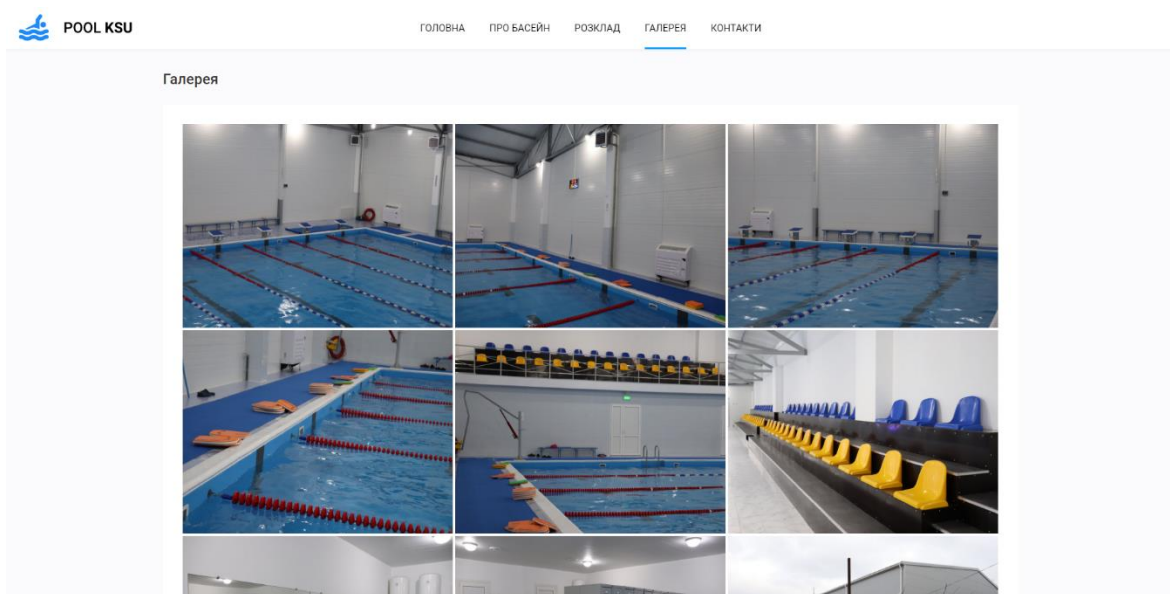


Рисунок. 3.4 - Сторінка «Галерея»

Сторінка «Розклад» надає інформацію про вартість абонементу на певний проміжок часу та кількість відвідувань. У нижній частині сторінки розміщений розклад роботи басейну.

Групи відвідувачів басейну презентовані у вигляді прямокутних блоків, розміщених у хронологічному порядку. В межах блоку вказано назву групи відвідувачів та її час у розкладі.

В один момент часу відображається розклад лише для одного дня тижня. Перемикаючи пункти – дні тижня, можна переглянути розклад роботи басейну для кожного з них. Обраний день виділяється кольором та підкресленням.

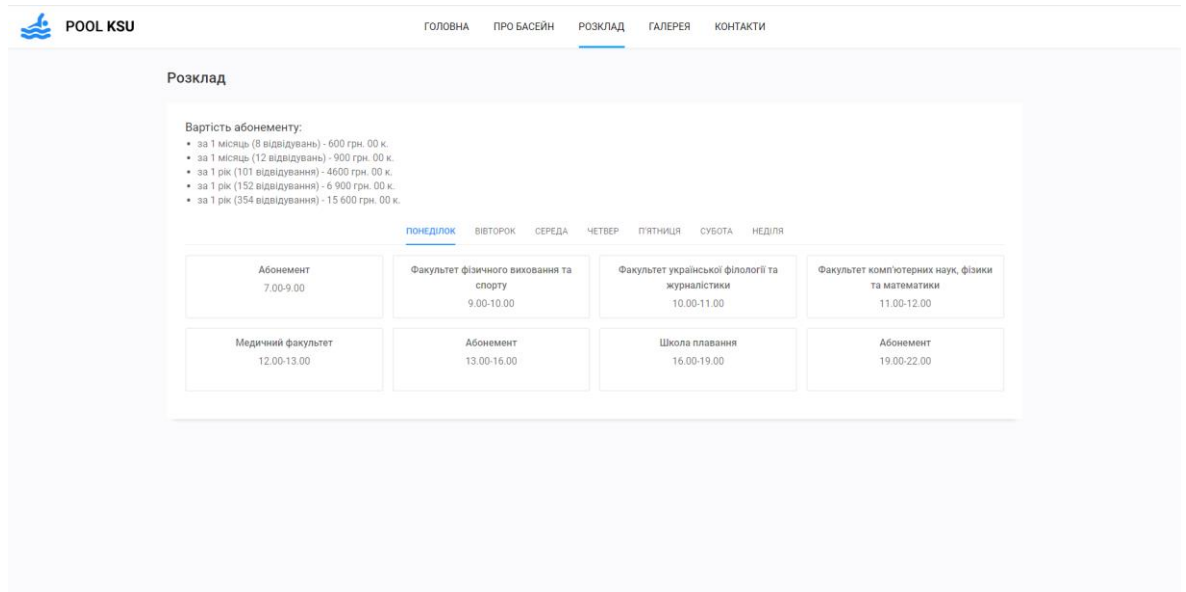


Рисунок. 3.4 - Сторінка «Розклад»

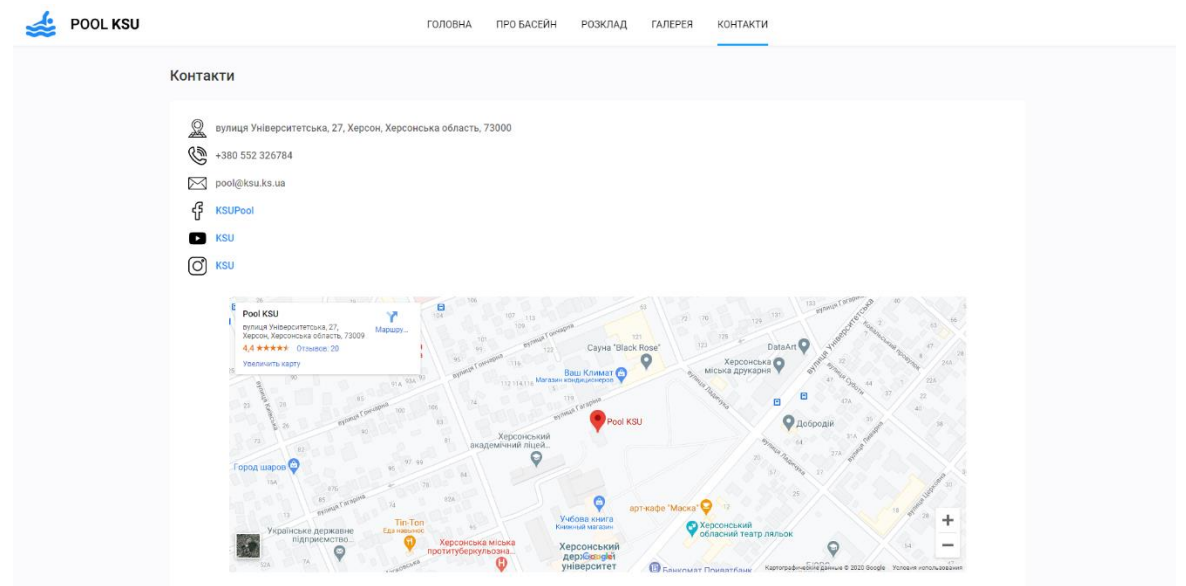


Рисунок. 3.5 - Сторінка «Контакти»

Інформаційна сторінка «Контакти» містить:

- адресу,
- телефон,

- адресу електронної скриньки,
- посилання на сторінки Басейну Херсонського державного університету в таких соцмережах:
 - Facebook,
 - YouTube,
 - Instagram.

На цій же сторінці розміщена інтерактивна карта Google Maps, що полегшить користувачам процес з'ясування, як краще дістатися до басейну.

Отже, реалізований web інтерфейс системи бронювання відповідає загальним вимогам до інтерфейсів систем бронювання та вимогам до розробки web інтерфейсів. Він спроектований та розроблений з урахуванням вимог користувачів сайту для забезпечення максимальної зручності користування системою бронювання басейну.

3.3 Реалізація мобільного інтерфейсу системи бронювання

Реалізація мобільного інтерфейсу системи бронювання басейну була здійснена з використанням набору технологій, зазначених у попередньому підрозділі.

Мобільний інтерфейс було реалізовано з використанням технології адаптивного дизайну. Таким чином, мобільна версія додатку автоматично адаптується під розміри екрану пристрою, на якому користувач переглядає цей сайт.

Під час розробки було враховано, що висока швидкість завантаження вебсторінки у мобільній версії додатку – це важливий параметр для користувачів.

З огляду на це, під час створення мобільної версії додатку було використано зображення у форматах *.jpg та *.svg для оптимізації швидкості завантаження.

Також варто зазначити, що всі зображення та графічні іконки легко адаптуються до різних розмірів екранів мобільних пристроїв.

З метою організації зручної для користувачів навігації у мобільному інтерфейсі, щоб великі меню навігації не займали надто багато місця (значну частину екрану) портативного пристрою, і при цьому не були надміру маленькими, головне меню було створено у вигляді «бургер-меню».

За використання такого способу меню залишається прихованим доти, доки користувач не натисне на нього. Таким чином ми забезпечили можливість роботи головного меню навігації сайту навіть на невеликих екранах.

Як вже було зазначено у розділі 1, застосування цієї технології надає ряд переваг, з-поміж них такі:

- для web та мобільної версій додатку використовується однакова URL-адреса, завдяки чому немає потреби у перенаправленні користувачів;
- структура сайту (набір сторінок та форм) залишається незмінною у порівнянні з web версією.
- зберігається індивідуально розроблений дизайн додатку, зокрема, шрифти, кольорова гама.

Для забезпечення зручності користування мобільним інтерфейсом додатку системи бронювання весь контент додатку має відступи зліва та справа по 15 пікселів. Цим самим забезпечується більша читабельність тексту та наявність місця для здійснення скролінгу, тобто прокрутки.

Отже, реалізований мобільний інтерфейс системи бронювання басейну відповідає загальним вимогам до інтерфейсів систем бронювання та вимогам до розробки мобільних інтерфейсів. Він був спроектований та розроблений з урахуванням вимог до розробки мобільних інтерфейсів з позиції зручності для користувача та збереженням функціональної частини додатку.

Висновки до розділу 3

Отже, до загальних вимог до інтерфейсу системи бронювання можна віднести наступні:

- інтерфейс не повинен бути перевантаженим;
- всі елементи меню та користувацького інтерфейсу повинні бути добре видимими і інтуїтивно зрозумілими;
- у користувача має бути можливість перейти з будь-якого розділу в необхідний, не здійснюючи при цьому занадто багато зайвих дій;
- мультимедійні дані повинні бути розміщені в зручному для більшості сучасних мобільних пристроїв форматі.

Реалізований web інтерфейс системи бронювання відповідає загальним вимогам до інтерфейсів систем бронювання та вимогам до розробки web інтерфейсів. Він спроектований та розроблений з урахуванням вимог користувачів сайту для забезпечення максимальної зручності користування системою бронювання басейну.

Реалізований мобільний інтерфейс системи бронювання басейну був спроектований та розроблений з урахуванням вимог до розробки мобільних інтерфейсів з позиції зручності для користувача та збереженням функціональної частини додатку.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досягнуто поставленої мети, яка полягала у розробці інтерфейсу системи бронювання з використанням технологій React.

У ході виконання роботи були виконані наступні завдання:

1. Визначено основні принципи створення інтерфейсу додатків. Користувальницький інтерфейс – це сукупність засобів, призначених для обробки та відображення інформації, що якнайкраще пристосовані для зручності використання користувачем. Створення інтерфейсу додатків підпорядковується певним принципам, зокрема елементи інтерфейсу повинні бути організованими, об'єднаними у групи логічно пов'язаних елементів, вирівняними, оформленими у єдиному стилі, і, крім того, обов'язковим є наявність вільного простору для розмежування інформаційних блоків, щоб зосередити увагу користувача на чомусь одному.

Якщо інтерфейс розроблений за всіма правилами, то це значно підвищує ефективність ресурсу і робить його конкурентноспроможним.

2. Проаналізовано особливості і вимоги до розробки web та мобільного інтерфейсів. Здійснено аналіз поняття UI/UX дизайну сайтів та вимог до нього, виконано огляд основних принципів проєктування дизайну додатків з позиції UI/UX-дизайнерів (дизайнерів інтерфейсів). Розглянуто особливості розробки web інтерфейсів, наведено основні вимоги до дизайну web версії сайту, зокрема, для таких елементів, як сітка, розміри, ширина контенту, зображення та іконки, тексти і шрифти. Виокремлено важливі параметри, які враховуються при розробці мобільних інтерфейсів з огляду на динамічність роботи з ними. Було виконано огляд різних підходів до адаптації вже існуючого сайту під мобільні пристрої, їх сильних та слабких сторін: створення мобільної версії, застосування технології динамічного показу, технології

адаптивного дизайну.

3. Розглянуто бібліотеку React як інструмент розробки інтерактивного інтерфейсу. React – популярна JavaScript-бібліотека для роботи з користувацькими інтерфейсами (UI). До ключових особливостей React належать декларативність, універсальність, компонентний підхід, віртуальний DOM, JSX.

Для визначення переваг React для розробки інтерактивних інтерфейсів було здійснене її порівняння з інструментами аналогічного або спорідненого призначення, а саме: JavaScript-фреймворками Angular та Vue (або Vue.js).

4. Виконано аналіз вимог до інтерфейсу системи бронювання. Загальні вимоги до інтерфейсу системи бронювання можна сформулювати наступним чином: інтерфейс не повинен бути перевантаженим; всі елементи меню та користувацького інтерфейсу повинні бути добре видимими і інтуїтивно зрозумілими; у користувача має бути можливість перейти з будь-якого розділу в необхідний за обмежену кількість кроків; мультимедійний контент повинен бути розміщеним у зручному для більшості сучасних мобільних пристроїв форматі.

5. Реалізовано web та мобільний інтерфейси системи бронювання за допомогою технології React. Перераховано, які технології були використані для реалізації як web, так і мобільної версії інтерфейсу системи бронювання басейну. Реалізовані інтерфейси системи бронювання відповідають загальним вимогам до інтерфейсів систем бронювання та вимогам до розробки відповідно web та мобільного інтерфейсів.

Результатом роботи є створення web та мобільного інтерфейсів системи бронювання басейну ХДУ з використанням технологій React.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 16 основних требований к дизайн-макету сайта: URL: <https://www.proofsites.com.ua/article-330.html>
2. A Beginner's Guide to User Experience (UX) Fundamentals/ URL: <https://generalassemb.ly/design/visual-design/user-experience-fundamentals>.
3. Ant Design: A design system for enterprise-level products. URL: <https://ant.design>
4. Beaird J. The Principles of Beautiful Web Design. Jason Beaird Melbourne: SitePoint Pty Ltd, 2010. 77 p.
5. Cooper A. About Face: The Essentials of Interaction Design. URL: <https://fall14se.files.wordpress.com/2017/04/about-face-3-the-essentials-of-interaction-design.pdf>
6. Curricula for Human-Computer Interaction, Chapter 2. Definition and Overview of Human-Computer Interaction. URL: <http://www.dainf.cefetpr.br/~merkle/thesis/CH2.pdf>
7. Duckett J. HTML and CSS: Design and Build Websites. Hoboken, New Jersey: John Wiley & Sons, 2011. 512 p.
8. Front-end розробка інтерфейсу користувача. UX-дизайн. Проектування інтерфейсів користувача програмних систем. Хмельницький національний університет, 2019. https://msn.khnu.km.ua/pluginfile.php/420851/mod_resource/content/1/Лекція6.pdf
9. Garrett J. J. The Elements of User Experience: User-Centered Design for the Web and Beyond. London: Pearson Education, 2010. 79 p.
10. Gordon Z. React Explained: Your Step-by-Step Guide to React (2020 Edition). OS Training, LLC, 2019. 371 p.
11. International Standards. *UsabilityNet*. URL: <http://www.usabilitynet.org/trump/resources/standards.htm>

12. Kiess C. A Practical Guide to UX Specifications. URL: <http://chriskiess.net/a-practical-guide-to-ux-specifications/>
13. Marcus A. Design, User Experience, and Usability. *Design Discourse*. 2015. P. 340.
14. React – JavaScript-библиотека для создания пользовательских интерфейсов. URL: <https://ru.reactjs.org>
15. Roboto: Google Fonts. URL: <https://fonts.google.com/specimen/Roboto>
16. Sidelnikov G. React.js Book: Learning React JavaScript Library. 2017. 117 p.
17. The Fundamentals of Great UX. URL: <https://uxdesign.cc/development-fundamentals-for-ux-designers-caf759724874>.
18. Treder M. Beyond Wireframing: The Real-Life UX Design Process. Smashing Magazine. URL: <https://www.ceros.com/resources/beyond-wireframing-real-life-ux-design-process/>
19. UI vs UX: в чем различие. URL: <https://lpgenerator.ru/blog/2016/06/23/ui-vs-ux-v-chem-razlichie/>
20. UX Curve: A method for evaluating long-term user experience / Kujala, S. and other. *Interacting with Computers*. 23 (5). P. 473-483.
21. Visual Design Web Style Guide. *Web Style Guide*: web-site. URL: <https://webstyleguide.com/wsg3/7-page-design/3-visual-design.html>.
22. Web UI Best Practices: Free e-book. URL: <https://speckyboy.com/free-e-book-web-ui-best-practices/>.
23. Weinschenk S. 100 Things Every Designer Needs to Know About People. San Francisco: New Riders, 2011. 256 p.
24. What's the Difference Between a User Experience (UX) Designer and a User Interface (UI) Designer? URL: <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/>
25. Wieruch R. The Road to learn React: Your journey to master plain yet pragmatic React.js. CreateSpace Independent Publishing Platform, 2018.

208 p.

26. Антонов В.М. Сучасні комп'ютерні мережі. К.: МК-Прес, 2015. 480с.
27. Гаевский А.Ю. Самоучитель по созданию Web-страниц: HTML, JavaScript, Dynamic HTML. К.: А.С.К., 2002. 464 с.
28. Из чего состоит сайт или основные структурные блоки сайта. URL: <https://www.inetgramotnost.ru/sajt/iz-chego-sostoit-sajt-ili-osnovnye-strukturnye-bloki-sajta.html>
29. Интерфейс. Основы проектирования взаимодействия / А. Купер, Р. Рейман, Д. Кронин, К. Носсел. Санкт-Петербург: Питер, 2018. 720 с.
30. Как начать работать UX дизайнером: 10 навыков. *Igor Yazov Blog*: веб-сайт. URL: <http://www.blog.jazov.com/uiux-design/kak-nachat-rabotat-ux-dizajnerom-10-navykov.html>
31. Кедлек Т. Адаптивный дизайн. Делаем сайты для любых устройств. Санкт-Петербург: Питер, 2013. 288 с.
32. Крейг Дж. Р., Скала И. К. Шрифт и дизайн. Современная типографика. Санкт-Петербург: Питер, 2016. 176 с.
33. Крокфорд Д. Как устроен JavaScript. Санкт-Петербург: Издательский Дом «Питер», 2018. 304 с.
34. Круг С. Как сделать сайт удобным. Юзабилити по методу Стива Круга. Санкт-Петербург: Питер, 2010. 170 с.
35. Мандел Т. Разработка пользовательского интерфейса. М.: ДМК Пресс, 2019. 416 с.
36. Маркотт И. Отзывчивый веб-дизайн. Москва: «МИФ», 2012. 176 с.
37. Нильсен Я., Лоранжер Х. Web-дизайн: зручність використання Web-сайтів. М.: «Вільямс», 2007. 368 с.
38. Норман Д. Дизайн привычных дверей. Москва: Издательский дом "Вильямс", 2006. 222 с.
39. Покришень Д.А. Технологічні принципи проектування

інформаційних систем. *Сборник научных трудов SWorld*. 2014. Вып. 1. Том 17. С. 37-44.

40. Порівняння UX та UI дизайнерів. URL: <https://internetdevels.ua/blog/ux-designer-vs-ui-designer>
41. Самсонов В.В., Єрохін А.Л. Методи та засоби Інтернет-технологій: Навч. посібник. Харків: СМІТ, 2008. 263 с.
42. Торрес Р.Дж. Практическое руководство по проектированию и разработке пользовательского интерфейса. Пер. с англ. – М.: Издательский дом Вильямс, 2002. – 401 с.
43. Требования к дизайн макетам сайта. URL: <https://flagstudio.ru/blog/trebovania-k-dizain-maketam-saitov>
44. Гудоран В.А. Людино – машинна взаємодія.- Миколаїв: Вид-во НУК, 2015.- 164 с.
45. Унгер Р. UX-дизайн. Практическое руководство по проектированию опыта взаимодействия. Санкт-Петербург: «Символ-Плюс», 2011. 27 с.
46. Уэйншенк С. 100 главных принципов дизайна. Санкт-Петербург: Питер, 2012. 272 с.
47. Чопа Д.А. Деякі аспекти створення та застосування мобільних додатків для вивчення військово-технічних дисциплін. *Interactive Models of Scientific Educational Environment Development in the Sphere of Security and Defence*. 2020. 1(37). С. 151-156.
48. Что такое UX и UI дизайн – особенности и отличия. URL: <http://www.kasper.by/help/chto-takoe-ux-i-uidizain/>

ДОДАТКИ

Додаток А

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО
ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Кравченко Олександр Олександрович,
учасник(ця) освітнього процесу Херсонського державного університету, УСВІДОМЛЮЮ, що академічна
добročесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

- дотримуватися:
 - вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
 - принципів та правил академічної доброчесності;
 - нульової толерантності до академічного плагіату;
 - моральних норм та правил етичної поведінки;
 - толерантного ставлення до інших;
 - дотримуватися високого рівня культури спілкування;
- надавати згоду на:
 - безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
 - оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
 - використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;
- самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
 - надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
 - не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
 - своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
 - не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
 - підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
 - поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
 - не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
 - відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;
 - запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
 - не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
 - не підроблювати документи;
 - не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
 - не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
 - не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
 - не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
 - не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
 - не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;
 - не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

05.05.2020
(дата)

[Підпис]
(підпис)

Кравченко О.
(ім'я, прізвище)