

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Медіасистем та технологій
Рівень вищої освіти другий (магістерський)
Спеціальність 186 Видавництво та поліграфія
Тип програми Освітньо-професійна
Освітня програма Технології електронних мультимедійних видань
(шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри МСТ _____
(підпис)

« 26 » жовтня 2020 р.

**ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ**

студентові Коміній Марині Максимівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка методики ефективного застосування анімації у WEB

Затверджена наказом по університету від 23 жовтня 2020р. № 1432 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 14 грудня 2020 р.

3. Вихідні дані до роботи

Методи та технології застосування анімації у WEB

Методи математичної статистики

Методи UML-моделювання

4. Перелік питань, що потрібно опрацювати в роботі

Вступ

1 Аналітичний огляд досягнень використання анімацій у веб-технологіях

2 Дослідження інструментів та методик створення веб-анімацій

3 Розробка методики ефективного застосування анімації у web

4 Практична реалізація розробленої методики

5 Проведення тестування розробленої методики

6 Економічна частина

Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

Мета і основна ціль роботи; Об'єкт та предмет дослідження; Аналітичний огляд досягнень; Дослідження інструментів та методик; Розробка методики; Практична реалізація; Проведення тестування; Економічна частина; Висновки.

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Єгорова І.М.		
Економічна частина	проф. Полозова Т.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналітичний огляд досягнень використання анімацій у веб-технологіях	26.10.20	
2	Дослідження інструментів та методик створення веб-анімацій	31.10.20	
3	Розробка методики ефективного застосування анімації у web	07.11.20	
4	Практична реалізація розробленої методики	15.11.20	
5	Проведення тестування розробленої методики	30.11.20	
6	Економічна частина	04.12.20	
7	Оформлення пояснювальної записки	08.12.20	
8	Оформлення графічної частини	11.12.20	

Дата видачі завдання 26 жовтня 2020 р.

Студент _____ Коміна М.М.
(підпис)

Керівник роботи _____ проф. Єгорова І.М.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 58 сторінок, 25 рисунків, 12 таблиць, 31 літературні джерела, 2 додатка.

ВЕБ-САЙТ, АНІМАЦІЯ, ВЕБ-ТЕХНОЛОГІЇ, JAVASCRIPT, CSS, MOTION UI, ДОСЛІДЖЕННЯ.

Мета атестаційної роботи – дослідження сучасних методик та інструментів розробки анімації веб-ресурсів, а також створення на основі проведеного дослідження методики, яка дозволить підвищити ефективність розробки веб-анімації, в тому числі за рахунок скорочення часу, який буде потрібен розробнику веб-видань для роботи з анімацією, та скорочення часу завантаження анімації, розробленої за даною методикою.

Об'єкт дослідження – процес створення веб-анімацій для різноманітних продуктів.

У даній атестаційній роботі були досліджені різні методи створення анімацій для веб-продуктів, були розглянуті плюси і мінуси їх використання в веб-розробці.

На основі проведеного дослідження була створена методика розробки веб-анімації. Ефективність запропонованої методики була доведена в ході проведеного експерименту за допомогою експерта і фокус-групи.

Також виконано розрахунок витрат на розробку методики, визначена ціна проведення дослідження.

РЕФЕРАТ

Пояснительная записка содержит 58 страниц, 25 рисунков, 12 таблиц, 31 литературный источник, 2 приложения.

САЙТ, АНИМАЦИЯ, ВЕБ-ТЕХНОЛОГИИ, JAVASCRIPT, CSS, MOTION UI, ИССЛЕДОВАНИЕ.

Цель аттестационной работы – исследование современных методик и инструментов разработки анимации веб-ресурсов, а также создание на основе проведенного исследования методики, которая позволит повысить эффективность разработки веб-анимации, в том числе за счет сокращения времени, которое потребуется разработчику веб-изданий для работы с анимацией, и сокращения времени загрузки анимации, разработанной по данной методике.

Объект исследования – процесс создания веб-анимации для различных продуктов.

В данной аттестационной работе были исследованы различные методы создания анимаций для веб-продуктов, были рассмотрены плюсы и минусы их использования в веб-разработке.

На основе проведенного исследования была создана методика разработки веб-анимации. Эффективность предложенной методики была доказана в ходе проведенного эксперимента с помощью эксперта и фокус-группы.

Также произведен расчет затрат на разработку методики, определена цена проведения исследования.

ABSTRACT

Explanatory note contains: 58 pages, 25 pictures, 12 tables, 31 literature sources, 2 applications.

SITE, ANIMATION, WEB TECHNOLOGIES, JAVASCRIPT, CSS, MOTION UI, RESEARCH.

The purpose of the certification work is to study modern techniques and tools for the development of animation of web resources, as well as to create, based on the research, a methodology that will improve the efficiency of the development of web animation, including by reducing the time that a developer of web publications will need to work with animation, and reducing the loading time of the animation developed using this technique.

The object of research is the process of creating web animations for various products.

This certification work explored various techniques for creating animations for web products and looked at the pros and cons of using them in web development.

Based on the research, a methodology for the development of web animation was created. The effectiveness of the proposed method has been proven in the course of the experiment with the help of an expert and a focus group.

Also, the calculation of the costs of developing the methodology was made, the price of the implementation of the study was determined.

ЗМІСТ

С.

1	АНАЛІТИЧНИЙ ОГЛЯД ДОСЯГНЕНЬ ВИКОРИСТАННЯ АНІМАЦІЙ В ВЕБ-ТЕХНОЛОГІЯХ	11
1.1	GIF-анімація	12
1.2	Flash-анімація	13
1.3	Сучасні технології для створення веб-анімації	13
1.4	Постановка завдання дослідження	14
2	ДОСЛІДЖЕННЯ ІНСТРУМЕНТІВ ТА МЕТОДИК СТВОРЕННЯ ВЕБ-АНІМАЦІЇ	15
2.1	CSS-анімація	15
2.2	SVG	16
2.3	Canvas-анімації	17
2.4	WebGL	18
2.5	JavaScript-анімація	18
2.6	jQuery	19
2.7	Motion UI	19
3	РОЗРОБКА МЕТОДИКИ ЕФЕКТИВНОГО ЗАСТОСУВАННЯ АНІМАЦІЇ У WEB	20
3.1	Формулювання гіпотези відповідно до мети роботи	20
3.2	Створення методики ефективного застосування анімації у WEB	21
4	ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОЇ МЕТОДИКИ	27
5	ПРОВЕДЕННЯ ТЕСТУВАННЯ РОЗРОБЛЕНОЇ МЕТОДИКИ	36
5.1	Проведення тестування з експертом	37
5.2	Дослідження часу завантаження веб-анімації	41
6	ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЕКТУ	45
6.1	Характеристика науково-дослідного рішення	45
6.2	Етапи виконання НДР, їх трудомісткість та заробітна плата	45
6.3	Розрахунок одноразових витрат на розробку НДР	48
6.4	Оцінка результатів науково-дослідної роботи	52

ВИСНОВКИ	55
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	56
ДОДАТОК А HTML-код та каскадна таблиця стилів основної сторінки для анімованого об'єкту	59
ДОДАТОК Б Приклади кодів реалізації анімаційних ефектів	62

ВСТУП

Розвиток нових технологій та високотехнологічного обладнання, зокрема різноманітної комп'ютерної техніки, призвів до створення нового інформаційно-комунікаційного середовища – глобальної мережі Інтернет, яка на даному етапі розвитку суспільства є найважливішою частиною його життя, ефективним середовищем для здійснення масової комунікації. На сьогоднішній день кількість користувачів мережі Інтернет досягло кількості в 5 млрд інтернет-користувачів, і з кожним днем це число стає більше [1]. Сучасний Інтернет наповнений величезною кількістю різних веб-сайтів на різноманітну тематику [2].

Популярність Інтернету, як джерела інформації, обумовлена тим, що він відрізняється від колишніх інформаційно-комунікаційних середовищ важливими особливостями, в першу чергу розширеними можливостями в поданні, оновленні та оформленні інформації. Тому на сьогоднішній день більшість фірм, інформаційних ресурсів, організацій та підприємств переходять від використання різних друкованих видань та інформаційних ресурсів к використанню різноманітних веб-ресурсів та електронних видань. Користувачі постійно шукають в всесвітній мережі нові додатки і веб-сайти, які дозволяють зробити життя простіше, а також надають різну інформацію, навіть ту яка без Інтернету була б недоступна для широких мас [3]. Веб-ресурси розробляються і використовуються для різноманітних цілей: залучення користувачів, залучення потенційних клієнтів, інформування співробітників фірми і утримання уваги цільової аудиторії на веб-виданні.

Стрімкий розвиток Інтернету та веб-ресурсів тягне за собою аналогічний розвиток різних сфер пов'язаних з оформленням і розробкою веб-продуктів [4]. В даний час на зміну традиційним методам розробки, приходять все нові, найчастіше більш ефективні методи [5]. Всі інновації спрямовані в основному на головну мету кожного веб-видання – надати інформацію користувачам у

зручному і привабливому вигляді. Одним з розділів веб-розробки, який дозволяє досягти описаної мети, є використання анімації в веб. З кожним днем веб-анімація стає все більш популярною. Анімація стала одним з головних інструментів для досягнення юзабіліті [6].

Жоден веб-ресурс не може обійтися без анімації, і пояснюється це не тільки бажанням користувача працювати з комфортом, але і необхідністю візуально відстежувати всі дії, які користувачі можуть здійснювати на даному веб-ресурсі. Для створення якісної анімації розробнику потрібно провести за роботою достатньо часу і іноді це сильно збільшує час роботи над проектом. Незважаючи на стрімкий розвиток галузі веб-розробок, створення веб-ресурсів все ще залишається досить дорогою роботою для замовників. Виходячи з цього існує необхідність в створенні методики ефективного застосування анімації у веб. Доцільно розробити методику, яка враховує всі необхідні фактори для створення якісної анімації [7].

Актуальність даної роботи полягає в тому, що розробка методики використання анімації в веб допоможе у вирішенні проблеми ефективного використання анімації під час розробки веб-проекту та скорочення часу завантаження розробленої анімації. Скорочення часу роботи з веб-анімаціями та процесу розробки веб-проекту, цілком пройшовши всі етапи, допоможе оптимізувати виробництво для ІТ-компаній і веб-розробників. Скорочення часу завантаження дозволить поліпшити якість розроблених анімаційних ефектів.

Основною метою дослідження є розробка ефективної методики створення анімації для веб-продуктів. В результаті використання запропонованої методики очікується поліпшення процесу розробки за певними розглянутими параметрами.

1 АНАЛІТИЧНИЙ ОГЛЯД ДОСЯГНЕНЬ ВИКОРИСТАННЯ АНІМАЦІЙ В ВЕБ-ТЕХНОЛОГІЯХ

Основна мета будь-якого веб-продукту – це залучити й утримати увагу користувачів [8]. На сьогоднішній момент, це стає дуже складним завданням, так як повсякденне життя звичайної людини наповнена нескінченним потоком інформації і різноманітної візуальної складової різних продуктів, починаючи з анімованих оголошень на вулицях міста, відеороликів на мобільних пристроях і закінчуючи сторінками в соціальних мережах, постійний рух на яких посилено повертає увагу кінцевого споживача. Рухомі елементи повертають додаткову увагу користувачів, так як люди інстинктивно реагують на рухи навколо себе. Така реакція на рух в природі з'явилася внаслідок потреби вижити та спричинена стимулами навколишнього середовища, які сприймаються як потенційно загрозливі чи шкідливі для людини.

Ці знання стимулюють веб-розробників до використання анімації в своїх проєктах, що дозволяє виконати дві головні завдання для розробника: залучити користувача до продукту і передати інформацію.

Веб-анімація – це технічні прийоми створення ілюзії рухомих зображень, які використовуються в роботі з різними веб-проєктами [9]. Веб-анімація може залучати та утримувати увагу людей довше, ніж статичні зображення, і може передати ідею чи концепцію веб-сайту більш чітко і ефективно, передати історію кожного руху користувача в системі.

Веб-анімація повинна бути плавною, змістовною і допомагати відвідувачеві в подорожі по веб-продукту. Розробник повинен знати, як анімація вписується в дії призначені для користувача, намагатися передбачити можливі його кроки, а потім підтримувати їх значущим чином.

Веб-розробники не повинні дивитися на веб-анімацію з чисто технічної точки зору – їм потрібно дивитися на це з точки зору користувача. Одна з

важливих речей в анімації – це час. Правильний вибір часу надає анімації фізичний і емоційний сенс. Досвід повинен бути плавним і логічним. Якщо анімація не плавна (є проблеми з синхронізацією), люди можуть сприйняти це як помилку і втратити будь-яку мотивацію для подальшого вивчення веб-сайту [10].

На початку становлення всесвітньої мережі Інтернет використання анімації було дуже складним і трудомістким процесом. Більшість веб-продуктів були статичними і базувалися виключно на графічному дизайні та на стилістиці друкованих продуктів. Однак розробники доклали багато зусиль щоб, незважаючи на обмежені можливості, включити ранні форми веб-анімації в продукти, щоб зробити їх більш динамічними та привабливими.

1.1 GIF-анімація

Отже на самому початку розвитку індустрії веб-розробники активно використовували GIF-анімацію. GIF-файли підходять для коротких циклів анімації, послідовностей зображень і навіть коротких циклів відео. На жаль, недоліком є те, що формат файлу GIF не має змінної прозорості і не підтримує альфа-канал; отже, всі пікселі або повністю непрозорі, або повністю прозорі.

GIF-анімації були початком зародження анімації, але вони не були ідеальними. GIF-файли сильно обмежували пропускну здатність, особливо за часів повільної швидкості Інтернету. В результаті даний тип анімації давав розробникам тільки нескладну піксельну послідовність з низьким дозволом. Основним недоліком GIF-анімації була обмежена 8-бітна палітра, що призводило до великої згладжування [11]. Ситуація змінилася, оскільки високошвидкісний Інтернет став більш звичним явищем в 21 столітті, і, як наслідок, веб-анімація стала краще виглядати з мільйонами кольорів і більш плавною з більш високою частотою кадрів.

1.2 Flash-анімація

Наступним етапом розвитку веб-анімації стала Flash-анімація.

Flash-анімація легка і відносно проста в створенні [12]. Їх розмір такої анімації становить всього кілька кілобайт, вона поширюються в форматі файлу SWF та також файлу зі звуком з чіткою векторною графікою. Створення веб-анімації стало спрощеним процесом, який не сильно збільшував час завантаження веб-сторінки. Але при цьому залишався істотний недолік – для його роботи потрібний плагін браузера. Складна взаємодія з Flash-анімацією, відсутність реакції на запити користувачів, неможливість роботи на всіх можливих пристроях привело до того, що популярність Flash-анімації сильно впала і це призвело до того що більшість веб-розробників відмовилися від використання Flash-анімації. Крім того описані вище типи анімації дуже складно було використовувати без участі веб-дизайнерів і вони підключалися до проектів окремо.

1.3 Сучасні технології для створення веб-анімації

Сьогодні, коли Flash застарів, у нас інші потреби в веб-анімації. Інструменти повинні бути гнучкими і легкими. Веб-розробники повинні створювати адаптивні проекти для різних пристроїв (настільні, планшетні та мобільні), беручи до уваги різні розміри екранів, різноманітні браузери, пропорції, щільність пікселів, і багато іншого.

Веб-розробники також повинні враховувати кросплатформову підтримку [13]. Десять років тому розробникам не потрібно було дбати про те, щоб їх творіння добре виглядали на такому широкому спектрі пристроїв. Необхідно враховувати різні співвідношення сторін, портретну і ландшафтну орієнтацію, а також різну щільність пікселів і відстань перегляду. Ця загадка ставить ряд нових проблем і потенційних пасток в розробці.

Сучасні технології для розробки веб-продуктів такі як HTML5, CSS3, JavaScript є кращими рішеннями для веб-анімації. Використання цих сучасних веб-технологій і мов допоможе вирішити більшість з перерахованих вище проблем. Однак великий розкид в технологіях може заплутати розробника під час роботи, так як кожна із запропонованих технологій має свої плюси і мінуси, і є потреба в розгляді всіх актуальних варіантів технологій і створення оптимальної методики для розробки веб-анімацій.

1.4 Постановка завдання дослідження

Метою роботи є дослідження сучасних методик та інструментів розробки анімації веб-ресурсів, а також створення на основі проведеного дослідження методики, яка дозволить підвищити ефективність розробки веб-анімації, в тому числі за рахунок скорочення часу, який буде потрібен розробнику веб-видань для роботи з анімацією, та скорочення часу завантаження анімації, розробленої за даною методикою.

Об'єктом роботи є процес створення веб-анімацій.

Предметом є методика ефективного застосування анімації у веб.

Практичне значення атестаційної роботи полягає у можливості застосування результатів розглянутої роботи для використання анімації під час розробки веб-проекту.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- здійснити аналітичний огляд досягнень в області створення анімацій веб-сайтів;
- досліджувати інструменти і методики створення веб-анімації для проектів;
- визначити проблеми створення анімацій в веб-індустрії;
- визначити критерії ефективності розробки анімації;
- здійснити практичну реалізацію методики;
- провести порівняльний аналіз ефективності методик створення анімації;
- провести економічне обґрунтування роботи.

2 ДОСЛІДЖЕННЯ ІНСТРУМЕНТІВ ТА МЕТОДИК СТВОРЕННЯ ВЕБ-АНІМАЦІЇ

З кожним днем веб-анімація стає все більш популярною. Жодне електронне видання не може обійтися без анімації, і пояснюється це не тільки бажанням користувача працювати з комфортом, але і необхідністю візуально відстежувати всі події. Елементи анімації в інтерфейсах виносять їх юзабіліті на новий рівень, роблячи їх інтуїтивно зрозумілими і доступними звичайному користувачеві [14].

Аналітичний огляд інструментів і методик створення веб-анімації здійснюється в роботі з метою виявлення нюансів, з якими стикається веб-розробник під час роботи з анімацією, при створюванні електронних видань. Згідно ДСТУ 7157:2010 «Електронні видання. Основні виходи і вихідні відомості», електронне видання – електронний документ (група електронних документів), який пройшов редакційно-видавниче опрацювання, призначений для поширення в незмінному вигляді, має вихідні відомості [15].

Проведений аналіз результатів дослідження покладено в основу розробки ефективної методики створення веб-анімацій, яка дозволить не тільки заощадити витрачений на розробку анімацій час, а й отримати в результаті якісний веб-продукт. Слід послідовно проаналізувати основні технології, з якими може зіткнутися веб-розробник при розробці анімацій веб-інтерфейсу. Широке поширення анімації тягне за собою і розвиток технологій, кожна з яких має свої сильні і слабкі сторони. У зв'язку з цим вибір ефективної технології для створення анімації в веб-додатках представляється завданням актуальною.

2.1 CSS-анімація

Першим з розглянутих способів створення анімації є *native animation*. Анімаційні ефекти в цьому випадку реалізуються за допомогою ряду

технологій, і зокрема, CSS- каскадних таблиць стилів. CSS створений в першу чергу для того, щоб відокремити дизайн веб-сторінок від їх верстки [16]. Анімація CSS дозволяє переходити між різними станами за допомогою ключових кадрів. За допомогою CSS веб-розробник може передавати значення ключових кадрів та властивостей синхронізації, а не створювати об'єкти для ключових кадрів та властивостей синхронізації. Високопродуктивна анімація CSS допоможе виключити необхідність використовувати різні зовнішні бібліотеки.

Елементи веб-сайту можна анімувати за допомогою об'єктної моделі документа (DOM). CSS також хороший варіант для адаптивного розвитку веб-продукту, оскільки анімацію можна змінювати за допомогою медіа-запитів.

Аналізуючи описану вище інформацію можна дійти такого висновку, що серед незаперечних переваг каскадних таблиць стилів можна зазначити наступні [17]:

- простоту зміни дизайну безлічі сторінок;
- можливість зміни анімації і реалізації адаптивного дизайну за допомогою медіа-запитів;
- відсутність необхідності використання зовнішніх бібліотек.

Одним з основних недоліків CSS є різне відображення веб-документів різними браузерами, що особливо характерно для застарілих браузерів.

Також недоліком даного методу є те, що за допомогою даної технології дуже складно створювати складні ефекти та копіювати реалістичні рухи. Крім цього чим більше буде розроблено веб-анімацій для продукту, тим складніше дотримуватися реалістичності в анімації.

2.2 SVG

CSS використовується також для анімації разом з форматом файлу SVG. SVG розшифровується як масштабована векторна графіка. Анімація SVG виглядає чітко, оскільки вектори не мають обмежень у пікселях [18].

Незалежно від того, як ви змінюєте розмір сторінки, SVG підтримують якість, як і растрові зображення.

Як було описано вище, елементи SVG можна анімувати за допомогою CSS. Але SVG також має власний синтаксис для анімації, який називається SMIL. SMIL є потужнішим за CSS для анімації SVG, оскільки з CSS ви не можете анімувати деякі властивості SVG, такі як форма контуру.

2.3 Canvas-анімації

Останнім часом для створення веб-анімації використовують елемент Canvas, який фактично надає доступ до бібліотек функцій для малювання 2D-графіки. Canvas – це растрове полотно, яке дозволяє генерувати графіком, статичні зображення, діаграми і анімацію, а також передбачає використання JavaScript [19]. Розробники веб-продуктів за допомогою анімації Canvas може анімувати велику кількість візуальних об'єктів одночасно. Також Canvas дозволяє створювати якісну та складну анімацію на візуальному просторі, пропонованому полотном.

Canvas-анімації використовують в роботі пікселі і тому дана технологія є ідеальним способом створення вдосконаленої анімації. Використання Canvas під час розробки веб-продукту допомагає створювати складні анімації без шкоди для роботи проекту.

Використання Canvas-анімації має ряд переваг, а саме:

- стабільна продуктивність;
- відсутність кордонів для створення об'єктів і анімацій;
- збереження продуктивності при промальовуванні найскладнішої анімації.

Недоліком технології є необхідність постійного перемальовування кожного елемента в інтерфейсі.

2.4 WebGL

Окремо слід розглянути WebGL – бібліотека веб-графіки, яка забезпечує API 3D графіки без використання сторонніх плагінів [20]. WebGL повністю інтегрований в усі стандарти, що дозволяє використовувати апаратне прискорення для обробки зображень і ефектів на полотні веб-додатка.

WebGL також використовувався для створення анімації для віртуальної реальності, а також для відтворення графіки зі швидкістю 60 кадрів в секунду. Найбільш креативні та інноваційні візуальні ефекти зроблені за допомогою WebGL. Використовуйте полотно для анімації WebGL, але це може бути досить складно. У наш час багато розробників веб-сайтів застосовують WebGL для створення інноваційних та привабливих візуальних ефектів. Цей веб-стандарт має великий каталог візуальних ефектів, а 3D графіка значно розширює можливості для взаємодії.

Складність роботи з веб-стандартом пояснюється складністю його вивчення. Також дану технологію найбільш оптимально використовувати для реалізації складних ефектів та 3D-об'єктів.

2.5 JavaScript-анімація

Анімація JavaScript пропонує більше потужності та гнучкості, ніж переходи або анімація на основі CSS, а також може використовуватися для переміщення елементів DOM на сторінці [21]. Базовий JavaScript має власну функцію анімації, але більшість анімацій виконуються за допомогою додаткових бібліотек, які суттєво впливають на продуктивність. Анімація JavaScript широко використовується для анімації підстрибування, призупинення, зупинки та уповільнення. Ефективність JavaScript залежить від обраної бібліотеки.

2.6 jQuery

Другим способом створення анімації є підключення зовнішніх бібліотек. jQuery – це кросплатформова JavaScript-бібліотека, призначена для спрощення клієнтського сценарію HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елемента DOM, звертатися до атрибутів і вмісту елементів DOM, маніпулювати ними [22]. Бібліотека jQuery містить ряд крос-браузерних методів для анімації елементів без залучення додаткових бібліотек або плагінів, наприклад, ковзання і плавне зникнення. Перевагами jQuery є доступність у багатьох браузерах і більшості версій; легкість синтаксису у використанні і навчанні; мінімізація часу для створення ефектів. Основним же недоліком бібліотеки є більш повільна робота в порівнянні з «чистим» JavaScript.

2.7 Motion UI

Motion UI – бібліотека Sass для створення власних CSS переходів і анімацій, яка дозволяє миттєво анімувати призначений для користувача інтерфейс програми [23]. Володіючи базовими знаннями в області CSS і JavaScript, легко навчитися користуватися Motion UI, а також анімувати практично будь-який елемент програми та задавати йому потрібне поведінку. Можна помітити, що Motion UI збирає в себе всі переваги CSS та JavaScript.

Отже, базуючись на розглянутих раніше технологіях, можна зробити висновок, що зараз існує великий вибір технології для створення анімацій, у кожної з яких є безліч плюсів і мінусів, отже веб-розробнику для роботи є з чого вибрати. Варто відзначити, що не існує єдиної методики оптимальної для розробки веб-анімації, а присутні безліч різноманітних технологій і загальне керівництво по роботі з ними.

Таким чином, в даній ситуації необхідно розробити нову методику, яка буде включати в себе всі потрібні і корисні для веб-розробника рекомендації, які йому знадобляться для роботи з анімацією.

3 РОЗРОБКА МЕТОДИКИ ЕФЕКТИВНОГО ЗАСТОСУВАННЯ АНІМАЦІЇ У WEB

3.1 Формулювання гіпотези відповідно до мети роботи

Створення анімації для кожного веб-продукту являє собою тривалий і копіткий процес для веб-розробника [24]. Для того, щоб цей процес був найбільш ефективний необхідно ретельно вибрати технологію для роботи і визначити етапи для роботи з нею.

Так як для розробки якісного веб-сайту найчастіше необхідно використовувати різні типи анімації одночасно в одному проекті. У таких ситуаціях веб-розробник може застосовувати кілька технологій для роботи, що збільшує вагу файлів проекту і також збільшує час роботи над веб-сайтом.

Як можна помітити такий підхід є вкрай неправильним і вимагає набагато більше часу для виконання роботи. Внаслідок такого процесу може вийти веб-продукт з розрізненої структурою анімації. На відміну від використання різних технологій, існує інший підхід, який передбачає використання єдиної технології, не тільки дозволить скоротити час розробки, підвищити ефективність праці, спростити роботу і зробити її простіше для розробника. Також одним з найважливіших показників, що впливають на якість анімації веб-систем, є швидкість їх завантаження, оскільки користувачі не готові до тривалого очікування завантаження анімаційних ефектів [25]. Отже цей показник також необхідно враховувати при роботі з методикою.

Аналізуючи надану вище інформацію, було вирішено в основі методики розробки анімації використовувати технологію Motion UI, так як дана технологія включає в себе плюси використання одночасно CSS animation та Javascript. Отже, в ході роботи необхідно перевірити наступну гіпотезу: якщо в ході розробки анімації використовувати Motion UI, тоді зменшиться час розробки ефекту та час завантаження анімації.

3.2 Створення методики ефективного застосування анімації у WEB

Проведене в роботі дослідження існуючих інструментів і методик створення анімацій в веб-технологіях, дозволило виявити основні плюси і мінуси, які виникають в процесі розробки веб-продукту [26].

Оптимальний процес роботи з веб-анімаціями повинен бути заснований на універсальній технології, яка буде підходити будь-якому розробнику, мати лінійну структуру роботи. Такий підхід дозволяє значно скоротити час створення анімації, зберігаючи при цьому цілісність розробки проекту.

Запропонована методика є лінійною, полягає в використанні універсальної технології Motion UI, а також передбачає послідовне виконання певного ряду кроків [27].

На першому етапі необхідно підключити бібліотеку Motion UI до проекту, що розробляється. Важливо вибрати правильно варіант підключення, який буде зручний для роботи. Існують кілька варіантів підключення:

- установка необхідних файлів через командний рядок;
- використання, спеціального для Motion UI, модуля Sass. Sass – це метамова на основі CSS, призначена для збільшення рівня абстракції CSS-коду і спрощення файлів каскадних таблиць стилів;
- скачування початкового пакету файлів з офіційного сайту;
- використання CSS;
- використання Javascript.

Найбільш прийнятним варіантом є скачування стартового пакету файлів з офіційного сайту Zurb. Розробники Zurb надали користувачам зручний стартовий пакет, який містить укомплектований CSS. Тому зовсім необов'язково користуватися Motion UI разом з Sass. Пакет можна завантажити зі сторінки сайту і швидко почати прототипування з використанням готових класів для анімацій і переходів CSS.

Наступним етапом є визначення складності анімації, і від цього залежить, що необхідно в подальшій роботі використовувати: CSS або Javascript.

CSS анімації використовуються для роботи з більш легкими анімаціями. Бібліотека включає багато готових класів CSS для швидшого запуску веб-продукту. Основними типами класів CSS є перехідні класи, класи анімації, класи модифікаторів.

Перехідні класи – вони дозволяють додавати переходи (ковзання, поступове зникнення і інші ефекти) до елементів HTML:

а) `slide` – додає анімацію ковзання, додаткові слова вказують на напрямок руху:

- 1) `slide-in-down`;
- 2) `slide-in-left`;
- 3) `slide-in-up`;
- 4) `slide-in-right`;
- 5) `slide-out-down`;
- 6) `slide-out-left`;
- 7) `slide-out-up`;
- 8) `slide-out-right`;

б) `fade` – додає анімацію зникнення (`fade-out`) або прояви елемента (`fade-in`):

- 1) `fade-in`;
- 2) `fade-out`;

в) `hinge` – додає анімацію прояви елемента, як завіси, з додатковим уточненням напрямку руху:

- 1) `hinge-in-from-top`;
- 2) `hinge-in-from-right`;
- 3) `hinge-in-from-bottom`;
- 4) `hinge-in-from-left`;
- 5) `hinge-in-from-middle-x`;

- 6) hinge-in-from-middle-y;
- 7) hinge-out-from-top;
- 8) hinge-out-from-right;
- 9) hinge-out-from-bottom;
- 10) hinge-out-from-left;
- 11) hinge-out-from-middle-x;
- 12) hinge-out-from-middle-y;

г) scale – додає масштабування, а додаткові слова вказують на напрямок руху анімації:

- 1) scale-in-up;
- 2) scale-in-down;
- 3) scale-out-up;
- 4) scale-out-down;

д) spin – додає анімацію обертання елемента:

- 1) spin-in;
- 2) spin-out;
- 3) spin-in-ccw;
- 4) spin-out-ccw.

Класи анімації – дозволяють використовувати різні ефекти тряски, ворухіння і обертання. У них задається ім'я анімації і правила за якими вона працює (що, звідки і куди анімувати):

- shake: качає елемент по горизонталі;
- wiggle: обертає елемент вперед і назад;
- spin-cw: один раз повертає елемент;
- spin-ccw: обертає елемент один раз проти годинникової стрілки.

Класи модифікаторів – взаємодіють з класами переходу і анімації, дозволяють задавати швидкість, часовий період, затримку руху і т.д.

а) speed – визначає швидкість анімації:

- 1) slow (750 мс);
- 2) fast (250 мс);

б) `timing` – визначає часовий період анімації:

- 1) `linear`;
- 2) `ease`;
- 3) `ease-in`;
- 4) `ease-out`;
- 5) `ease-in-out`;
- 6) `bounce-in`;
- 7) `bounce-out`;
- 8) `bounce-in-out`;

в) `delay` – визначає затримку руху анімації:

- 1) `short-delay` (300 мс);
- 2) `long-delay` (700 мс).

Для роботи зі складнішими анімаціями необхідно використовувати JavaScript. Motion UI включає невелику бібліотеку JavaScript, яка може відтворювати переходи, хоча ця конкретна бібліотека не потрібна, щоб скористатися перевагами CSS інтерфейсу Motion UI.

Бібліотека доступна у `window.MotionUI`, або може бути імпортована за допомогою модульної системи.

Об'єкт Motion UI має два методи: `animateIn ()` та `animateOut ()`. Обидві функції приймають три параметри:

- елемент: елемент DOM для анімації;
- анімація: перехідний клас для використання;
- `cb` (необов'язково): зворотний виклик для запуску, коли перехід закінчується. В межах зворотного виклику значенням є елемент DOM, який був перенесений.

Наступним етапом є додаткове опрацювання за допомогою Sass. Попередньо зроблені Motion UI класи CSS використовують довільне значення, які легко налаштовуються за допомогою Sass. За кожен перехід та анімацію відповідає домішка спеціальна комбінація налаштувань Sass, що дозволяє

змінювати налаштування ефекту. Таким чином, і з'являється можливість легко створити анімацію або перехід виключно під свої завдання.

Щоб встановити перехід або анімацію, перш за все треба знайти відповідну комбінацію налаштувань. У спочатку завантаженому пакеті файлів присутній окремий файл, який містить назви зібраних класів CSS з відповідними налаштуваннями.

Останнім етапом є тестування розробленої анімації. Тестування – один з важливих життєвих етапів будь-якої розробки, після якого, пройдений етап вважається успішно завершеним. Тестування, як завершальний етап розробки веб-анімації, грає життєву важливу роль в процесі створення якісного веб-продукту.

Таким чином, розроблена методика створення анімацій веб-продуктів, що дозволяє скоротити час, відведений на розробку веб-анімації. Дану методику можна надати у вигляді діаграми активностей, зображеної на рисунку 3.1.

Наступним етапом є створення анімацій з використанням запропонованої методики.

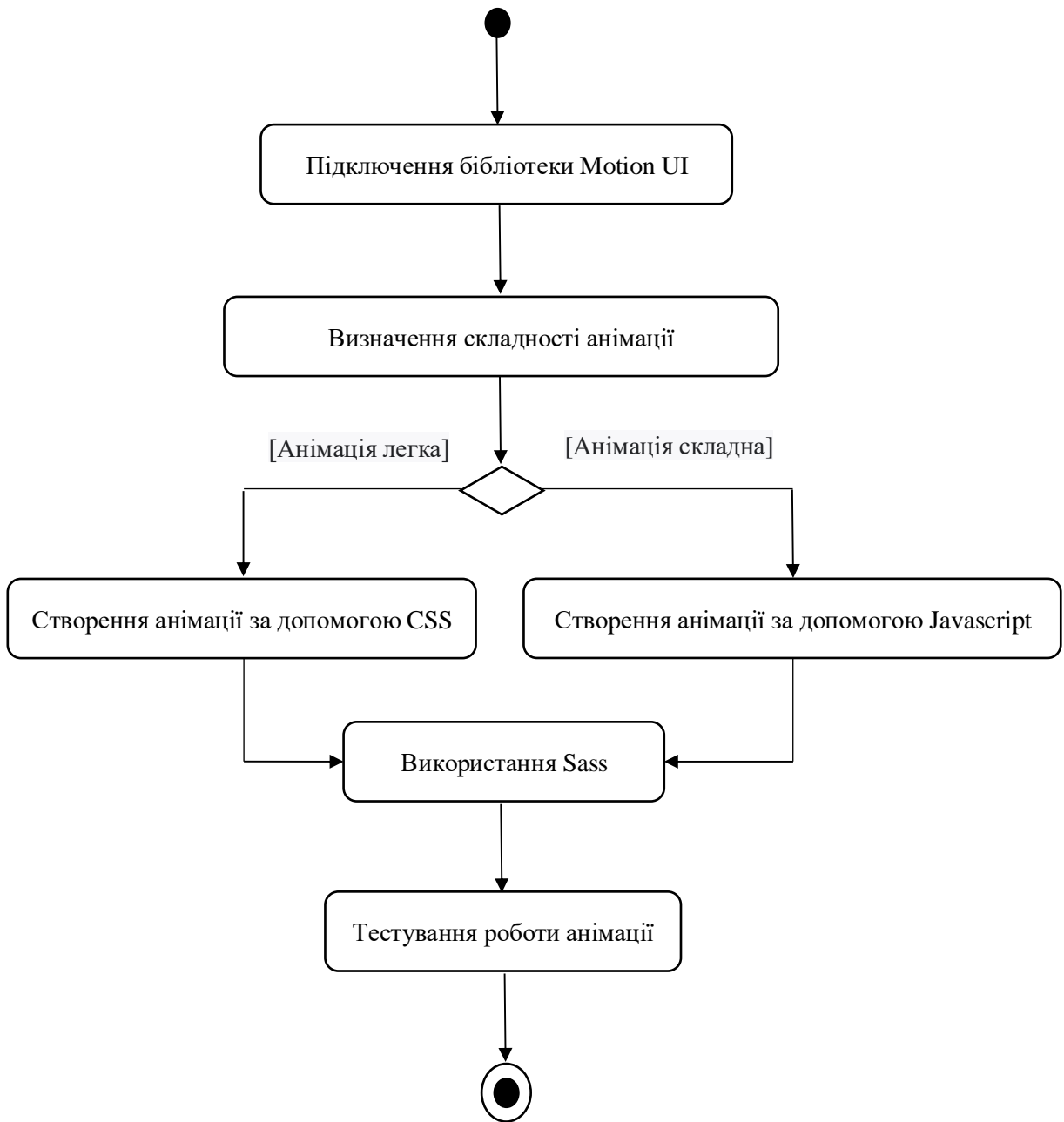


Рисунок 3.1 – Діаграма активностей методики ефективного застосування анімації у Web

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ РОЗРОБЛЕНОЇ МЕТОДИКИ

В ході проведеного експерименту, експерту було запропоновано виконати 20 різних веб-анімацій. В якості анімованого зображення був створений простий об'єкт у вигляді морозива. Даний об'єкт може використовуватися, як в якості елемента декору для веб-продукту, так як елемент завантаження для тематичного веб-сайту (рис 4.1).



Рисунок 4.1 – Приклад створеного зображення

Дизайн елемента було вирішено зробити в сучасній стилістиці, використовуючи яскраві кольори з додаванням світла і тіні, орієнтуючись на мистецтво ХХІ століття [28].

В першому етапі експерименту експерту був виданий список наступних анімацій для розробленого об'єкта:

- поява об'єкта;
- зникнення об'єкта;
- висування об'єкта з верхньої частини екрану;
- висування об'єкта з нижньої частини екрану;
- висування об'єкта з лівої частини екрану;
- висування об'єкта з правої частини екрану;
- рух елемента у верхній бік екрану з подальшим зникання об'єкта;

- рух елемента в нижній бік екрану з подальшим зникання об'єкта;
- рух елемента у лівий бік екрану з подальшим зникання об'єкта;
- рух елемента у правий бік екрану з подальшим зникання об'єкта;
- завантаження елемента в якості завіси, що рухається зверху вниз;
- анімація нескінченного руху фону елемента;
- збільшення масштабу об'єкта;
- зменшення масштабу об'єкта;
- збільшення об'єкта з подальшим його зниканням;
- зменшення об'єкта з подальшим його зниканням;
- поворот об'єкту в праву сторону;
- поворот об'єкту в ліву сторону;
- поворот об'єкту в праву сторону з його подальшим зниканням;
- поворот об'єкту в ліву сторону з його подальшим зниканням.

В подальшій роботі ці анімації використовувалися для проведення тестування разом з групою школярів на визначення часу завантаження виконаних ефектів.

Розроблені анімації реалізуються за допомогою кодів, наданих в додатку А та додатку Б. Розкадрування отриманих анімацій надані на рис. 4.2-4.21.

Завершальним етапом є проведення тестування і математичне опрацювання його результатів задля з'ясування ефективності запропонованої методики.

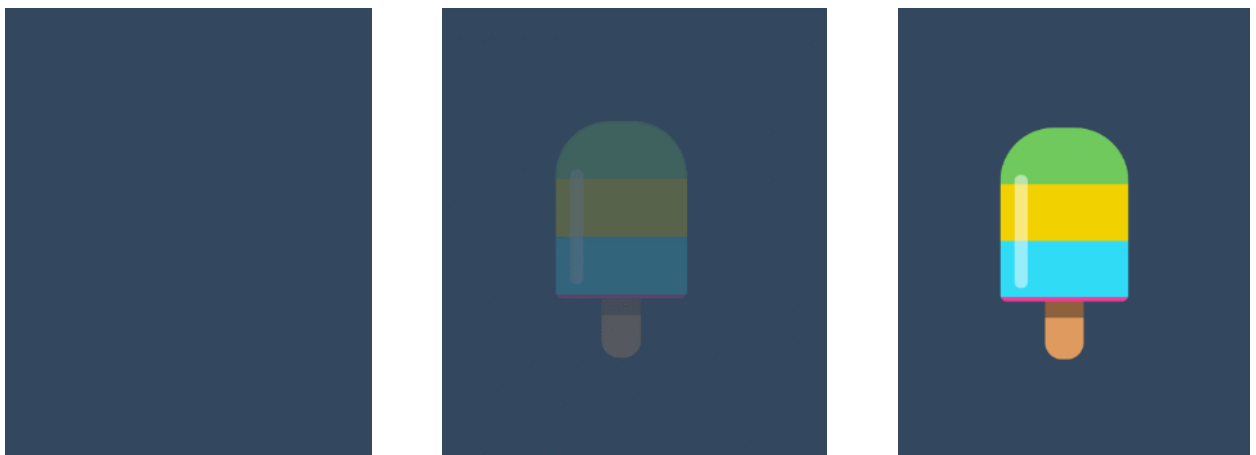


Рисунок 4.2 – Розкадрування анімації появи об'єкта

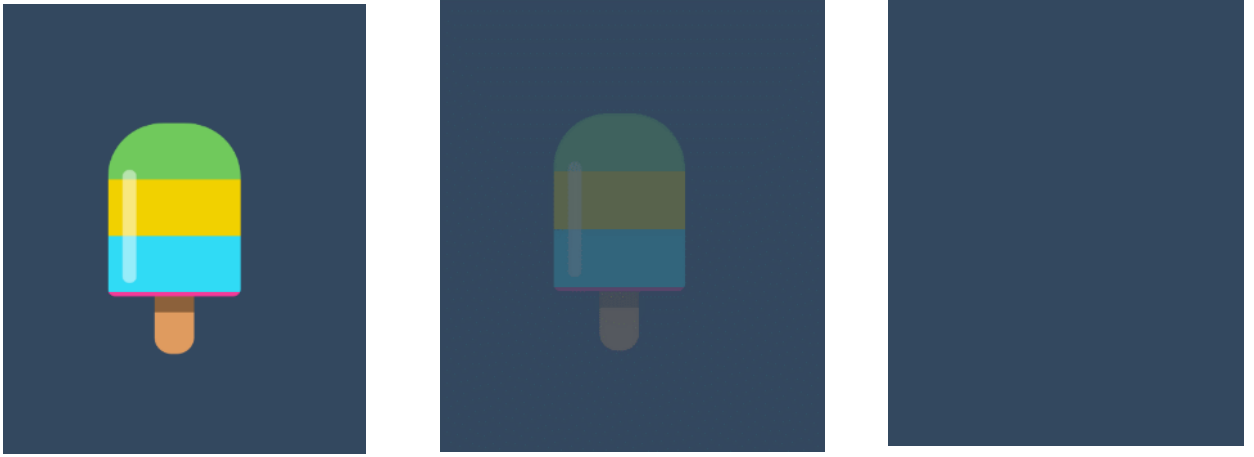


Рисунок 4.3 – Розкадрування анімації зникнення об’єкта

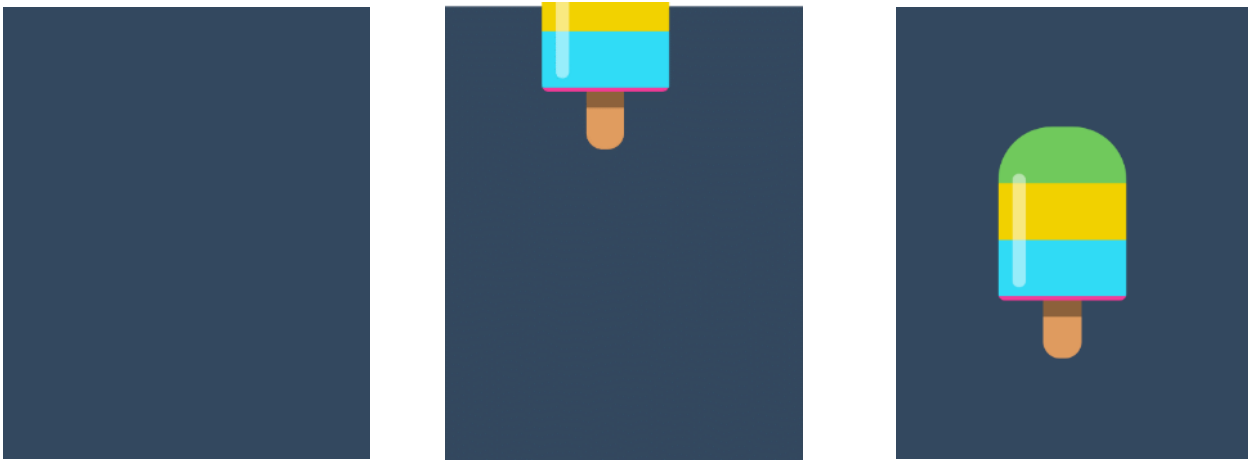


Рисунок 4.4 – Розкадрування анімації висування з верхньої частини екрану



Рисунок 4.5 – Розкадрування анімації висування з нижньої частини екрану



Рисунок 4.6 – Розкадрування анімації висування з лівої частини екрану



Рисунок 4.7 – Розкадрування анімації висування з правої частини екрану



Рисунок 4.8 – Розкадрування анімації зникання в нижній частині екрану



Рисунок 4.9 – Розкадрування анімації зникання в лівій частині екрану

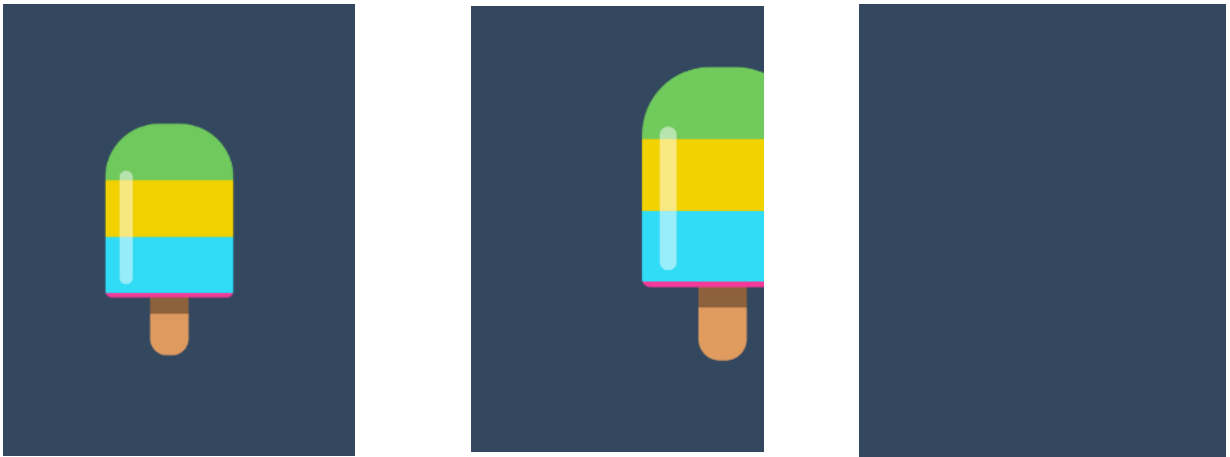


Рисунок 4.10 – Розкадрування анімації зникання в правій частині екрану

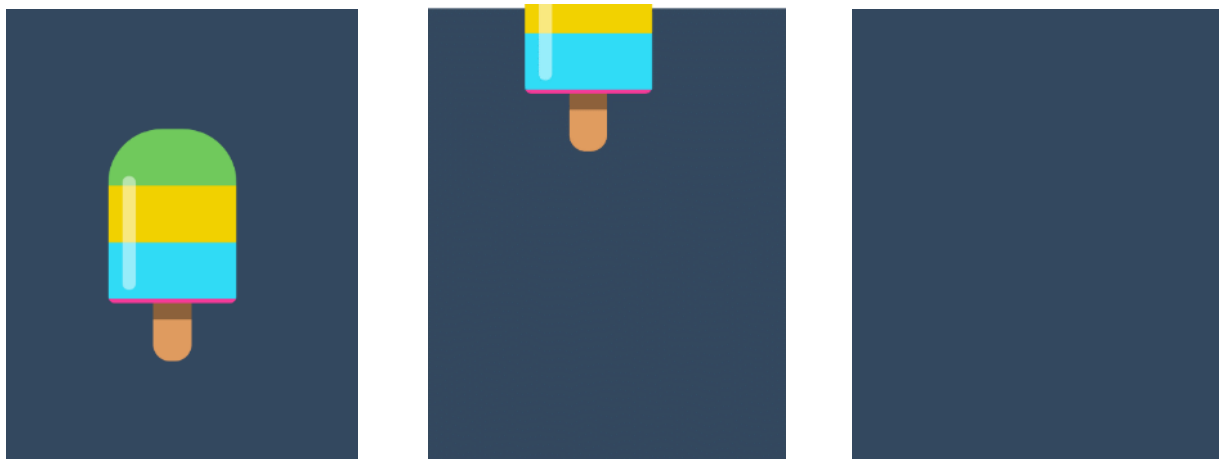


Рисунок 4.11 – Розкадрування анімації зникання в верхній частині екрану

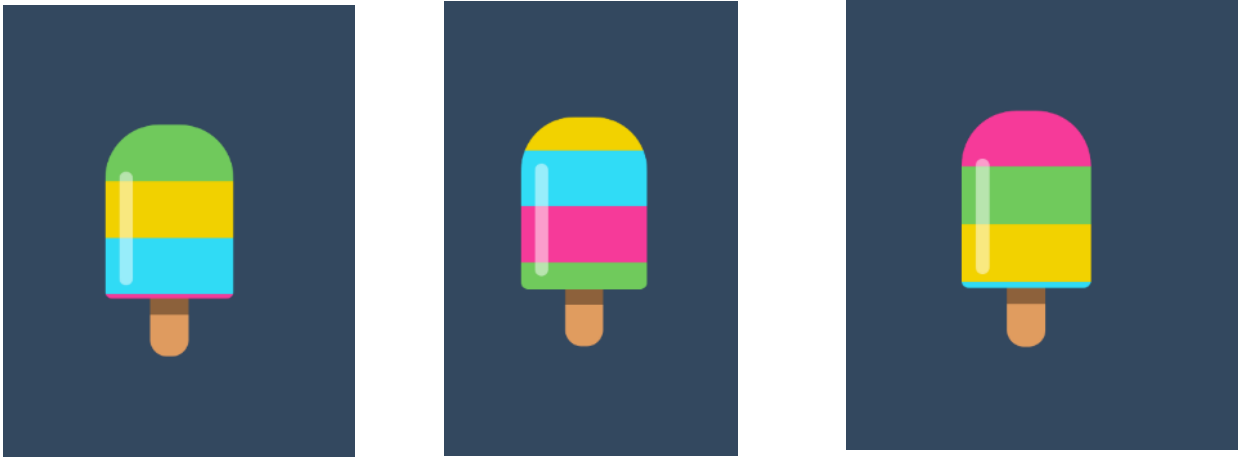


Рисунок 4.12 – Розкадрування анімації фонові частини об'єкту



Рисунок 4.13 – Розкадрування анімації збільшення с подальшим зниканням



Рисунок 4.14 – Розкадрування анімації зменшення с подальшим зниканням

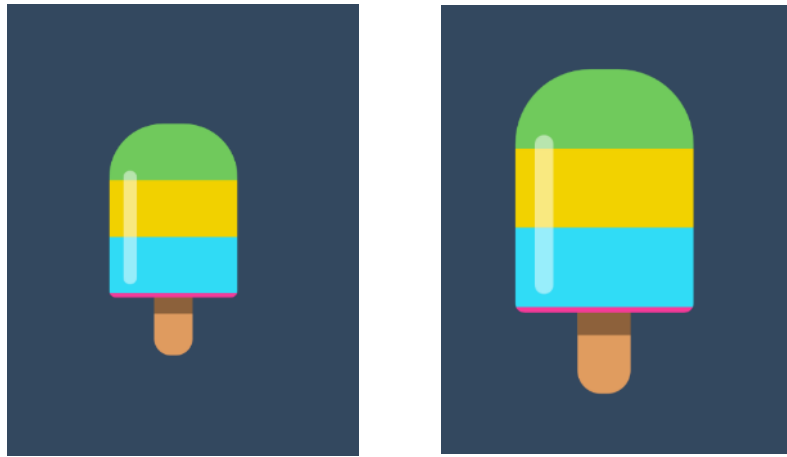


Рисунок 4.15 – Розкадрування анімації збільшення

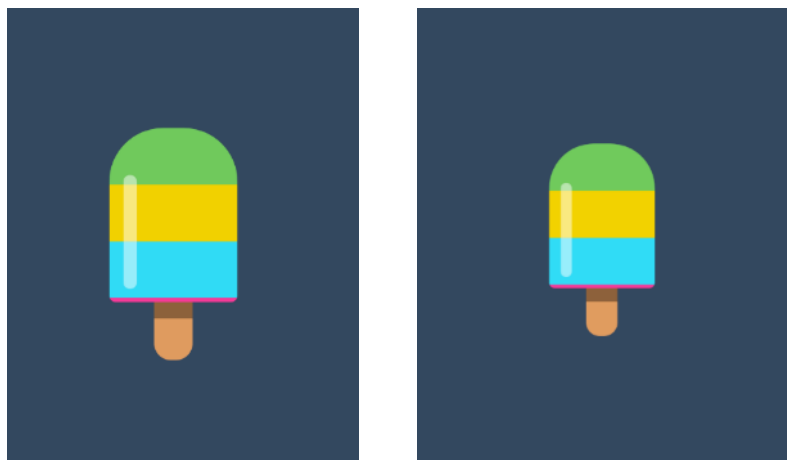


Рисунок 4.16 – Розкадрування анімації зменшення

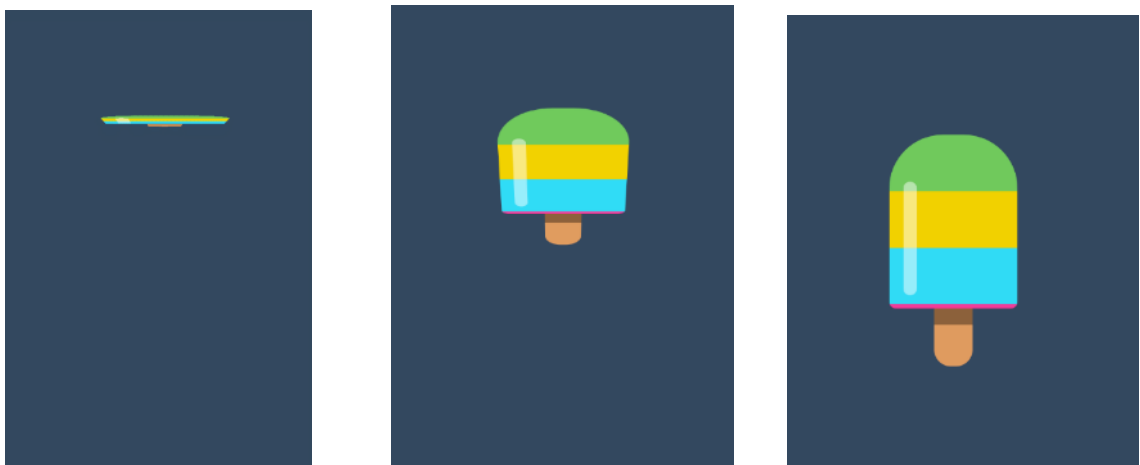


Рисунок 4.17 – Розкадрування анімації «завіси»



Рисунок 4.18 – Розкадрування анімації повороту у праву частину с зниканням



Рисунок 4.19 – Розкадрування анімації повороту у ліву частину с зниканням



Рисунок 4.20 – Розкадрування анімації повороту у праву частину



Рисунок 4.21 – Розкадрування анімації повороту у ліву частину

5 ПРОВЕДЕННЯ ТЕСТУВАННЯ РОЗРОБЛЕНОЇ МЕТОДИКИ

Для проведення тестування був складений наступний список учасників: перше тестування проводилося за допомогою професійного веб-розробника, друге тестування проводилося на базі загальноосвітньої школи. В групу було вирішено визначити учнів 11-го класу, в кількості 20 осіб. Такий розподіл обумовлено необхідністю перевірити ефективність методики для різних типів користувачів від просунутих до початківців з різних сторін. Кожній з груп учасників були приготовані відмінні один від одного завдання для перевірки різних аспектів досліджуваної методики. Перед початком експерименту учасники були проінформовані про структуру дослідження і можливі обмеження, які можуть виникнути під час виконання експерименту.

Для запису результатів дослідження на кожному комп'ютері була заздалегідь встановлена програма запису екрану OBS Studio. Після закінчення тестування програма збереже відеозапис експерименту на комп'ютер для подальшого аналізу.

Експерту в особі веб-розробника було запропоновано виконати 20 нескладних анімацій. Другій групі було запропоновано завантажити на різних 20 пристроях запропоновані анімації і відзначити час їх завантаження. Далі результати двох груп були зіставлені. Метою дослідження було вивчення двох основних параметрів: час розробки анімації і час завантаження анімації. Групи отримали різні завдання для дослідження різних цілей. Отже для першої групи експеримент виражається у виконанні практичної роботи, для другої групи був розроблений спеціальний опитувальник для відзначення часу завантаження. Після виконання всіх завдань і відповідей на всі питання, вся інформація збирається для аналізу і подальшої обробки. Весь процес проілюстрований на рисунку 5.1.

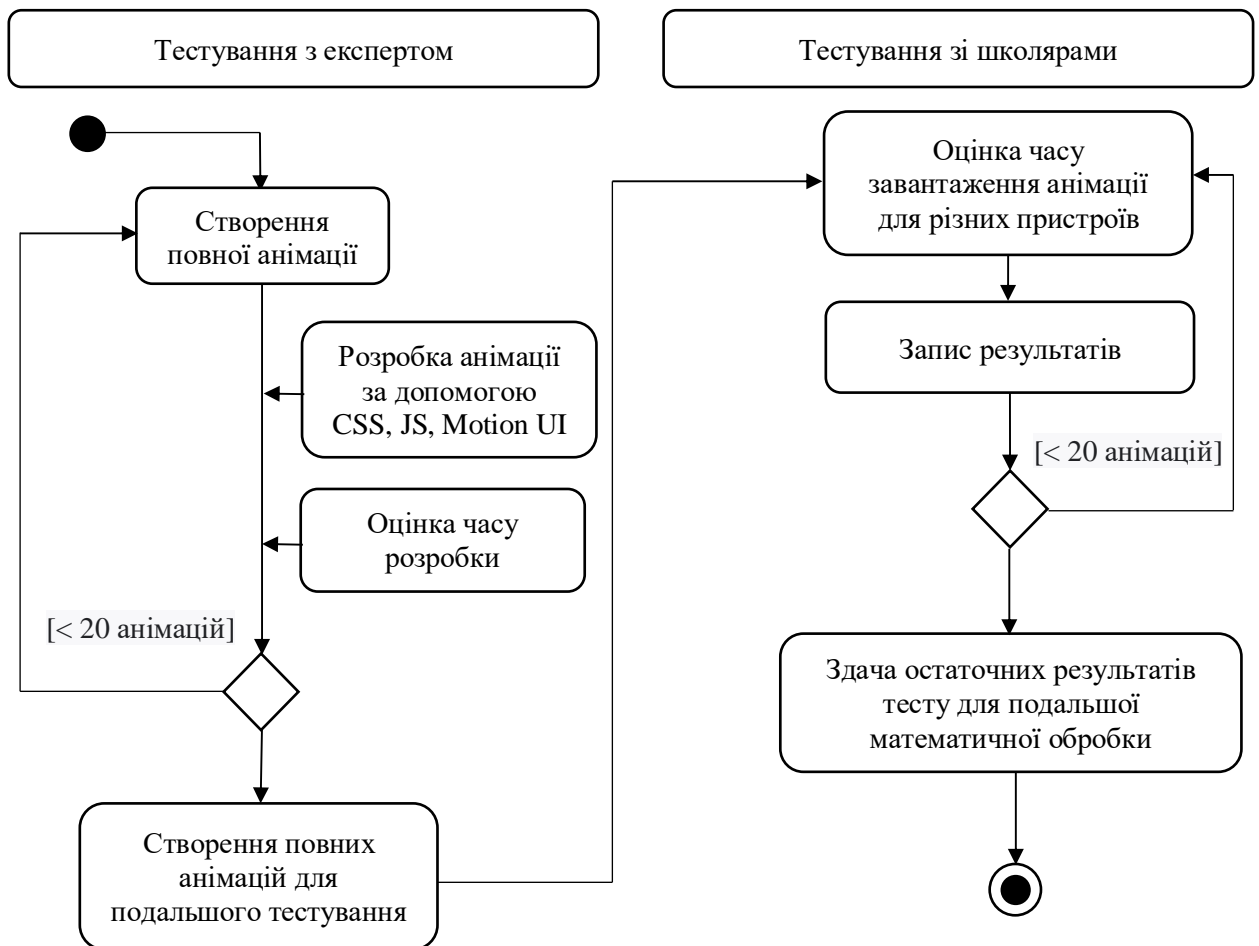


Рисунок 5.1 – Activity- діаграма процедури тестування

5.1 Проведення тестування з експертом

Результати тестування проведеного з експертом були виміряні з урахуванням часу виконання завдання. В даному етапі дослідження експерту необхідно виконати 20 простих анімацій, що відповідає методу малих вибірок. Потрібно розробити 20 нескладних анімацій, використовуючи різні методи створення веб-анімації, а саме технологію CSS, технологію Javascript і бібліотеку Motion UI. Група в кількості 20 чоловік проходила опитування в порівнянні візуальних моментів також 3 різних технологій зі створення веб-анімації за допомогою 20 різних пристроїв.

Час виконання розробки – це кількість часу (в хвилинах), яку учасник витратив з моменту початку розробки та переходу до наступної анімації. Час виконання кожної анімації було виміряно й зафіксовано вручну під час

перегляду записаних відео екрану. Після цього отримана інформація була записана в таблиці, а також була розрахована дисперсія і математичне очікування. Результати виконання тестування користувачами наведені нижче в таблиці 5.1.

Таблиця 5.1 – Результати тестування з експертом.

№ анімації	Час виконання завдання (хв)		
	1 (CSS)	2 (Javascript)	3 (Motion UI)
1	23	12	16
2	22	20	13
3	30	15	12
4	20	17	12
5	23	18	11
6	21	14	11
7	25	11	14
8	22	20	15
9	20	13	15
10	28	13	12
11	20	12	15
12	20	17	14
13	27	17	12
14	20	13	13
15	20	17	11
16	25	16	20
17	21	19	13
18	23	12	15
19	22	20	13
20	30	20	11

В ході подальшого дослідження були використані формули теорії ймовірності для випадкових величин.

Час виконання завдання є випадковою величиною. Випадкової називають величину, яка в результаті випробування прийме одне і тільки одне числове значення, залежне від випадкових чинників і заздалегідь непередбачуване [29].

Таблиця 5.4 – Ряд розподілу часу виконання завдання для Motion UI

x_c	16	13	12	12	11	11	14	15	15	12	15	14	12	13	11	20	13	15	13	11
p_c	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05

$$M[A] = 23.4, D[A] = 12.82.$$

Аналогічно розраховано математичне очікування і дисперсія часу виконання завдання для інших розглянутих технологій.

$$M[B] = 15.8, D[B] = 9.2.$$

$$M[C] = 13.4, D[C] = 4.64.$$

Аналізуючи математичне очікування для розглянутих технологій, можна зробити висновок про те, що середній час виконання розробки анімації, був швидше для бібліотеки Motion UI. Таким чином, можна зробити висновок, що використання бібліотеки Motion UI значно скорочує час роботи над анімацією. Результати дослідження узагальнені на діаграмі нижче (рис. 5.1).

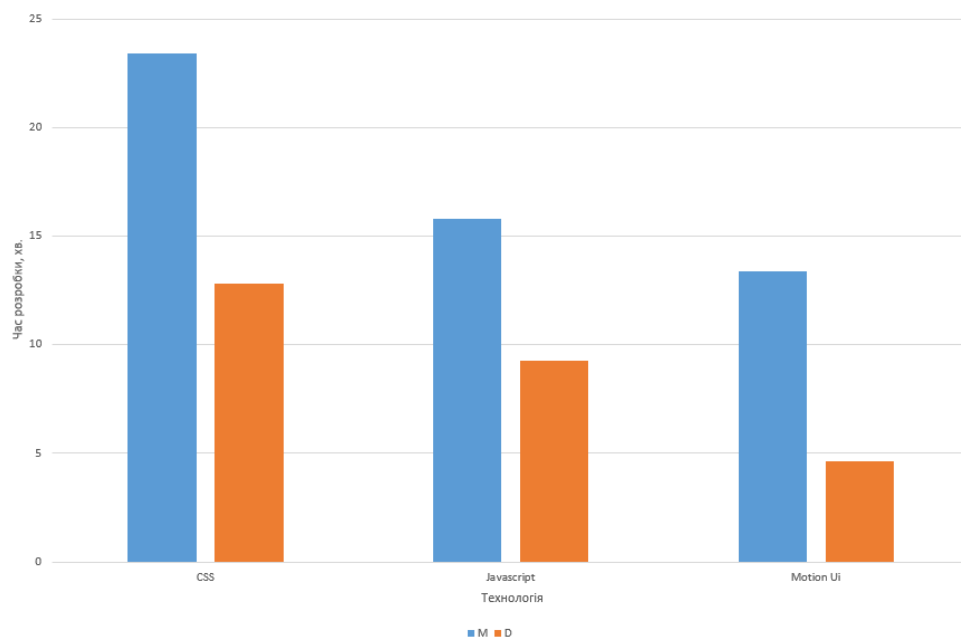


Рисунок 5.2 – Результати експерименту за участю веб-розробника

5.2 Дослідження часу завантаження веб-анімації

Учасникам експерименту було запропоновано розглянути створену заздалегідь анімацію, розроблену трьома методами, за допомогою різноманітних пристроїв . Учасникам експерименту пропонувалася анкета з питанням, на яке вони давали оцінку від 1 до 5. Опитування засновано на суб'єктивному сприйнятті анімації, перевагах і оцінці кожного з 20 учасників експерименту. Кожен учасник експерименту під час тестування використовував певний пристрій для роботи: різні моделі планшетів, телефонів, і комп'ютерів.

В анкеті для експерименту учасникам було запропоновано відповісти на одне питання «Наскільки швидко завантажилася веб-сторінка з анімацією?». Учасники могли вибрати оцінку від 1 до 5, де оцінка 5 означала «Дуже швидко», а 1 – «Дуже повільно».

Для проведення експерименту учням старшої школи були надано наступний список пристроїв для тестування запропонованих веб-анімацій:

- ноутбук Lenovo Legion 5 15IMH05;
- ноутбук Dell Inspiron 3593;
- ноутбук Asus Laptop X540UA-DM3087R;
- ноутбук HP 15-db1014ua;
- ноутбук Prestigio SmartBook 141 C4;
- ноутбук Lenovo L340-15IRH Gaming;
- ноутбук HP 15-dw2048u;
- ноутбук игровой Acer Nitro 5 AN515-55;
- ноутбук Lenovo V14-III;
- ноутбук Dell Vostro 3591;
- телефон Samsung J7;
- телефон Realme 6;
- телефон Nokia 5.3;
- телефон Samsung Galaxy A51;

- телефон Huawei P Smart;
- планшет Samsung Galaxy Tab S6 Lite;
- планшет Lenovo Tab M10;
- планшет Pixus Blast 10.1;
- планшет Lenovo Tab M8;
- планшет Huawei MediaPad.

Результати виконання тестування користувачами наведені нижче в таблиці 5.5.

Таблиця 5.5 – Результати тестування з учнями школи

№ учасника експерименту	Оцінка часу завантаження анімації		
	1 (CSS)	2 (Javascript)	3 (Motion UI)
1	2	3	4
1	5	5	5
2	4	4	4
3	4	4	4
4	4	4	5
5	3	3	5
6	3	4	4
7	3	5	4
8	4	5	4
9	3	4	5
10	5	4	4
11	4	4	3
12	3	3	4
13	3	4	5
14	4	5	4
15	4	4	3
16	5	4	5
17	3	4	5
18	4	3	4
19	4	5	4
20	4	4	4

В даному дослідженні, як і в попередньому, також для обробки результатів тестування технологій проводилось порівняння математичного очікування і дисперсії виконаної задачі для кожної технології.

Головним завданням учасників експерименту було визначення швидкості завантаження анімації для різних технологій.

Виходячи з результатів часу виконання завдання 1 побудовані ряди розподілу величин для різноманітних технологій (табл.5.6-5.8).

Таблиця 5.6 – Ряд розподілу оцінки завантаження анімації для CSS

x_A	5	4	4	4	3	3	3	4	3	5	4	3	3	4	4	5	3	4	4	4
p_A	0,05	0,05	0,1	0,05	0,1	0,05	0,05	0,05	0,05	0,05	0,05	0,1	0,05	0,01	0,05	0,02	0,05	0,01	0,01	0,05

Таблиця 5.7 – Ряд розподілу оцінки завантаження анімації для Javascript

x_B	5	4	4	4	3	4	5	5	4	4	4	3	4	5	4	4	4	3	5	4
p_B	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05

Таблиця 5.8 – Ряд розподілу оцінки завантаження анімації для Motion UI

x_C	5	4	4	5	5	4	4	4	5	4	3	4	5	4	3	5	5	4	4	4
p_C	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,05

$$M[A] = 3.67, D[A] = 0.46.$$

Аналогічно розраховано математичне очікування і дисперсія часу виконання завдання для інших розглянутих технологій.

$$M[B] = 4.1, D[B] = 0.39.$$

$$M[C] = 4.25, D[C] = 0.39.$$

Аналізуючи математичне очікування для розглянутих технологій, можна зробити висновок про те, що учні оцінили позитивно час завантаження для бібліотеки Motion UI. Таким чином, можна зробити висновок, що

використання бібліотеки Motion UI значно скорочує час завантаження анімації для веб-документів.

Результати дослідження узагальнені на діаграмі нижче (рис. 5.2).

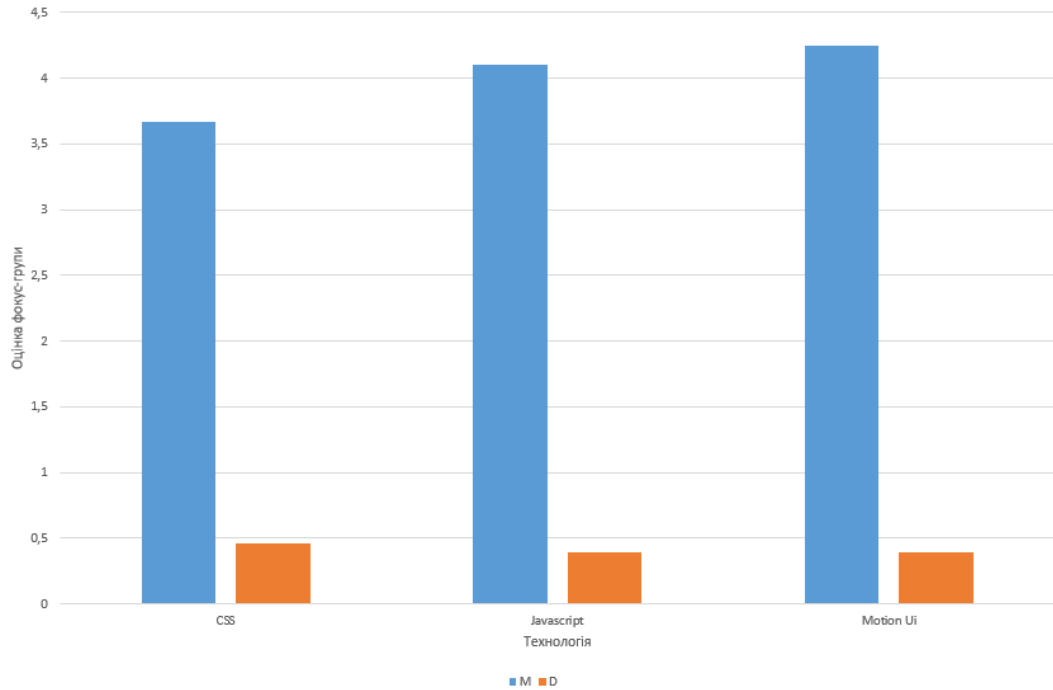


Рисунок 5.2 – Результати експерименту з опитуванням учнів школи

6 ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЕКТУ

6.1 Характеристика науково-дослідного рішення

Науково-дослідницька робота (НДР) – це комплекс теоретичних (експериментальних) досліджень, що проводяться з метою отримання обґрунтованих початкових даних, визначення принципів і шляхів створення (модернізації) продукції [30].

Метою даного розділу є економічне обґрунтування витрат на проведення науково-дослідної роботи з дослідження методів розробки анімацій для веб-ресурсів (а саме, розрахунок трудовитрат та заробітної плати працівникам, одноразових витрат, прибутку, оцінка економічної ефективності НДР).

6.2 Етапи виконання НДР, їх трудомісткість та заробітна плата

У ході виконання науково-дослідної роботи був проведений огляд існуючих теоретичних відомостей розглянутої галузі, досліджено основні методи та алгоритми розробки веб-анімацій, а також за обраними методами створено методику для створення анімацій на веб-ресурсах.

Умовно науково-дослідну роботу (НДР) можна розділити на три етапи: підготовчий, основний і заключний.

На стадії виконання підготовчого етапу були виконані підбір і аналіз інформації для проведення відповідних до постановки задачі робіт. Проведено пошук інформації в електронних джерелах та у відповідній літературі.

На етапі виконання основної частини НДР були виконані такі роботи:

- розгляд існуючих методів створення веб-анімацій;
- аналіз розглянутих методів та вибір оптимальної технології;
- створення остаточної методики для розробки анімації;
- визначення основних критеріїв якості розробленої методики;

- створення за допомогою різних технологій веб-анімацій (тестування часу розробки);
- проведення тестування створених заздалегідь анімацій (тестування часу завантаження анімацій);
- узагальнення результатів тестувань.

У заключній частині проводяться: аналіз результатів виконання НДР, складання звіту по НДР, захист звіту.

Найскладніша і відповідальна частина при плануванні НДР є розрахунок трудомісткості робіт, тому що трудові витрати часто становлять основну частину вартості науково-дослідних робіт і безпосередньо впливають на строки розробки.

Дану роботу виконували 4 фахівця: керівник роботи, керівник фокус-групи, Web developer, Web analyst.

Середня заробітня плата керівника роботи 40 000 грн, веб-розробника становить 30 000 грн, керівника фокус-групи – 12 000 грн, веб-аналітика – 25 000 грн.

Проведемо розрахунок трудовитрат і заробітної плати виконавця робіт.

Середньоденна заробітна плата виконавця робіт ($Z_{\text{ср.дн.}}$) розраховується за формулою:

$$Z_{\text{ср.дн.}} = \frac{Z_{\text{ср.міс.}}}{n}, \quad (6.1)$$

де $Z_{\text{ср.міс.}}$ – середньомісячна зарплата виконавця роботи;

n – число робочих днів у місяці, ($n=22$).

Середньоденна заробітна плата керівника роботи складає:

$$Z_{\text{ср.дн.}} = \frac{40000}{22} = 1818,18 \text{ (грн).}$$

Середньоденна заробітна плата керівника фокус-групи складає:

$$z_{\text{ср.дн.}} = \frac{12000}{22} = 545,45 \text{ (грн).}$$

Середньоденна заробітна плата веб-розробника складає:

$$z_{\text{ср.дн.}} = \frac{30000}{22} = 1363,64 \text{ (грн).}$$

Середньоденна заробітна плата веб-аналітика складає:

$$z_{\text{ср.дн.}} = \frac{25000}{22} = 1136,36 \text{ (грн).}$$

Етапи виконання НДР, перелік і зміст робіт, трудомісткість їх виконання, заробітна плата виконавців робіт представлені в табл. 6.1.

Таблиця 6.1 – Розрахунок трудовитрат і заробітної плати виконавців робіт

Перелік робіт	Кількість виконавців	Посада виконавця	Трудомісткість робіт, люд. днів	Середньоденна заробітна плата, грн.	Сума заробітної плати, грн.
1	2	3	4	5	6
1. Підготовчий етап					
1.1. Розробка та затвердження ТЗ	1	Керівник роботи	1	1818,18	1818,18
1.2 Підготовка довідкових матеріалів та даних для виконання НДР	1	Керівник роботи	2	1818,18	3636,36
2. Основний етап					
2.1 Постановка задачі	1	Веб-аналітик	1	1136,36	1136,36

Продовження таблиці 6.1

1	2	3	4	5	6
2.2 Розгляд сучасних методів створення веб-анімацій	1	Веб-аналітик	3	1136,36	3409,09
2.3 Аналіз розглянутих методів та вибір найбільш підходящої технології	1	Веб-аналітик	4	1136,36	4545,45
2.4. Створення методики оптимізації для розробки анімації	1	Веб-аналітик	2	1136,36	2272,73
3 Тестування					
3.1 Створення за допомогою різних технологій веб-анімацій (тестування часу розробки)	1	Веб-розробник	2	1363,64	2727,27
3.2 Проведення тестування створених заздалегідь анімацій (тестування часу завантаження анімацій)	1	Керівник фокус-групи	1	545,45	545,45
3.3 Узагальнення результатів тестувань	1	Веб-аналітик	1	1136,36	1136,36
Всього			18		17954,55

6.3 Розрахунок одноразових витрат на розробку НДР

Калькуляція собівартості розраховується відповідно до існуючих нормативних актів України. До складу калькуляції входять наступні статті витрат:

- матеріальні витрати;
- витрати на оплату праці;
- єдиний соціальний внесок;
- амортизація основних засобів (вартість машинного часу);

- витрати на спожиту електроенергію;
- інші витрати.

До інших витрат відносяться адміністративні витрати (водопостачання, водовідведення, опалення, освітлення) та вартість послуг зв'язку.

Матеріальні витрати визначаються витратами на матеріали, визначені їх потребою для виконання робіт, і цін, що діють на момент складання калькуляції. Для проведення НДР потрібно: 3 шт. ручки та 3 шт. блокноти. Данні матеріальні витрати потрібні керівнику роботи, керівнику фокус-групи та веб-аналітику. Матеріальні витрати розраховуються за такою формулою:

$$M = \sum_{j=1}^n Q_j \times C_j, \quad (6.2)$$

де M – сумарні витрати на матеріали, в тому числі малоцінні предмети, що швидко зношуються (носії, папір, канцелярське приладдя тощо), або на літературу, яка необхідна для проведення роботи, тощо;

Q_j – кількість використаних одиниць j -го виду матеріалів, $j=(1 \div n)$;

C_j – ціна одиниці j -го виду матеріалів.

Розрахунок матеріальних витрат представлено в табл. 6.2.

Таблиця 6.2 – Розрахунок матеріальних витрат

Найменування	Од. вим.	Кількість (Q_j)	Ціна, (C_j), грн	Сумарні витрати (M), грн.
Ручка	шт.	3	2,00	6,00
Блокнот	шт.	3	14,00	42,00
Усього				48,00

Витрати на оплату праці розраховуються виходячи з необхідного для виконання робіт складу й кількості працівників, а також із середньомісячної заробітної плати. Відповідно до проведених розрахунків витрати на оплату праці виконавців роботи дорівнюють 17954,55 грн.

Єдиний соціальний внесок (ЄСВ) – обов'язкові відрахування на загальнодержавне соціальне страхування. ЄСВ оплачується підприємцем за себе і за кожного найманого працівника. Є внеском в загальнодержавну систему соціального страхування з метою захисту у випадках, передбачених законодавством, прав застрахованих осіб на отримання страхових виплат. Єдиний соціальний внесок сплачується на рахунки органів доходів і зборів за місцем обліку підприємця.

Для об'єкта дослідження ставка єдиного соціального внеску дорівнює 22% від витрат на оплату праці, тобто розмір ЄСВ дорівнює 3950 грн.

При виконанні НДР застосовувалось наступне обладнання: ноутбуки в кількості 4 шт. вартістю 30000 грн.

Вищенаведене устаткування є власністю організації виконавця, тому доцільно розрахувати суму амортизаційних відрахувань на період виконання НДР. Амортизація основних засобів розраховується за формулою:

$$AB = \sum_{k=1}^L \frac{BO_k}{TE_k} \times T, \quad (6.3)$$

де AB – сума амортизаційних відрахувань, нарахованих під час проведення науково-дослідницької роботи;

BO_k – вартість основних засобів k -го виду;

TE_k – термін експлуатації основних засобів k -го виду, днів;

T – термін науково-дослідницької роботи, днів;

L – кількість видів обладнання.

Підставивши відомі значення у формулу (6.3), визначимо величину амортизаційних відрахувань. Отже маємо:

$$AB = \frac{30000 \cdot 3}{730} + \frac{30000 \cdot 12}{730} + \frac{30000 \cdot 2}{730} + \frac{30000}{730} = 739,73 \text{ (грн.)}$$

Витрати на використану обладнанням електроенергію розраховуються за формулою:

$$Z_e = M \cdot t \cdot T_{кВт}, \quad (6.4)$$

де M – потужність устаткування, тобто кількість енергії, споживаної за одиницю часу (кВт/година);

t – кількість годин використання устаткування за період проведення науково-дослідницької роботи;

$T_{кВт}$ – тариф, тобто вартість використання 1 кВт електроенергії.

Споживна потужність комп'ютера складає 0,2 кВт за годину. Тариф споживачів за першим класом напруги, тобто 35 кВт та більше), складає 1,68 грн./кВтгодин (без ПДВ). Підставивши значення у формулу (6.4), визначимо величину витрат на спожиту електроенергію:

$$Z_e = 0,2 \cdot 1,68 \cdot 24 + 0,2 \cdot 1,68 \cdot 96 + 0,2 \cdot 1,68 \cdot 16 + 0,2 \cdot 1,68 \cdot 8 = 48,38 \text{ (грн.)}$$

До інших статей витрат відносяться такі:

- адміністративні витрати: (водопостачання, водовідведення, освітлення, опалення), які прийнято у розмірі 20% від витрат на оплату праці;
- вартість оплати послуг зв'язку.

Вартість оплати послуг зв'язку становитиме:

Інтернет – із розрахунку 200 грн. на місяць (безлімітний пакет); всього 200 грн. за 18 днів виконання НДР.

За час виконання НДР витрати на відрядження, інформаційні послуги та маркетингові заходи не мали місця.

Результати розрахунку кошторису витрат, тобто одноразових витрат, на виконання НДР «Розробка методики ефективного застосування анімації у WEB» наведені в табл. 6.3.

Таблиця 6.3 – Кошторис витрат на розробку НДР

№ з/п	Стаття витрат	Сума, грн.
1	Заробітна плата	17954,55
2	Єдиний соціальний внесок (22,0 % від п.1)	3950
3	Матеріальні витрати	48
4	Амортизація основних засобів	739,73
5	Витрати на спожиту електроенергію	48,38
6	Інші витрати, у тому числі:	
6.1	адміністративні витрати (20% від п.1)	3590,91
6.2	вартість послуг зв'язку	200
	Всього витрати(<i>Bp</i>)	26531,57

Таким чином, кошторис витрат на виконання даної НДР відбиває сумарні витрати за статтями п.1 ÷ п.6 та складає 26531,57 грн.

6.4 Оцінка результатів науково-дослідної роботи

Останнім етапом є визначення результатів НДР. Результат – наслідок ланцюжка (низки) дій або подій, виражених якісно або кількісно. Можливі результати включають перевагу, незручність, вигоду, втрату, цінність і перемогу [31]. В загальному випадку оцінка результатів НДР – це визначення ефективності отриманих рішень порівняно з сучасним науково-технічним рівнем.

Відповідно до теми даної роботи можна зробити висновок про те, що у якості результату впровадження НДР є зменшення часу розробки веб-анімацій та зменшення часу завантаження анімацій на веб-сторінці, що має досить велике значення для веб-розробників.

Результат від впровадження НДР визначається за такою формулою:

$$\Delta P_j = |X_{б_j} - X_{н_j}|, \quad (6.5)$$

де ΔP_j – покращення j -ої характеристики досліджуваного процесу за рахунок впровадження результатів НДР ($j=1, m$);

m – кількість досліджуваних характеристик;

$X_{бj}$ – базове значення j -ої характеристики, тобто до впровадження результатів НДР;

$X_{нj}$ – нове значення j -ої характеристики після впровадження запропонованих рішень.

В якості досліджуваних характеристик виступають час розробки і час завантаження веб анімацій. У проведеному дослідженні були створені 20 різних анімацій, які розроблялися спочатку за допомогою двох стандартних методик, і після за допомогою розробленої нової методики.

Підставивши відповідні значення отримані в ході експерименту для першої технології в формулу (6.5), визначимо результат від впровадження НДР у чисельному вигляді:

$$\Delta P_1 = |19,1 - 13,4| = 5,7 \text{ (КБ);}$$

$$\Delta P_1 = |3,67 - 4,25| = 0,58.$$

Подібні розрахунки були також проведені для інших анімацій, та наведені у таблиці 6.4.

Таблиця 6.4 – Результат від впровадження НДР в порівнянні з іншими технологіями

Критерії	Технологія 1			Технологія 2		
	До	Після	Різниця	До	Після	Різниця
Час розробки, хв	19,1	13,4	5,7	15,8	13,4	2,4
Оцінка швидкості завантаження анімації	3,67	4,25	-0,58	4,1	4,25	-0,15

Таким чином, отриманий результат свідчить про те, що за допомогою запропонованої методики можна зменшити час розробки веб-анімацій, а також

збільшити швидкість з завантаження на веб-сайті. Дані параметри впливають на час виконання всього веб-проекту і також на можливість утримання користувачів на розроблюваних веб-ресурсах. Роботу у цілому можна враховувати ефективною або такою, що має високий науковий та технічний рівень.

ВИСНОВКИ

В представленій атестаційній роботі було проведено дослідження технологій створення анімацій, виявлені їх переваги, недоліки та галузі використання. Сформульовано гіпотезу щодо мети роботи. На основі проведеного дослідження розроблено методику ефективного застосування анімації у WEB, що дозволяє скоротити як час розробки веб-анімації, так і час її завантаження. Розроблену методику використано задля створення прикладів анімацій, які у свою чергу взяли участь у тестуванні. Результати тестування досліджені методами математичної статистики. Проведене дослідження довело ефективність розробленої методики.

Таким чином, розроблена у атестаційній роботі методика ефективного застосування анімації у WEB, що відрізняється від існуючих методик використання більш універсальної технології, підвищеною ефективністю за рахунок скорочення часу розробки та часу завантаження, а також підвищенням якості продукту в цілому, є доцільною. Запропонована методика буде корисна як для початківців веб-розробників, так і для фахівців.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Егорова И.Н., Филипенко О.В. Разработка методики создания графического интерфейса веб-сайтов // ScienceRise. 2016. № 1/2 (18). С. 58-61.
2. Егорова И.Н., Гладкая А.А. Исследование возможностей резервного копирования веб-сайтов, созданных на основе WordPress // Вісник Національного технічного університету «ХПІ». Серія: Нові рішення в сучасних технологіях. 2017. № 23 (1245). С. 95-99.
3. Егорова И.Н., Худолей А.Ю. Исследование возможностей компонентного подхода при разработке веб-сайтов // Системи обробки інформації. 2017. Випуск 4 (150). С.76-78.
4. Егорова И.Н., Кадушкевич О.Н. Методика эффективного использования инструментов Google Analytics // ScienceRise. 2016. № 1/2 (18). С. 40-44.
5. Егорова И.Н., Кочура Л.А. Разработка обучающей игры «Web-технологии» // Вісник НТУ «ХПІ». Серія: Нові рішення в сучасних технологіях. 2019. № 2. С. 49-53.
6. Егорова И.Н., Антипенко К.Д. О применении кривых Безье для улучшения CSS анимации // Системи обробки інформації. 2019. Випуск 2 (157). С. 40-44.
7. Егорова И.Н., Горелова Р.А. Разработка методики поисковой оптимизации веб-сайтов // Системи обробки інформації. 2017. Випуск 4 (150). С. 73-75.
8. Цель создания сайта // RBS webmarket. URL: [https:// http://rbs-webmarket.com/faq/f-vop/fv-what-order-site/the-goal-of-creating-a-website.html](https://http://rbs-webmarket.com/faq/f-vop/fv-what-order-site/the-goal-of-creating-a-website.html) (дата звернення: 01.11.2020).
9. What is Animation? Definition and Types of Animation. URL: <https://www.studiobinder.com/blog/what-is-animation-definition/> (дата звернення: 01.11.2020).

10. Приёмы создания плавной анимации. URL: <https://html5book.ru/priyomu-sozdaniya-plavnoy-animacii/> (дата звернення: 01.11.2020).
11. Что такое GIF-анимация? URL: <https://sites.google.com/site/kurspoadobephotoshop/sozдание-gif-animacii/cto-takoe-gif-animacia> (дата звернення: 01.11.2020).
12. Что такое Flash-анимация? URL: <https://www.internet-technologies.ru/articles/chto-takoe-flash-animaciya.html> (дата звернення: 01.11.2020).
13. На чём писать мобильные кроссплатформенные приложения // URL: <https://livetyping.com/ru/blog/na-chem-pisat-krossplatformennye-prilozhenija> (дата звернення: 01.11.2020).
14. Как оживить свой сайт: 5 инструментов для анимации в современном WEB. URL: <https://tproger.ru/digest/5-animation-tools-for-web/> (дата звернення: 01.11.2020).
15. ДСТУ 7157:2010. Електронні видання. Основні виходи і вихідні відомості.
16. CSS. SEO Wiki. URL: <https://wiki.rookee.ru/css/> (дата звернення: 01.11.2020).
17. CSS-анимация. URL: <https://learn.javascript.ru/css-animations> (дата звернення: 01.11.2020).
18. Что дизайнеру нужно знать о SVG: за и против. URL: <https://habr.com/ru/company/htmlacademy/blog/257039/> (дата звернення: 01.11.2020).
19. Єгорова І.М. Проектування та розробка Web-документів: навч. посібник. Харків: ХНУРЕ, 2018. 264 с.
20. WebGL – OpenGL ES 2.0 for the Web. URL: [Khronos.org](https://www.khronos.org) (дата звернення: 01.11.2020).
21. Javascript-анимации. URL: <https://learn.javascript.ru/js-animation> (дата звернення: 01.11.2020).
22. Фримен А. jQuery для профессионалов = Pro jQuery. М.: Вильямс, 2012. 960 с.

23. Как создать анимацию и переходы с помощью Motion UI. URL: <https://habr.com/ru/company/ua-hosting/blog/279067/> (дата звернення: 01.11.2020).

24. Лучшие инструменты анимации для вашего веб-приложения UI. URL: <https://techrocks.ru/2018/11/16/web-animation-tools-for-web-app/> (дата звернення: 01.11.2020).

25. Егорова И.Н., Самокиш В.В. Об использовании спрайтовой анимации в веб-изданиях // Системи обробки інформації. 2018. Випуск 3(154). С. 100-104.

26. Создание анимации для сайта – часть успешности проекта. URL: <https://artjoker.ua/ru/blog/sozдание-animatsii-dlya-sayta-chast-uspeshnosti-proekta/> (дата звернення: 01.11.2020).

27. Комина М.М. О выборе эффективной технологии создания анимации для веб-приложений // Радіоелектроніка та молодь у ХХІ столітті: матеріали ХХІV Міжнародного молодіжного форуму (19-21 жовтня 2020, м. Харків). 2020. С. 353-354.

28. Комина М.М. Стили изобразительного искусства в современном веб-дизайне // Поліграфічні, мультимедійні та web-технології (PMW-2019): матеріали молодіжної школи-семінару IV Міжнародної науково-технічної конференції (14-17 травня 2019, м. Харків). 2019. С. 140-145.

29. Глава 5. Случайные величины. 5.1. Понятие случайной величины // Математика и информатика. Учебное пособие по всему курсу. URL: <https://tproger.ru/digest/5-animation-tools-for-web/> (дата звернення: 01.11.2020).

30. Методичні рекомендації до виконання економічної частини дипломних проектів, робіт для студентів денної та заочної форми навчання усіх спеціальностей / Соколова Л.В. та ін. Харків, 2015. 49 с.

31. Результат. Толковый словарь Ушакова // dic.academic.ru. URL: <https://dic.academic.ru/dic.nsf/ushakov/1008122> (дата звернення: 01.11.2020).