

УДК 004.51

БАШОВИЙ В. М., СТАЦЕНКО В. В., СТАЦЕНКО Д. В.

Київський національний університет технологій та дизайну, Україна

РОЗРОБКА АДАПТИВНОГО WEB-ІНТЕРФЕЙСУ ДЛЯ ДОДАТКУ КЕРУВАННЯ ОСОБИСТИМИ ФІНАНСАМИ

Мета. Створення адаптивного web-інтерфейсу для додатку керування та обліку особистих фінансів.

Методика. Використовуються методи побудови web-інтерфейсів, мова розмітки гіпертексту (HTML), каскадна мова стилю сторінок (CSS), інструменти дослідження web-додатків.

Результати. Проаналізовано вимоги до адаптивних web-інтерфейсів, що дозволяють використовувати їх на пристроях з різними параметрами екранів. Визначено складові елементи web-інтерфейсу додатку для обліку та аналізу використання особистих фінансів. Розроблено три варіанти компоновки цих елементів для екранів персональних комп'ютерів, планшетів та смартфонів. Запропоновані варіанти компоновки дозволяють уникнути появи горизонтальної прокрутки на екрані та працювати з інформацією без використання функції масштабування. Встановлено, що медіа запити, які входять до специфікації CSS3, дозволяють гнучко керувати взаємним розміщенням елементів інтерфейсу та їх зовнішнім виглядом. Перетворення табличної інформації у форму списків для відображення на вузьких екранах смартфонів здійснено за рахунок зміни HTML розмітки. Створено та розміщено у мережі Інтернет прототип запропонованого додатку.

Наукова новизна. Проаналізовано можливості медіа запитів, що входять до специфікації CSS3, з точки зору керування взаємним положенням компонентів web-інтерфейсів.

Практична значимість. Розроблено прототип web-інтерфейсу додатку керування та обліку особистих фінансів, що дозволяє уникнути появи горизонтальної прокрутки на екрані та працювати з інформацією без використання функції масштабування.

Ключові слова: web-додаток; web-інтерфейс; HTML; CSS; JavaScript; фреймворк.

Вступ. Додатки для зберігання та обробки інформації користувачів є одним з найбільш розповсюджених типів програмного забезпечення. Можливість здійснювати швидко пошук, систематизацію, автоматичний аналіз та копіювання даних завжди зумовлювали інтерес до таких додатків та їх популярність. З моменту появи персональних комп'ютерів створені тисячі програмних продуктів для роботи з даними користувачів. Водночас, поява нових пристроїв, широкосмугового доступу до мережі Інтернет, оновлення операційних систем потребує розробки нових версій прикладного програмного забезпечення, яке має враховувати як нові технічні характеристики обладнання, так і зміни в очікуваннях користувачів до функціональних можливостей.

Сьогодні більшість користувачів одночасно працює з декількома пристроями: персональним комп'ютером, планшетом, смартфоном. Це зумовлює появу проблеми синхронізації даних між пристроями та забезпечення зручного доступу до неї. Також екрани мобільних та стаціонарних пристроїв суттєво відрізняються за розмірами, роздільною здатністю та співвідношенням сторін. Відповідно, забезпечення зручної роботи з додатками на різних платформах потребує їх адаптації під розміри екранів [1]. Наявність цих проблем призвела до поширення програмних систем з клієнт-серверною архітектурою, та web-інтерфейсів [2] для взаємодії з користувачами. У таких додатках інформація централізовано зберігається на сервері та є доступною для всіх пристроїв через мережу Інтернет. В результаті вирішується проблема синхронізації даних між пристроями. Використання web-інтерфейсів дозволяє уникнути створення окремих додатків для кожної з платформ, оскільки в даному

випадку в якості цільової платформи виступає браузер. Водночас, проблема адаптації інтерфейсу під параметри екранів залишається актуальною.

Постановка завдання. Метою роботи є створення web-інтерфейсу для додатку керування та обліку особистих фінансів. Додаток має забезпечувати можливість створення записів про надходження та витрати, групування цих записів по категоріям, відображення статистики, інформації про баланс та поточні курси валют. Інтерфейс повинен підтримувати три типи пристроїв: персональні комп'ютери, планшети та смартфони. Всі елементи мають відображатись без горизонтальної прокрутки та необхідності масштабування зображення.

Результати дослідження. Перед початком розробки додатку були проаналізовані існуючі web-інтерфейси. Результати показали, що існують два основні підходи до забезпечення підтримки декількох пристроїв з різними параметрами екранів:

1. Створення окремого web-інтерфейсу для мобільних пристроїв.
2. Створення адаптивного web-інтерфейсу.

В першому випадку всі пристрої розділяються на дві великі групи. До першої відносять комп'ютери та планшети, до другої – смартфони. Для кожної з груп пристроїв розробляють окремий інтерфейс. Додаток визначає тип пристрою виходячи з заголовків, які відправляє браузер у складі HTTP запиту та перенаправляє користувача на потрібну версію інтерфейсу. До недоліків такого рішення відносяться: необхідність розробляти та підтримувати дві версії інтерфейсу, тільки два можливі варіанти відображення інтерфейсу, необхідність аналізувати заголовки web браузера користувача.

При розробці адаптивних інтерфейсів задаються правила відображення окремих блоків та їх взаємного розташування в залежності від параметрів пристрою. Ці правила аналізує браузер користувача та відповідно до них формує зовнішній вигляд елементів інтерфейсу. Таким чином сервер відправляє однакові відповіді всім типам клієнтів.

Інтерфейс додатку керування та обліку особистих фінансів було вирішено створити на основі адаптивного підходу. Для його реалізації використовуються три основні технології: HTML (HyperText Markup Language), CSS (Cascading Style Sheets) та мова JavaScript [3].

За допомогою HTML формується структура сторінки, розміщуються текстові написи та зображення. CSS відповідає за відображення елементів на сторінці, наприклад, розміри шрифтів, відступи між блоками, колір, тощо [4]. JavaScript забезпечує інтерактивну поведінку інтерфейсу, зокрема реакцію на дії користувача [5].

Інтерфейс додатку із робочою назвою Wallet розроблено з використанням зазначених технологій та фреймворку ReactJS [6]. Додаток розміщено за адресою <https://wallet-app-1.netlify.app/>. Для перевірки роботи додатку потрібно зареєструватись та внести будь-які тестові дані. Після цього можна перевірити його роботу на різних пристроях.

Виходячи з вимог до функціональних можливостей додатку, були визначені основні елементи інтерфейсу (рис. 1 та 4):

- Головне меню (1).
- Блок з інформацією про поточний баланс (2).
- Блок з інформацією про курси валют (3).
- Таблиця з переліком надходжень та видатків (4).
- Кругова діаграма з інформацією про надходження та видатки по категоріях (5).
- Таблиця з інформацією про надходження та видатки по категоріях (6).

Серед зазначених елементів найбільшу площу займають таблиця (4) та блоки (5) і (6) зі статистикою по категоріях. Водночас, інформація про поточний баланс (2) має бути у користувача постійно перед очима. Виходячи з цих міркувань, інтерфейс додатку було вирішено розділити на дві сторінки. На першій показати блоки (1)-(4). На другій – блоки (1)-(3), (5) та (6).

Відображення цих сторінок на екрані комп'ютера показано на рис. 1 та 4. В цьому випадку блоки (1)-(3) розташовані один під одним та займають одну третю ширини екрану. Таблиця (4) розташована праворуч та займає дві третіх екрану.

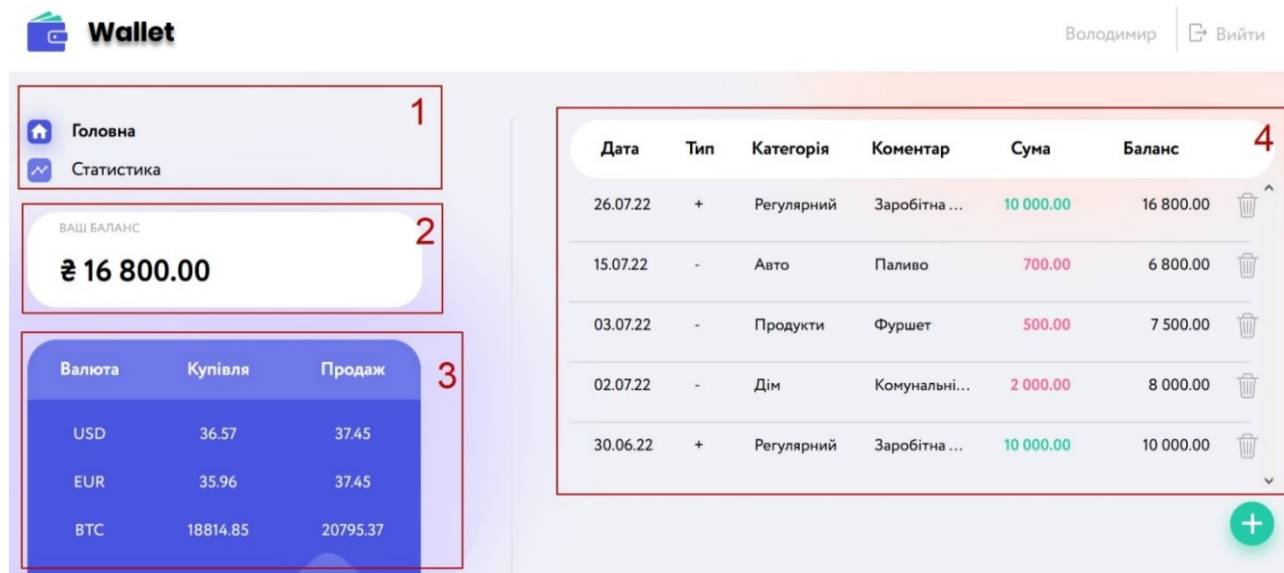


Рис. 1. Відображення сторінки «Головна» на екрані комп'ютера

Розташування цих самих елементів на екранах планшетів та смартфонів показано на рис. 2 та 3, відповідно. На екрані планшета таблиця (4) переміщується вниз сторінки, відносний розмір блоків (1) та (2) збільшується до 50% від ширини екрану, а блок (3) переміщується праворуч від блоків (1) та (2). Така компоновка дозволяє забезпечити використання всієї площі екрану та виділити для таблиці (4) достатню кількість місця щоб нею можна було користуватись без горизонтальної прокрутки.

Адаптивне розташування блоків здійснюється за допомогою медіа запитів, що входять до специфікації CSS3 [7-10]. Вони дозволяють задавати правила оформлення елементів на сторінці в залежності від параметрів екрану. У загальному випадку запити виглядають наступним чином:

```
@media (max-width: 1250px) {  
  /* ... */  
}
```

Кожен запит починається з ключового слова `@media` після якого в дужках вказується умова за якої правило застосовується. Для даного прикладу CSS стилі застосовуватимуться для екранів з шириною 1250 пікселів або меншою.

У додатку Wallet такі запити створені для всіх елементів, що змінюють розташування на екранах з різними параметрами. Зокрема, для блоку з основним контентом сторінки використовуються наступні правила:

```
@media screen and (min-width:768px) {  
  .container {  
    max-width: 768px;  
    padding: 0 32px;
```

```
}  
}
```

```
@media screen and (min-width:1280px) {  
.container {  
display: flex;  
max-width: 1280px;  
padding: 0 16px;  
}  
}
```

Складніше перетворення виконується для блоку (4) при відображенні на екранах смартфонів (рис. 3). В цьому випадку ширина екрану пристрою не дозволяє розмістити всю інформацію (всі сім стовпчиків) в один рядок.

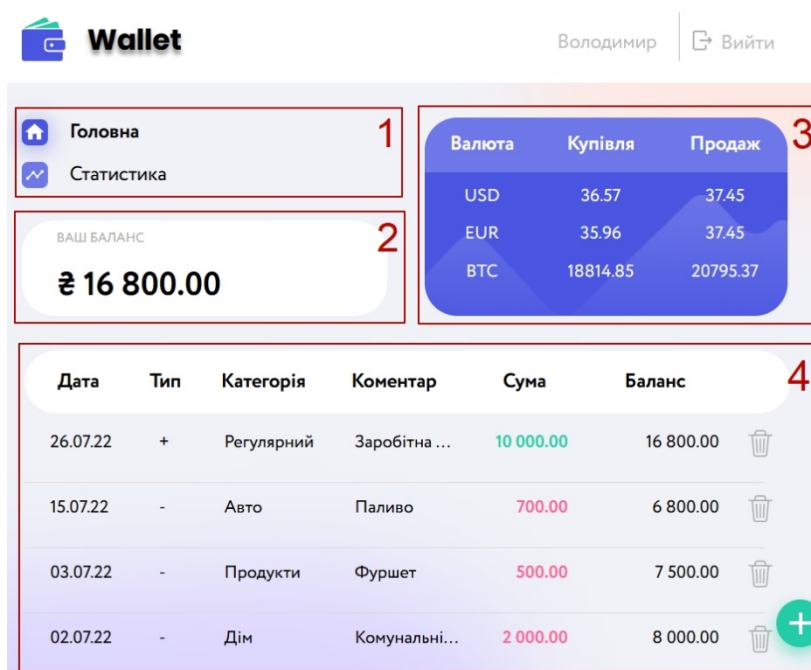


Рис. 2. Відображення сторінки «Головна» на екрані планшета

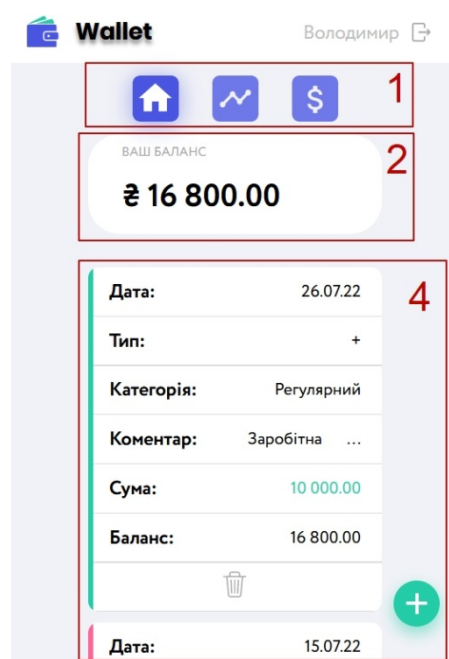


Рис. 3. Відображення сторінки «Головна» на екрані смартфона

Тобто табличний варіант представлення інформації в цьому випадку не відповідає встановленим вимогам. Для вирішення цієї проблеми було прийнято рішення змінити html розмітку даних у блоці (4) при відображенні на екрані смартфона.

На широких екранах персональних комп'ютерів та планшетів блок 4 формується як html-таблиця:

```
<table idth="100%">  
<tbody>  
<tr class="homeTabItemTabletAndDesktop">  
<td>26.07.22</td>  
<td>+</td>
```

```
<td class=""><span>Регулярний</span></td>
<td class=""><span title="Заробітна плата">Заробітна <span style="cursor: auto;"
class="more">...</span></span></td>
<td class="
homeTabItemAndDesktopResult homeTabItemAndDesktopResult--green "><span>10
000.00</span></td>
<td class=""><span>16 800.00</span></td>
<td><button class="buttonDelete" data-id="635812a13ba1427b35df8665"><svg
class="iconDelete">
<use href="/static/media/sprite.1e32b6dc.svg#buttonDel"></use>
</svg></button></td>
</tr>
...
</tbody>
</table>
```

При відображенні на екрані смартфона html-таблиця за допомогою JavaScript коду замінюється на список:

```
<ul class="homeTabList">
<li class="homeTabItem homeTabItem--green">
<p>Дата:<span>26.07.22</span></p>
<p>Тип:<span>+</span></p>
<p data-name="category" class="">Категорія:<span><span>Регулярний</span></span></p>
<p data-name="comment" class="">Коментар:<span><span title="Заробітна плата">Заробітна
<span
style="cursor: auto;" class="more">...</span></span></span></p>
<p data-name="sum" class="">Сума:<span class="homeTabItemResult"><span>10
000.00</span> </span></p>
<p data-name="balance" class="">Баланс:<span><span>16 800.00</span></span></p>
<p class="homeTabItem--center"><button class="buttonDelete" data-
id="635812a13ba1427b35df8665"><svg
class="iconDelete">
<use href="/static/media/sprite.1e32b6dc.svg#buttonDel"></use>
</svg></button></p>
</li>
...
</ul>
```

Такий варіант відображення є менш компактним, оскільки назви стовпчиків таблиці дублюються в кожному рядку списку, але він дозволяє уникнути проблеми з появою горизонтальної прокрутки.

Сторінка «Статистика» не містить елементів, що потребують зміни html розмітки для відображення на різних типах пристроїв (рис. 4-6).

Блоки (5) та (6) при відображенні на екранах комп'ютерів займають дві третіх ширини екрану, тобто знаходяться на місці таблиці (4).

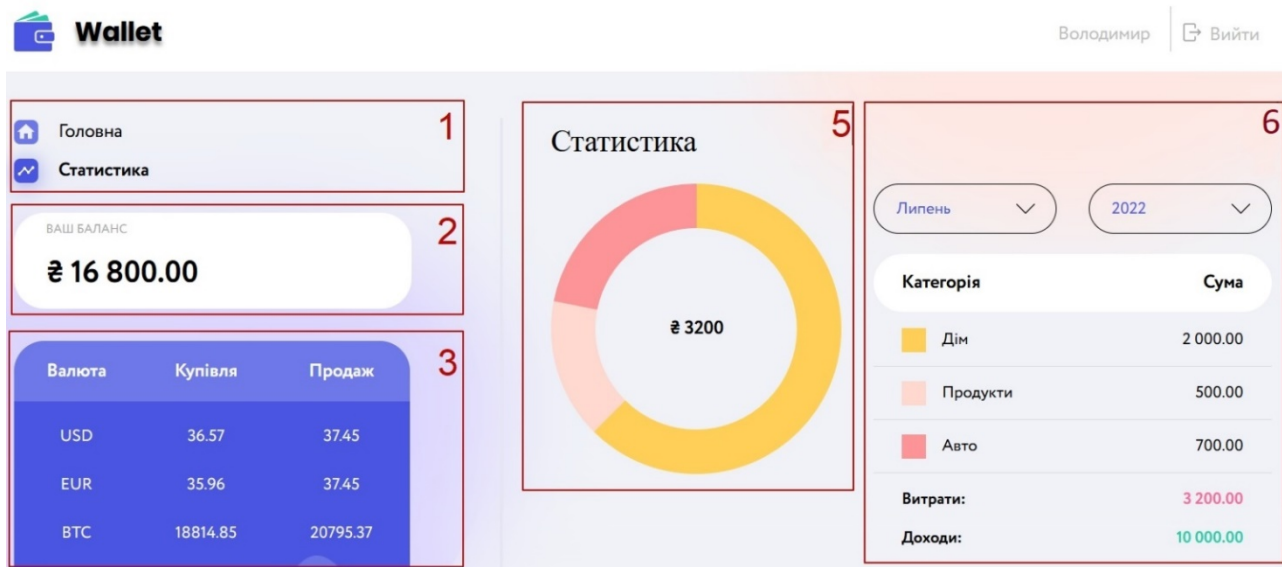


Рис. 4. Відображення сторінки «Статистика» на екрані комп'ютера

На екрані планшету ширина екрану розподіляється рівномірно (по 50%) між всіма блоками (рис. 5). На мобільних пристроях (рис. 6) змінюється відображення фільтру у блоці (6). Списки вибору місяця та року, за які формується статистика, займають всю ширину екрану та розміщуються один під одним.

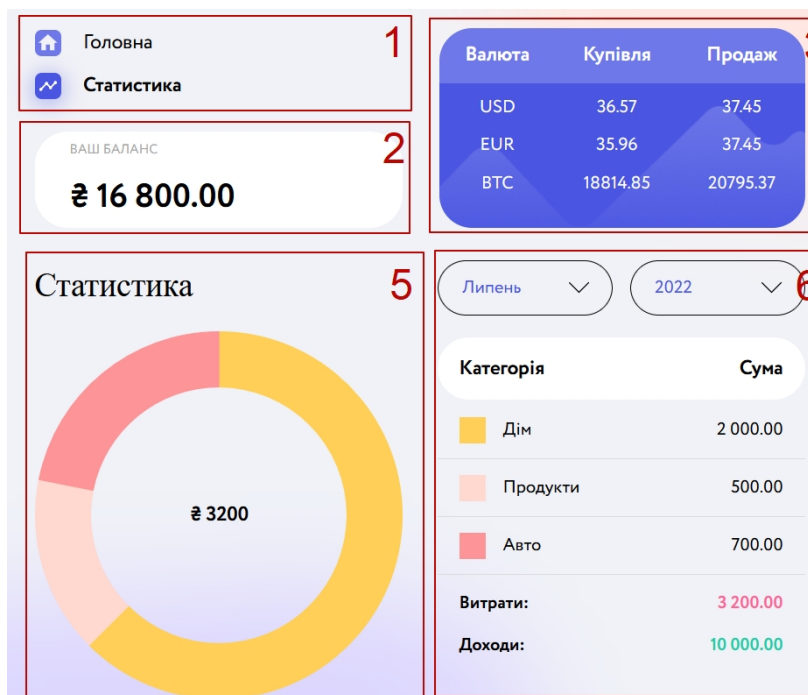


Рис. 5. Відображення сторінки «Статистика» на екрані планшета

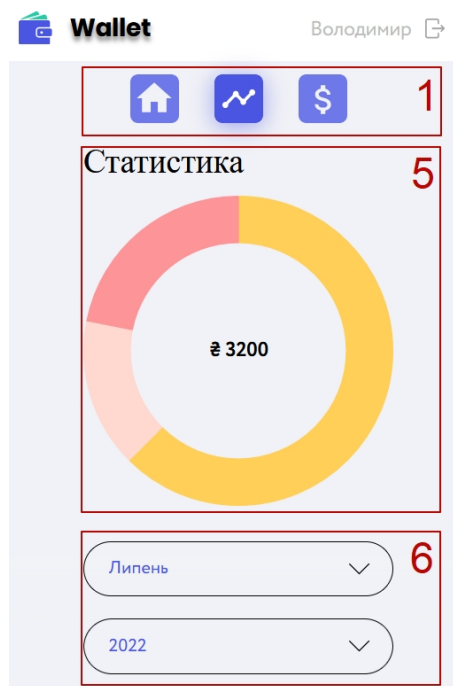


Рис. 6. Відображення сторінки «Статистика» на екрані смартфона

Висновки.

1. У роботі створено web-інтерфейс для додатку Wallet, що дозволяє вести облік та аналізувати статистику використання особистих фінансів.

2. Розроблений інтерфейс- дозволяє відображати всі елементи додатку без горизонтальної прокрутки та необхідності масштабування на екранах трьох типів пристроїв: персональних комп'ютерів, планшетів та смартфонів.

3. Встановлено, що забезпечити необхідне взаємне розміщення блоків на екранах з різною шириною можна за допомогою медіа запитів, що входять до специфікації CSS3.

4. Створено та розміщено у мережі Інтернет за адресою <https://wallet-app-1.netlify.app/> прототип розробленого додатку.

References

1. Adaptivna verстка: veb-sait [Responsive layout: website]. URL: <https://webtune.com.ua/stati/web-rozrobka/adaptivna-verstka-sajtu/#id2> [in Ukrainian].
2. Veb-interfeisy: veb-sait [Web Interfaces: Website]. URL: <https://www.alltechbuzz.net/uk/the-most-essential-frontend-web-development-tools/> [in Ukrainian].
3. Meloni, J., Kymin, J. (2018). HTML, CSS, and JavaScript All in One. 3rd edition. Sams Publishing. 800 p.
4. Ohliad CSS-freimvorkiv: veb-sait [Overview of CSS Frameworks: Website]. URL: <http://iantonov.me/page/obzor-css-freimvorkov> [in Ukrainian].
5. JavaScript (JS): Website. URL: <https://www.techopedia.com/definition/3929/javascript-js>.
6. React – A JavaScript library for building user interfaces: веб-сайт. URL: <https://reactjs.org/>.
7. Adaptivni CSS-freimvorky, sitky, klasy vydymosti: veb-sait [Responsive CSS frameworks, grids, visibility classes: website]. URL: <http://klondike-studio.ru/blog/responsive-cssframework/> [in Ukrainian].
8. Ohliad CSS-freimvorkiv: veb-sait [Overview of CSS Frameworks: Website]. URL: <http://iantonov.me/page/obzor-css-freimvorkov> [in Ukrainian].
9. Bashovyi, V. M., Statsenko, V. V., Statsenko, D. V. (2022). determining of modern frameworks speed for creating web-interfaces. *Technologies and Engineering*, 4(9), 9–16 [in Ukrainian].
10. Media Queries Level 3 website. URL: <https://www.w3.org/TR/mediaqueries-3/>.

Література

1. Адаптивна верстка: веб-сайт. URL: <https://webtune.com.ua/stati/web-rozrobka/adaptivna-verstka-sajtu/#id2>.
2. Веб-інтерфейси: веб-сайт. URL: <https://www.alltechbuzz.net/uk/the-most-essential-frontend-web-development-tools/>.
3. Meloni J., Kymin J. HTML, CSS, and JavaScript All in One. 3rd edition. Sams Publishing, 2018. 800 p.
4. Огляд CSS-фреймворків: веб-сайт. URL: <http://iantonov.me/page/obzor-css-freimvorkov>.
5. JavaScript (JS): веб-сайт. URL: <https://www.techopedia.com/definition/3929/javascript-t-js>.
6. React – A JavaScript library for building user interfaces: веб-сайт. URL: <https://reactjs.org/>.
7. Адаптивні CSS-фреймворки, сітки, класи видимості: веб-сайт. URL: <http://klondike-studio.ru/blog/responsive-cssframework/>.
8. Огляд CSS-фреймворків: веб-сайт. URL: <http://iantonov.me/page/obzor-css-freimvorkov>.
9. Башовий В. М., Стаценко В. В., Стаценко Д. В. Визначення швидкості роботи сучасних фреймворків для створення веб-інтерфейсів. *Технології та інжиніринг*. 2022. No 4(9). С. 9–16.
10. Media Queries Level 3 веб-сайт. URL: <https://www.w3.org/TR/mediaqueries-3/>.

BASHOVYI VADYM

Graduate Student, Kyiv National
University of Technologies
and Design, Ukraine

<https://orcid.org/0000-0001-7557-8841>

E-mail: Andragorn1996@gmail.com

STATSENKO VOLODYMYR

Doctor of Technical Sciences, Associate Professor,
Department of Computer Engineering and
Electromechanics, Kyiv National University of
Technologies and Design, Ukraine

<https://orcid.org/0000-0002-3932-792X>

Scopus Author ID: 57210344190

Researcher ID: C-3646-2017

E-mail: statsenko.v@knuutd.edu.ua

STATSENKO DMITRY

Candidate of Technical Sciences, Associate professor,
Department of Computer Engineering and Electromechanics
Kyiv National University of Technologies and Design, Ukraine

<https://orcid.org/0000-0002-3064-3109>

Scopus Author ID: 57210341005

ResearcherID: C-3644-2017

E-mail: statsenko.dv@knuutd.edu.ua

BASHOVYI V. M., STATSENKO V. V., STATSENKO D. V.

Kyiv National University of Technologies and Design, Ukraine

DEVELOPMENT OF ADAPTIVE WEB INTERFACE FOR PERSONAL FINANCE MANAGEMENT APPLICATION

Purpose. Creation of an adaptive web interface for a personal finance management and accounting application.

Methodology. Methods of building web interfaces, hypertext markup language (HTML), cascading style page language (CSS), web application research tools are used.

Results. The requirements for adaptive web-interfaces, which allow their use on devices with different screen parameters, have been analyzed. The components of the application web interface for accounting and analysis of personal finances use are defined. Three layout options for these elements have been developed for the screens of personal computers, tablets and smartphones. The proposed layout options allow you to avoid the appearance of horizontal scrolling on the screen and work with information without using the zoom function. It has been established that media requests, which are included in the CSS3 specification, allows to flexibly manage the relative placement of interface elements and their appearance. The transformation of tabular information into the form of lists for display on narrow smartphone screens was carried out by changing the HTML markup. A prototype of the proposed application was created and placed on the Internet.

Scientific novelty. The possibilities of media queries included in the CSS3 specification were analyzed from the point of view of managing the relative position of web interface components.

Practical significance. A prototype of the personal finance web interface management and accounting application has been developed, which allows you to avoid the appearance of horizontal scrolling on the screen and work with information without using the zoom function.

Keywords: web-application; web-interface; HTML; CSS; JavaScript; framework.