МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет “Києво-Могилянська академія”

Факультет інформатики

Кафедра інформатики

КУРСОВА РОБОТА

освітній ступінь – бакалавр

на тему: **“USER-CENTERED ДИЗАЙН ІНТЕРАКТИВНИХ СИСТЕМ”**

Виконала: студентка 3-го року навчання,
Освітньої програми «Комп’ютерні науки», 122

Станіславська Катерина Євгеніївна

Керівник Афонін А.О.,_____
кандидат фізико-математичних наук

Рецензент _____
(прізвище та ініціали)

Курсова робота захищена
з оцінкою _____

Секретар ЕК _____
«____» _____ 2023 р.

Київ – 2023

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
проф., д.ф-м.н.
_____С. С. Гороховський
(підпис)
„____ ”_____2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студентці 3-го курсу, факультету інформатики
Станіславській Катерині Євгеніївні___

ТЕМА: User-Centered дизайн інтерактивних систем

Вихідні дані:

- Критерії розробки інтерактивних систем, орієнтованих на користувача
- Прототипи, розроблені за даними критеріями для двох різних цільових груп користувачів

Зміст ТЧ до курсової роботи:

Зміст
Анотація
Вступ
1.User-Centered дизайн
1.1 Поняття дизайну, орієнтованого на користувача
1.2 Проблеми та критерії розробки
1.3 Сучасні вимоги до взаємодії системи та користувача
2. Інтерактивний дизайн
2.1 Типи взаємодії системи та користувача
2.2 Етапи інтерактивності
Висновок
Список літератури

Дата видачі „___” _____ 2023 р. Керівник _____
(підпис)
Завдання отримав _____
(підпис)

ЗМІСТ

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ **3**

ЗМІСТ **5**

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ **6**

АНОТАЦІЯ **7**

ВСТУП **8**

РОЗДІЛ 1: USER-CENTERED ДИЗАЙН **10**

1.1 Поняття дизайну, орієнтованого на користувача 10

1.2 Проблеми та критерії розробки 10

1.3 Сучасні вимоги до взаємодії системи та користувача 13

РОЗДІЛ 2: ІНТЕРАКТИВНИЙ ДИЗАЙН **16**

2.1 Типи взаємодії системи та користувача 16

2.2 Етапи інтерактивності 19

РОЗДІЛ 3: ІНТЕРАКТИВНА СИСТЕМА, ОРІЄНТОВАНА НА РІЗНІ ТИПИ КОРИСТУВАЧІВ **29**

3.1 Взаємодія першої групи користувачів та соцмережі для пошуку медіаресурсів для натхнення Inspo 30

3.2 Взаємодія другої групи користувачів та додатку для доставки їжі Eatery 41

ВИСНОВКИ **50**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **51**

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	01.09.2022	
2.	Пошук та огляд літератури за темою роботи.	26.12.2022	
3.	Визначення основних критеріїв розробки на основі літератури.	07.01.2023	
4.	Написання теоретичної частини.	10.02.2023	
5.	Аналіз отриманих результатів з науковим керівником.	15.02.2023	
6.	Коригування роботи.	01.03.2023	
7.	Реалізація прототипів з мінімальним функціоналом.	30.03.2023	
8.	Аналіз прототипів з науковим керівником.	10.04.2023	
9.	Покращення прототипів	30.04.2023	
10.	Створення презентації до захисту.	10.05.2023	

АНОТАЦІЯ

Робота присвячена розробці дизайну, орієнтованих на користувача, та його впливу на досвід користувача при взаємодії з інтерактивними системами. Робота аналізує проблеми та критерії розробки дизайну, стандартизує принципи та вимоги до дизайну систем, орієнтованих на користувача. Наведено приклади прототипів системи реєстрації для різних груп користувачів і підкреслюється важливість user-centric підходу до дизайну для різних груп користувачів.

ВСТУП

Дизайн це важлива складова технологій, адже він формує спосіб людської взаємодії з продуктом та його сприйняття. Продукт з якісним дизайном забезпечує позитивний досвід користувача. Своєю чергою недостатньо пропрацьований дизайн призводить до фрустрацій та збентеження особи, що користується технологією, а також краху продукту загалом.

На сьогодні існують різні підходи до розробки дизайну продукту, в залежності від конкретного проєкту, команди та цілей. Задля забезпечення продуктивної взаємодії користувача з інформаційною системою використовують наступні підходи: розробка зручності використання (Usability Engineering), дизайн, орієнтований на користувача (User-Centered Design), спілий дизайн (Participatory design), інформаційна архітектура, доступний дизайн (Accessibility design) та дизайн досвіду користувача (User Experience Design). Хоча і перелічені методи мають відмінності, межі та засоби їх застосування часто перетинаються та є спільними для всіх підходів.

Актуальність дизайну, орієнтованого на користувача для інтерактивних систем потребує особливої уваги в сучасному цифровому світі. З постійним розширенням можливостей технологій і зростанням кількості користувачів, важливо забезпечити ефективну і зручну взаємодію між людиною і цифровою машиною.

Метою роботи є аналіз взаємодії між користувачем і системою, а саме технічне забезпечення дотримання критеріїв, які впливають на досвід користувача, а також методів забезпечення потреб різних типів користувачів на кожному етапі взаємодії для різних типів систем.

Досягнення цієї мети передбачає вирішення наступних **завдань**:

- схарактеризувати проблеми та критерії розробки дизайну на основі джерельної бази
- стандартизувати принципи та вимоги до дизайну систем, орієнтованих на користувача
- зазначити основні типи інтерактивності та проаналізувати етапи взаємодії між системою та користувачем
- продемонструвати важливість user-centric підходу до дизайну для різних груп користувачів на основі прототипів системи реєстрації

Об'єктом дослідження є взаємодія між користувачем і системою. **Предмет** дослідження – критерії, що забезпечують унікальний досвід користувача, методи задоволення потреб різних груп користувачів на основі стандартів до розробки інтерактивного дизайну, а також прототипи систем.

Структура курсової роботи оформлена відповідно до мети та наукових завдань: складається з трьох розділів, висновків та списку використаних джерел.

РОЗДІЛ 1: USER-CENTERED ДИЗАЙН

1.1 Поняття дизайну, орієнтованого на користувача

User-centered design (UCD), або дизайн, орієнтований на користувача є найбільш всеосяжним підходом до розробки дизайну технологічних продуктів, адже центральним об’єктом цієї структури є кінцевий користувач. Це означає, що фокус на кожному етапі розробки продукту зосереджений на задоволенні потреб користувачів, вирішенні їх проблем та болей. Дизайн, орієнтований на користувача, належить до ряду процесів проєктування, в яких кінцеві користувачі відіграють впливову роль у формуванні фінального дизайну. Як визнають Абрас, Малоні-Крічмар та Пріс у своїй роботі, рівень залученості користувача в процес розробки UCD може різнитися в залежності від типу продукту, його поширення, бізнес-цілей установи, що стоїть за цим продуктом та інших факторів, проте ключова ідея полягає в тому, що юзери беруть участь у створенні дизайну певною мірою [1].

1.2 Проблеми та критерії розробки

User-Centered дизайн набув популярності через працю Дональда Нормана “User Centered System Design: New Perspectives on Human-computer interaction”, опубліковану у 1986, хоча термін «дизайн, орієнтований на користувача» був введений Робом Клінгом у праці “The Organizational Context of User-Centered Software Designs” ще у 1977 році.

Клінг описує користувачів як “забутих” людей в розробці програмного забезпечення [2]. За майже 50 років, що пройшли з моменту публікації його статті, можна спостерігати позитивні зміни у контексті піклування про потреби користувачів у процесі дизайну технологій. Однак і сьогодні питання забезпечення балансу між технічними вимогами, такими як ефективність

програми (runtime efficiency) та увагою до потреб користувача залишається актуальним.

У праці автор наводить порівняння фокуса на користувачеві та на машині, тобто комп'ютері або іншому пристрої обробки даних (рис. 1.1). Наголошується на проблемі дегуманізації комп'ютеризованих систем, а також на тому, що "люди, які вступають у контакт з інформаційною системою, варті пильної уваги під час проєктування - чи то керівники, прості робітники чи клієнти організації" [2].

USER-CENTERED	MACHINE-CENTERED
1. Systems are valued that increase personal competence and pride in work.	1. Efficiency is emphasized.
2. People are accepted as non-rational and error-prone.	2. Systems are designed in purely functional terms.
3. Jobs are designed to be personally satisfying. Automated procedures are designed to fit job needs.	3. Human error is not tolerated. Systems are not error tolerant and provide limited diagnostic information.
4. The burden of precision is placed on the machine. Systems are forgiving.	4. Users are forced to match the precision required by the machine.
5. Users easily obtain/create systems that meet their needs.	5. Jobs and procedures are designed to simplify machine processing.
6. Users can initiate, veto, and collaborate in system designs.	6. Human relations are ignored as long as the job gets done.
7. Designs and assumptions are intelligible to users through appropriate technique (modular structures) and clear documentation.	7. System designs are imposed on users. They initiate but never veto system designs.

Рис. 1.1 Характеристики середовищ, орієнтованих на користувача та машину

Аби вирішити проблеми взаємодії комп'ютеризованої машини та людини, необхідно окреслити принципи, за якими відбуватиметься зміна пріоритету від виключно ефективності роботи машини до задоволення потреб користувача, що є об'єктом взаємодії з системою. У праці [3] Теодор Стерлінг ідентифікує основні критерії до дизайну систем, що забезпечують задовільний користувальницький досвід. Серед них:

1. *Мова системи повинна бути зрозумілою*, щоб користувач міг легко читати та розуміти "вихідні дані", які отримує. Необхідно уникати використання спеціальних кодів та скорочень, якщо це можливо.

2. *Операції з системою повинні бути проведені "ввічливо"*. Хоча поважний тон інтерфейсу не є гарантією якості обслуговування, він є необхідною умовою гуманного ставлення до користувачів.

3. *Система має швидко реагувати.* У випадку, коли система є посередником у розподілі товарів або послуг, її розробка повинна бути спрямована на швидке виконання заходів. При вимушених затримках у роботі системи, користувач повинен отримати негайне повідомлення як відповідь на свої дії.

4. *Система має позбавити користувача від зайвих клопотів.* Комп'ютеризовані системи повинні позбавляти користувача непотрібної роботи. Вхідна інформація, яка не є необхідною для потреб обробки інформації, або яка може бути надана з уже збереженої інформації не має вимагатись від користувача.

5. Для системи важливо мати *"людський" інформаційний інтерфейс*, оскільки дії системи часто можуть бути незрозумілими для тих, на кого вона впливає. Користувачам можуть знадобитися пояснення або заспокоєння, особливо тим, хто перебуває у вразливому стані.

6. *Система повинна надавати можливості для виправлення помилок*. Якщо певні помилки виникають часто, у систему можна інтегрувати процедури, щоб виправити їх у разі виявлення. Крім того, існує потреба в загальних процедурах, які можуть усунути непередбачені помилки.

7. Щоб гарантувати, що *система реагує на потреби людей*, може виникнути необхідність притягнути розробника та/або керівництво до

відповідальності за неприємності, заподіяні користувачеві. Інакше, система може не реагувати або не виправляти свої помилки, що призведе до негативного користувальницького досвіду.

Критерії наведені вище є підґрунтям для забезпечення якісного інтерактивного дизайну, тобто взаємодії системи та людини з технічної точки зору та психологічного впливу інтерфейсу на користувача.

1.3 Сучасні вимоги до взаємодії системи та користувача

Зі значним зростанням кількості інтерактивних систем протягом останніх 50 років, вимоги для забезпечення позитивного користувальницького досвіду деформувались та були доповнені, порівнюючи з роботами Роба Клінга та Теодора Стерлінга. Саме через таку велику кількість технологій та підходів до їх розробки, вимоги до дизайну систем, орієнтованих на користувачів є досить розмитими у літературі: терміни накладаються на один одного, дизайн принципи підмінені дизайн процесом, а також цілі конкретних підходів чітко не визначені.

Задача цього розділу – стандартизувати принципи та вимоги до розробки дизайн систем, орієнтованих на користувача. Аби окреслити їх, необхідно зрозуміти яким має бути дизайн. Щоб описати інтерфейс, який задовольняє потреби користувача, часто використовують термін usability, тобто зручність використання. Відповідно до праці [4] юзабіліті – це "простота використання та прийнятність системи або продукту для певного класу виконання конкретних завдань у певному середовищі, де «простота використання» впливає на продуктивність і задоволення користувача, а «прийнятність» впливає на те, чи використовується продукт."

Аби мати змогу виміряти “продуктивність і задоволення користувача”, необхідно зрозуміти вимоги до системи, аби вона була зручною для

використання. Беручи до уваги проблеми кінця 20 століття, до яких звертаються раніше згадані Роб Клінг та Теодор Стерлінг, а також сучасні стандарти [5-7] можна визначити такі складові юзабіліті:

1. *Здатність до навчання (learnability):* дозволяє новим користувачам легко опанувати систему.
2. *Ефективність використання:* описує як швидко користувач виконує задачі та досягає своєї мети під час роботи із системою.
3. *Запам'ятовуваність (memorability):* належить до спроможності користувачів, які не часто використовують систему, легко орієнтуватися, як оперувати даною системою. Система має бути побудована таким чином, аби "дозволити людям розпізнавати інформацію в інтерфейсі, а не змушувати їх запам'ятовувати її."[8] Один із методів покращення здатності до навчання - збереження консистентності (постійності) системи.
4. *Обробка помилок.*
    - Вимірює частоту скоєння помилок користувачами, наприклад, випадкове видалення певної інформації, неправильний формат вводу даних
    - Дозволяє особі "відновитися" після скоєної помилки та знайти вихід, наприклад, за допомогою наявності функції "Скасувати".
    - Запобігання частих помилок системою, мінімізація ризику та небажаних наслідків помилок.
5. *Суб'єктивне задоволення:* описує сприйняття, почуття та думки користувача - тобто його досвід щодо використання продукту.
6. *Доступність (accessibility):* ставить питання "Чи можуть користувачі з обмеженими можливостями ефективно використовувати систему". Це може означати, наприклад, створення інтерфейсу, який підтримує великі шрифти для людей з вадами зору, або надання можливості використовувати систему голосовим вводом для людей з обмеженою рухливістю.

7. *Інтуітивність:* належить до інтуїтивного розуміння мови, архітектури, навігації системи та використання її без особливих зусиль. Запровадження необхідної доступної документації та підказок.

8. *Чуйність (responsiveness):* описує необхідність для користувача отримання зворотної відповіді та статусу від системи. Природний діалог між системою та користувачем

9. Аби система була зручною та ефективною для користувачів, сьогоднішні реалії також вимагають *відповідність стандартам безпеки*, які можуть бути непомітними для користувача, проте є важливими для його позитивного досвіду. Наприклад, система, яка обслуговує медичну сферу, повинна дотримуватись високих стандартів безпеки, щоб запобігти неприємностей для пацієнтів.

Підсумовуючи, задача розробки систем – зробити її орієнтованою на користувача та зручною для використання. Для вимірювання зручності застосовують такі вимоги як здатність до навчання, ефективність використання, запам'ятовуваність, обробка помилок, суб'єктивне задоволення, доступність, інтуїтивність, чуйність та безпека.

РОЗДІЛ 2: ІНТЕРАКТИВНИЙ ДИЗАЙН

2.1 Типи взаємодії системи та користувача

Спираючись на загальні критерії до розробки систем, орієнтованих на користувача, перелічені в попередньому розділі, а також кінцеву мету їх проєктування – зручність користування – варто розуміти, яким способом можна задовольнити дані вимоги.

Інтерактивний дизайн фокусується на розумінні потреб користувача та займається проєктуванням технологій, що забезпечують якісну взаємодію людини та інтерфейсу системи.

Обов'язковими етапами взаємодії між системою та користувачем є ввід даних користувачем та вивід від системи, тобто її реакція. Будь-який ввід інформації від людини, чи то текст, голосове повідомлення, графіка, конвертуються згодом у потік команд, що обробляються операційною системою. Проте, аби цей backend процес був максимально непомітним для користувача, варто зазначити типи "прямої" комунікації між системою та користувачем і сфокусуватися на user-friendly підході до кожного з них. Під прямою комунікацією мається на увазі така взаємодія, при якій реакція від системи залучає один або декілька чуттів людини:

- зір (графічна реакція),
- слух (аудіо- відповідь),
- дотик (haptic feedback),
- нюх
- смак

Водночас ввід даних у систему від людини здійснюється найчастіше дотиково (напр. натисканням клавіш), або ж голосом, за використання мікрофона. Отже, мета цього розділу – окреслити типи взаємодії системи та користувача із залученням графічних, тактильних та слухових елементів.

1. *Маніпулювання графічним інтерфейсом користувача (GUI)*

Ця комунікація включає усі типи користувальницької взаємодії з цифровими об'єктами на екрані інтерфейсу [9], що передбачає графічну відповідь, де пристроєм виводу слугує дисплей; та аудіо- фідбек пристроєм виводу для якого слугують колонки. GUI не є винятковим для комп'ютерів, а й використовуються у портативних електронних пристроях, наприклад, смартфонах, ігрових пристроях, портативних медіа- та аудіо- плеєрах та інших системах керування.

Девайси вводу даних включають клавіатуру, мишку, тачпад, геймпад, джойстик, пульт управління, мікрофон, сенсорний екран та інше.

- Графічна та звукова реакція

Найочевиднішим прикладом взаємодії користувача з GUI є буденна робота з комп'ютером. За даної взаємодії користувач найчастіше маніпулює елементами, зображеними на дисплеї та отримує візуальний фідбек. Це також включає введення інформації, що як наслідок з'являється на дисплеї. Прикладом може слугувати робота з текстовими та графічними редакторами, заповнення форм, переглядом медіа- та відео- контенту тощо.

Інший спосіб взаємодії з графічним інтерфейсом – через меню, де користувачу надається набір опцій. Прикладом такої взаємодії можуть бути вбудовані системи, такі як банкомати та термінали, при комунікації з інтерфейсом яких, користувач натискає опцію з меню, а система реагує відповідно, видаючи відповідь на екрані, що може також супроводжуватись звуковим сигналом.

2. *Маніпулювання командами (CLI)*

Маніпулювання командами здійснюється за допомогою інтерфейсу командного рядка (CLI), який є "предком" GUI. Фактично, як і в GUI, вивід

системою здійснюється за допомогою екрана інтерфейсу, забезпечуючи користувачеві найчастіше графічну реакцію після введення команд для виконання задач. Проте, введення кодів користувачем обмежене командним рядком або терміналом.

3. *Обробка природної мови (Natural Language Processing)*

Обробка природної мови передбачає спілкування користувача із системою природною мовою. Прикладом є комунікація з віртуальним помічником, чат-ботом, або з AI-асистентом, що набуває особливої популярності зараз. Що стосується даних, які система приймає на вхід, незалежно від типу вводу користувачем, чи то голосом, або з використанням клавіатури, сигнал трансформується у текстовий масив, аби бути обробленим системою надалі.

- Графічна реакція

Ця категорія також межує з графічним інтерфейсом користувача, адже серед інструментів виводу даних також присутній графічний інтерфейс.

- Звукова реакція

Найкраще аудіо-фідбек можна описати прикладом комунікації людини з голосовими помічниками, такими як Siri, Alexa та інші, що видають звукову реакцію у відповідь на запит природною мовою від людини.

4. *Маніпулювання жестами*

При роботі з інтерфейсами, що базовані на дотику, тобто мають сенсорний дисплей, застосування жестів є ключовим методом взаємодії. Користувач натискає, свайпає та маніпулює елементами дисплея в інший спосіб, на що найчастіше отримує візуальний або звуковий фідбек.

- Тактильна реакція

Взаємодія екрана інтерфейсу та користувача може супроводжуватися тактильною відповіддю від системи теж. Наприклад, перетягування іконок застосунків на домашньому екрані мобільних пристроїв Apple супроводжується вібрацією.

5. *Доповнена та віртуальна реальність (AR та VR)*

Коли питання стосується AR та VR, взаємодія системи та користувача може здійснюватись за допомогою маніпулювання всіма чуттями людини: зором, слухом, дотиком, нюхом та навіть смаком. Це означає, що можуть бути застосовані різноманітні типу вводу даних користувачем. Так само і система може надавати графічний, звуковий, тактильний та інші типи реакції окремо, або одночасно.

2.2 Етапи інтерактивності

Етапи взаємодії між системою та користувачем можуть відрізнятися в залежності від типу взаємодії, проте в загальному вона описується чотирма компонентами:

- Користувач,
- Ввід,
- Система,
- Вивід,

при чому ввід та вивід разом формують Інтерфейс [10]. При кожній дії користувача відбувається новий цикл взаємодії чотирьох вищезазначених компонентів, забезпечуючи трансляції між ними (див. рис 2.1).

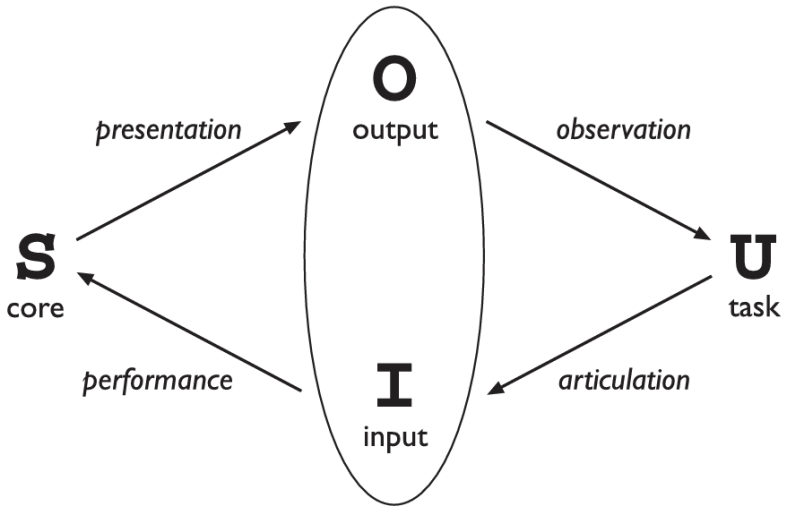

Рис. 2.1 Перехід між компонентами взаємодії

Ефективність взаємодії користувача та системи розглядатиметься з точки зору задоволення критеріїв юзабіліті, описаних у розділі 1.3. У праці [10, с. 201] зазначено наступні тактики забезпечення виконання вимог щодо зручності використання системи: «ініціатива користувача», «ініціатива системи» та «змішана ініціатива». Автори наводять наступний приклад цих етапів взаємодії:

"При скасуванні команди користувач має “ініціативу користувача” *скасувати*, і система надає відповідь. Проте під час скасування система може запропонувати “ініціативу системи”, показуючи індикатор прогресу. Таким чином, скасування може продемонструвати “змішану ініціативу”. [10, с. 201]"

Тобто, “змішана ініціатива” характеризується не тільки наявністю виводу системи, але й зворотним зв’язком для користувача. Своєю чергою, такі функції як cancel, undo, pause/resume, та aggregate допомагають користувачеві виправити помилку, або уникнути її [10, с. 202]. Отже, можна вважати, що “змішана ініціатива” є однією з ознакою якісної взаємодії користувача та системи.

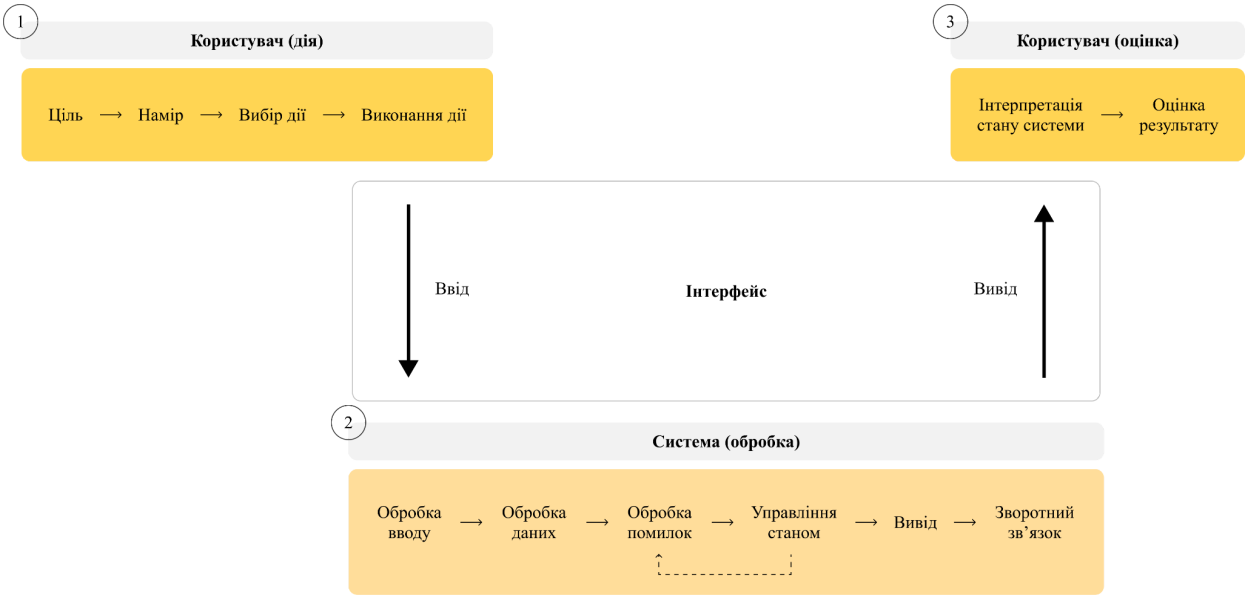

Рис 2.2 Загальні етапи взаємодії системи та користувача

Умовно взаємодію системи та користувача можна поділити на 3 етапи: дія, обробка та оцінка, де виконавцем цих етапів є користувач та система згідно з схемою. До того ж Інтерфейс регулює ввід даних від користувача та вивід від системи.

Варто зазначити, що етапи взаємодії не завжди строго послідовні, і між проміжними етапами також може бути взаємодія, або цикл взаємодії. Наприклад, система може оновлювати свій внутрішній стан кілька разів під час обробки одного вводу або вона може генерувати вивід і зворотний зв’язок одночасно.

Розглянемо співдію окреслених вище етапів та як забезпечується юзабіліті на кожному з них на прикладі використання сервісу "GitLab" користувачем вперше. У даному аналізі етапів взаємодії ми будемо використовувати терміни та їх аргументацію, посилаючись на працю Дональда Нормана [11].

- Етап 1 (Дія користувача)

1. *Формування цілі:*

Нехай користувач ставить за мету створити власний репозиторій на "GitLab", аби відстежувати версії проєкту. GitLab пропонує почати роботу з авторизації або реєстрації, якщо особа ще немає акаунту (рис. 2.3). Отже, система *легка для вивчення (learnability) та інтуїтивно зрозуміла,* адже дає чітко зрозуміти різницю між реєстрацією та авторизацією для нових користувачів за а) наявністю відповідних заголовків та підписів, б) різницею у кількості та значенні полів вводу.

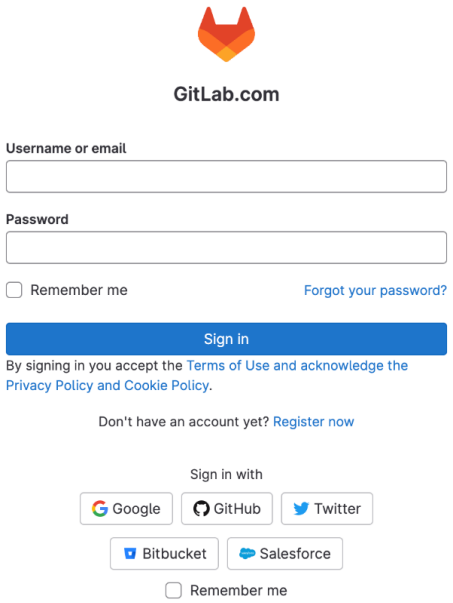

Рис 2.3 Сторінка авторизації

2. *Формування наміру*

Користувач має намір авторизуватись або зареєструватись на платформі, аби досягти кінцевої цілі.

3. *Вибір дії*

Користувач обирає опцію “авторизуватися через Google”, адже за його попереднім досвідом це найшвидший варіант реєстрації, а отже *найефективніший*. Дійсно, при авторизації через Google акаунт, користувачеві не потрібно вводити email, ім’я тощо, достатньо обрати бажаний обліковий

запис. Завдяки системі ключів та паролів забезпечується вимога *запам'ятовуваності*, інтегрована авторизація (рис 2.4) не потребує введення пароля повторно.

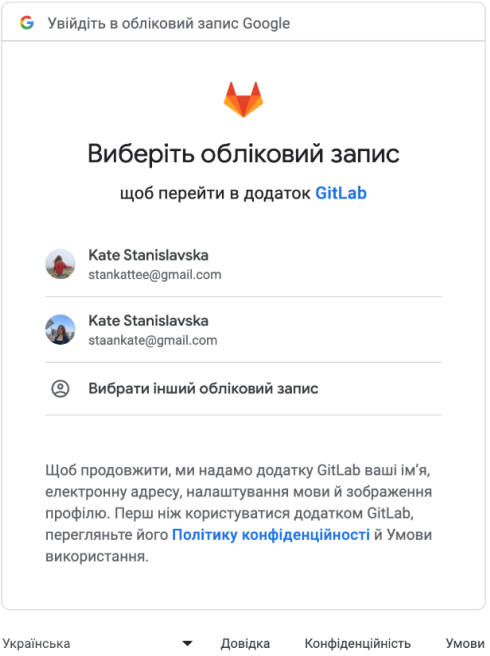

Рис 2.4 Інтегрована авторизація через Google акаунт

З безпекових цілей GitLab запрошує верифікаційний код (рис 2.5), надісланий на пошту облікового запису, через який було розпочато авторизацію.

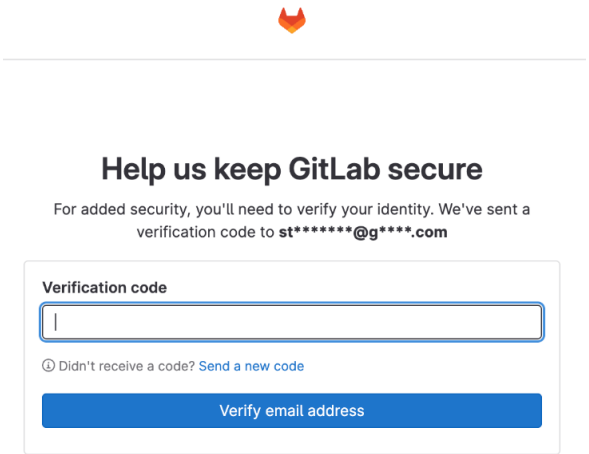

Рис 2.5 Сторінка для введення верифікаційного коду

- Етап 2 (Обробка системою)

Варто зазначити, що при кожній новій дії залучені всі етапи взаємодії між системою та користувачем - дія, обробка, оцінка. Проте робота з полями є найбільш показовим прикладом переходу одного етапу в інший.

Отже, у поле для верифікаційного коду (див. рис 2.5), користувач має намір ввести код, який мав би бути надісланий на email користувача. Через внутрішню помилку, код не було отримано особою, а отже вона приймає рішення запросити код повторно за допомогою кнопки “Send a new code”. Обробка запиту відбувається наступним чином:

1. *Обробка вводу:* система отримує вхідні дані від користувача у вигляді події натискання кнопки «Send a new code».
2. *Обробка даних:* система обробляє вхідні дані та визначає, що потрібно надіслати новий код підтвердження на електронну пошту користувача.
3. *Управління станом:* система оновлює свій внутрішній стан, щоб відобразити, що новий код підтвердження був запитаний і потребує надіслання користувачеві
4. *Вивід:* система генерує вихідні дані за допомогою попередньо визначеного шаблону та динамічних даних, отриманих під час обробки, а саме електронної адреси, та надсилає код (рис. 2.6).

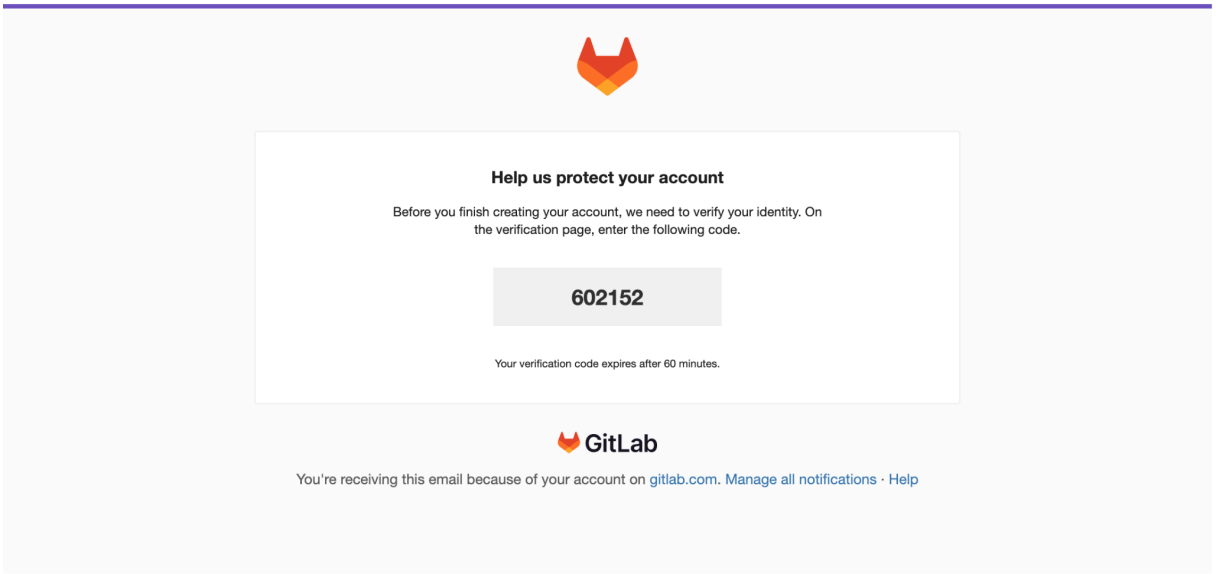

Рис. 2.6 Верифікаційний код надісланий на пошту користувачеві

5. *Зворотний зв'язок:* система забезпечує зворотний зв'язок з користувачем у вигляді сповіщення на інтерфейсі користувача, інформуючи його про те, що новий код надіслано. Наявність зворотного зв'язку від системи є ознакою її юзабіліті, а конкретно *чуйності (responsiveness).*

6. *Моніторинг:* система реєструє відповідні дані та події під час взаємодії, такі як подія натискання кнопки, згенероване сповіщення та інші повідомлення про помилки чи сповіщення, що виникли.

Наступною ітерацією дій користувача є введення інформації у поле, а саме, верифікаційного коду, отриманого на пошту (див. рис. 2.6). Користувач вводить код, проте допускається помилки при введенні й натискає кнопку "Verify email address". Система реагує відповідним чином:

1. *Обробка вводу:* система отримує вхідні дані від користувача у вигляді коду підтвердження, введеного в призначене поле введення, і подію натискання кнопки «Verify email address».

2. *Обробка даних:* система обробляє вхідні дані та намагається перевірити збіжність email адреси та коду, надісланого на нього. У такому випадку система визначає, що введений код перевірки неправильний або недійсний.

3. *Управління станом:* система оновлює свій внутрішній стан, щоб відобразити, що введений код перевірки є неправильним або недійсним.

4. *Обробка помилок:* система обробляє помилку, генеруючи повідомлення про проблему. Система також надає вказівки для користувача щодо виправлення помилки (рис. 2.7 та 2.8). Така реакція дозволяє користувачеві *оговтатись від помилки* та мати змогу її виправити, що свідчить про *чуйність* системи та загалом зручність її використання.

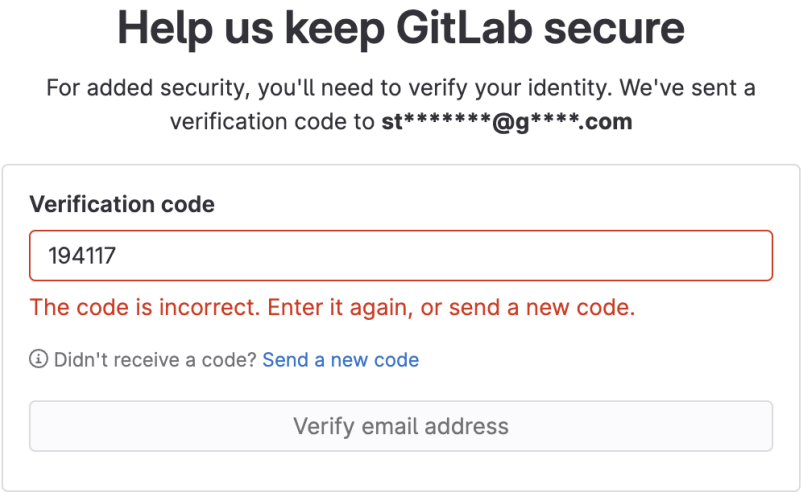

Рис. 2.7 Неправильний або недійсний верифікаційний код

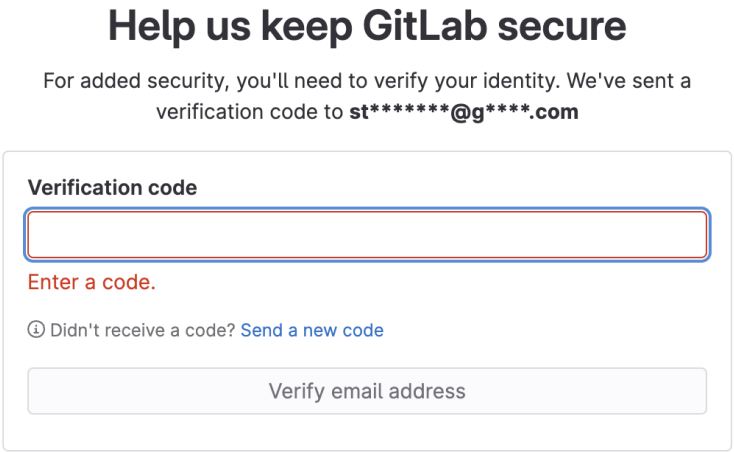

Рис. 2.8 Відсутність обов'язкового вводу даних у поле

5. *Вивід та зворотний зв'язок:* система надає зворотний зв'язок користувачеві, вказуючи, що сталася помилка та що введений код підтвердження є неправильним або недійсним (див. рис. 2.7 та 2.8). Важливим сигналом є також зміна кольору кнопки «Verify email address» із синього на сірий, що надає користувачеві *інтуїтивне розуміння* її неактивного стану, допоки введені дані не є коректними.

6. *Моніторинг:* система реєструє відповідні дані та події під час взаємодії, наприклад введений код підтвердження, згенероване повідомлення про помилку.

Варто зазначити також дотримання *стандартів безпеки* для користувача у даному прикладі, виявлене у приховуванні повної поштової адреси.

- Етап 3 (Оцінка користувачем):

1. *Інтерпретація стану системи*

Після допущення помилки користувачем при введенні верифікаційного коду система відреагувала на це належним чином, вивівши повідомлення про помилку, а також змінивши стан поля вводу через зміну кольору на червоний. Дані індикатори легко розпізнаються користувачем, як такі, що вказують на наявність помилки при запиті, завдяки *інтуїтивності* системи.

2. *Оцінка результату*

Наступним кроком користувач починає шукати шляхи вирішення проблеми, а отже шукає *способи відновлення* після допущеної помилки. У цьому випадку користувач може обрати один із наступних шляхів вирішення проблеми:

a) Перевірити відповідність верифікаційного коду, що був надісланий на електронну пошту та того, що був введений у поле. Також, користувач може скопіювати код або ввести його заново.
b) Запросити новий код натиснувши кнопку “Send a new code”. Її наявність, як вказівка для користувачів, у яких виникли проблеми із кодом підтвердження, забезпечує *чуйність* системи та орієнтованість на потреби користувача.

Підсумовуючи, загальні критерії розробки систем, орієнтованих на користувача, передбачають зручну взаємодію між людиною та системою. Інтерактивний дизайн зосереджується на розумінні потреб користувача та створенні якісної взаємодії. Взаємодія може відбуватись через графічний інтерфейс користувача (GUI), командний рядок (CLI), обробку природної мови (Natural Language Processing), жести, тактильну відповідь та доповнену/віртуальну реальність (AR/VR). Кожен тип взаємодії має свої особливості та може включати графічну, звукову та інші форми реакції від системи на користувацькі дії. Взаємодія включає чотири компоненти: користувач, ввід, систему та вивід, при чому інтерфейс регулює ввід даних від користувача та вивід від системи. Кожна дія користувача спричиняє новий цикл взаємодії між компонентами, а ефективність взаємодії оцінюється з точки зору юзабіліті. Серед тактик забезпечення зручності використання системи вирізняють "ініціативу користувача", "ініціативу системи" і "змішану ініціативу". Взаємодію системи та користувача можна поділити на етапи: дія, обробка та оцінка. Етапи взаємодії можуть бути нестрого послідовними, і між ними може відбуватися взаємодія або цикл взаємодії.

РОЗДІЛ 3: ІНТЕРАКТИВНА СИСТЕМА, ОРІЄНТОВАНА НА РІЗНІ ТИПИ КОРИСТУВАЧІВ

У попередніх розділах було досліджено критерії забезпечення якісної взаємодії користувача та системи, види інтерактивності, а також її етапи. З урахуванням попередніх вимог до розробки, метою даного розділу є демонстрація відмінності у підходах до створення інтерактивної системи, що орієнтована на різні групи користувачів на прикладі двох прототипів для відповідно двох різних цільових аудиторій. У даних прототипах наявна лише графічна реакція від системи з дотиковим вводом від користувача (натискання клавіш на клавіатурі, мишка, тачпад, сенсорний екран), отже прототипи характеризуються як представники GUI.

Інструментами створення даних прототипів є фреймворк JavaScript – React JS, а також застосування стилів CSS Tailwind. Використано авторизації через Google із застосуванням firebase[] та вбудовано анімації[].

Першим кроком у взаємодії користувача і системи є дія користувача, яка розпочинається *формуванням мети* взаємодії із системою. Цільовими групами у дослідженні виступають дві аудиторії, що переслідують наступні цілі:

- Перша група: знайти референси, тобто фотографії або інші графічні зображення для натхнення, фотосесій, подорожей, кулінарного мистецтва тощо, або ж просто для естетичного задоволення за допомогою соцмережі.
- Друга група: замовити страву через застосунок для доставки їжі.

Об’єктами взаємодії відповідно виступають соцмережа Inspo та застосунок для доставки їжі Eatery. Предметом дослідження слугуватимуть форми реєстрації на дані платформи.

3.1 Взаємодія першої групи користувачів та соцмережі для пошуку медіаресурсів для натхнення Inspo

Аби досягти своєї *мети* – знайти референси для натхнення – користувачі першої групи мають *намір* розпочати взаємодіяти із платформою. Серед наявних варіантів для початку роботи є вхід в акаунт ("Log In") та реєстрація ("Sign Up" або "Become a new member"), рис 3.1.

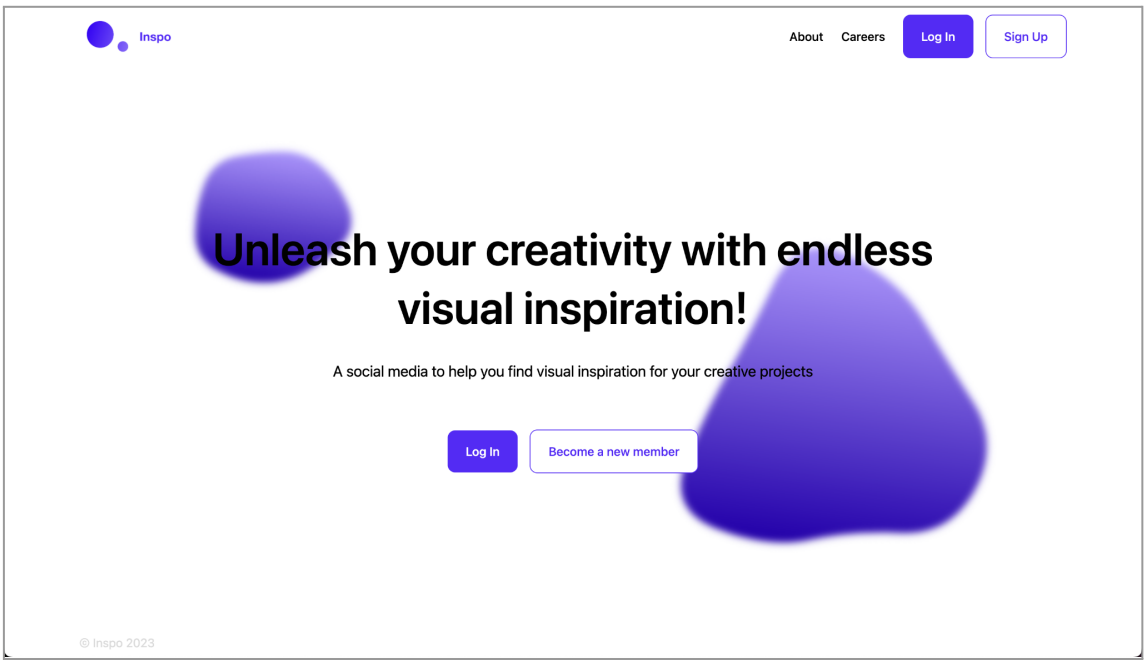

Рис. 3.1 Домашня сторінка соцмережі Inspo

Система забезпечує одну з вимог юзабіліті *інтуїтивність,* а саме легкість розуміння різниці між двома CTA (Call To Action), за допомогою різних варіацій кнопок для переходу на сторінку реєстрації або входу в акаунт – одна з кнопок повністю заповнена основним кольором, інша – має обвідку основного кольору, що дозволяє розрізнити їх значення. Ще один критерій зручності використання, *доступність*, забезпечується за допомогою використання контрастних кольорів. Наприклад, кнопки цільової дії відповідають стандартам WCAG 2 Contrast and Color Requirements [12], із значенням індикатора контрастності 6.95 при мінімумі 4.5 (рис. 3.2).

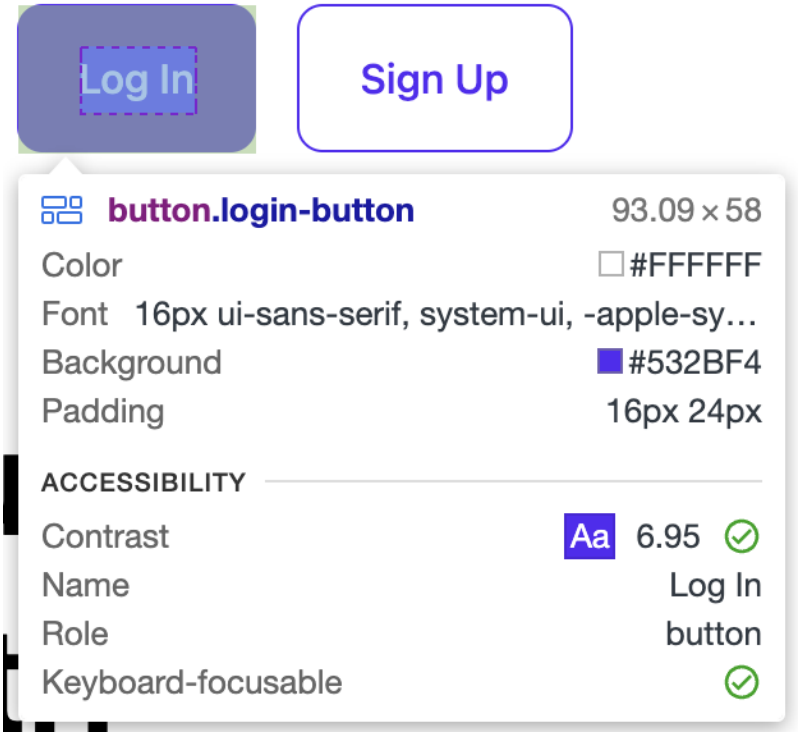

Рис. 3.2 Відповідність кнопки стандартам доступності

Як наслідок, система відповідає вимозі *learnability,* оскільки користувач може легко опанувати нову для нього систему завдяки інтуїтивно зрозумілому та доступному інтерфейсу та *вибрати необхідну дію* для досягнення цілі. Людина натискає “Sign Up” або “Become a new member”, аби зареєструвати новий акаунт, та адресована на сторінку із формою реєстрації (рис. 3.3).

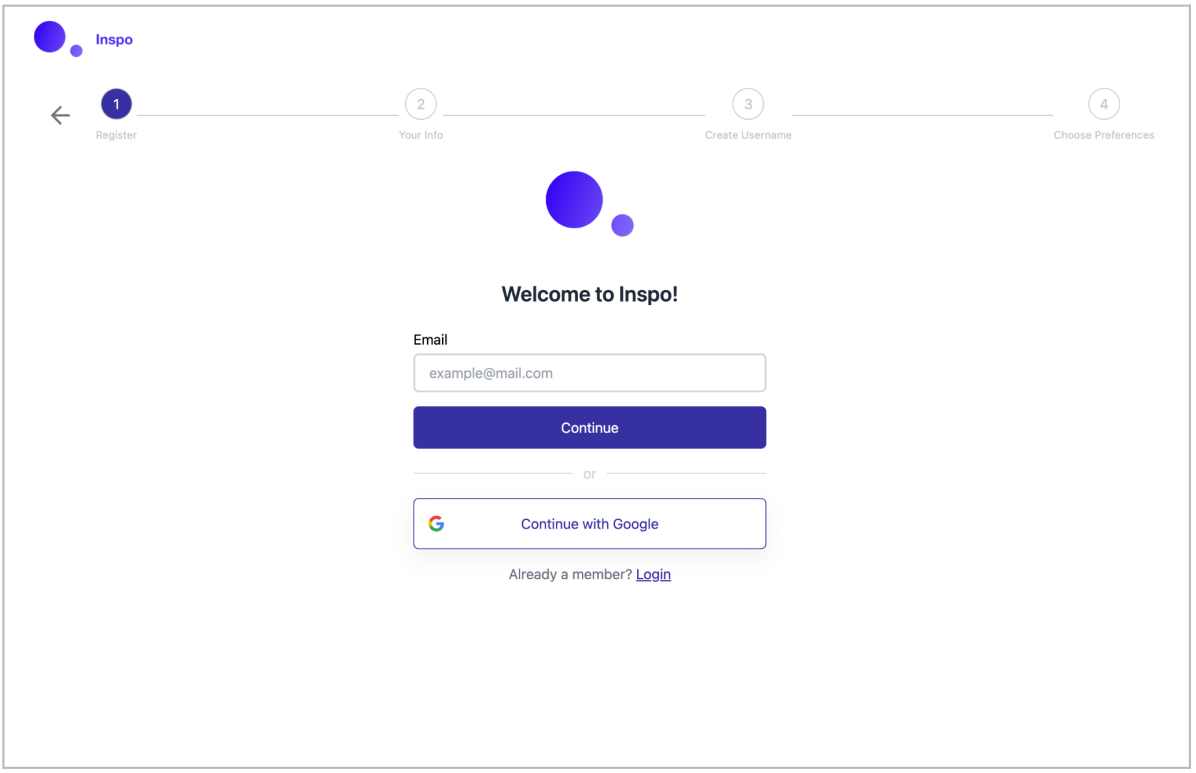

Рис. 3.3 Стартова сторінка реєстрації в Inspo

Даний етап передбачає авторизацію з використанням Google акаунту, або через мануальне введення електронної пошти. До того ж якщо користувач помилково обрав реєстрацію замість входу в акаунт, є опція переадресування на сторінку "Login", при натисканні на посилання, що знаходиться під кнопкою “Continue with Google”. У такий спосіб втілена *обробка помилок і наявність “виходів”,* що є складовою зручності використання. Під час реєстрації система також забезпечує такий критерій юзабіліті, як *чуйність,* за використання індикатору кроків (рис. 3.4). Таким чином інтерфейс дає користувачеві розуміння, яку кількість етапів необхідно пройти, щоб завершити реєстрацію, а отже є орієнтованим на потреби користувача, який може розраховувати власні можливості і час, спираючись на індикатори системи.

Рис. 3.4 Індикатор кроків

Тож користувач має дві опції для реєстрації і відповідно два сценарії.

- Сценарій №1: “мануальна реєстрація”

Користувач власноруч вводить пошту. Система переходить у стан обробки, *валідує ввід даних*. При введені некоректних значень, або за їх відсутності система видає повідомлення про помилку, гарантуючи користувачеві *зворотний зв’язок* і забезпечуючи позитивний досвід використання у такий спосіб (рис. 3.5 та 3.6).

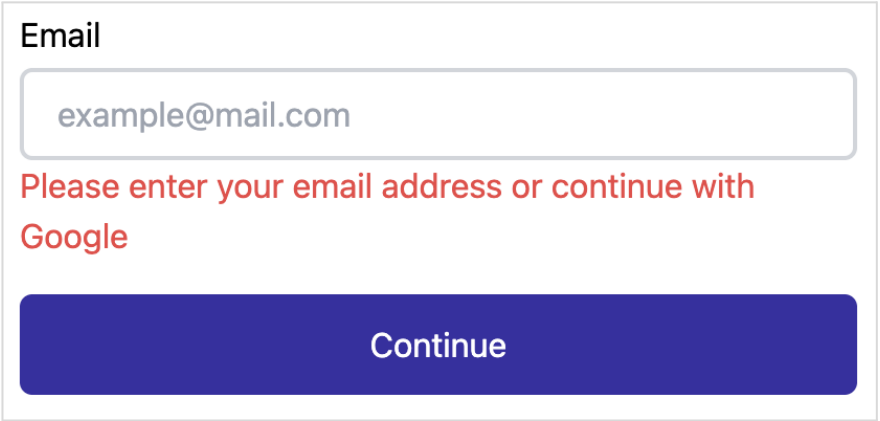

Рис. 3.5 Порожнє поле

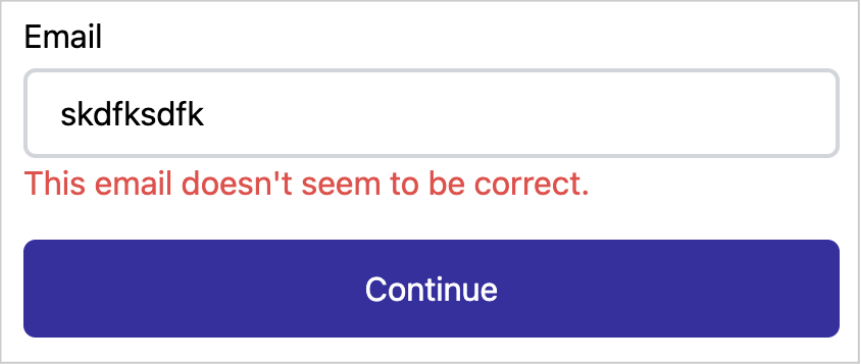

Рис. 3.6 Некоректний ввід даних у поле

При наявності помилок користувач *оцінює стан системи* та як *результат* виправляє їх. Кнопка “Continue” спрямовує користувача на наступний етап реєстрації.

Другий крок “мануальної реєстрації” пропонує користувачеві створити пароль, а також вказати ім’я та прізвище (рис. 3.7). Аби не навантажувати користувача та не змушувати його запам’ятовувати раніше введені дані [3], поле для введення електронної пошти також наявне на даному кроці, користувач має змогу змінити email не повертаючись на попередню сторінку. У такий спосіб система забезпечує її *запам’ятовуваність*.

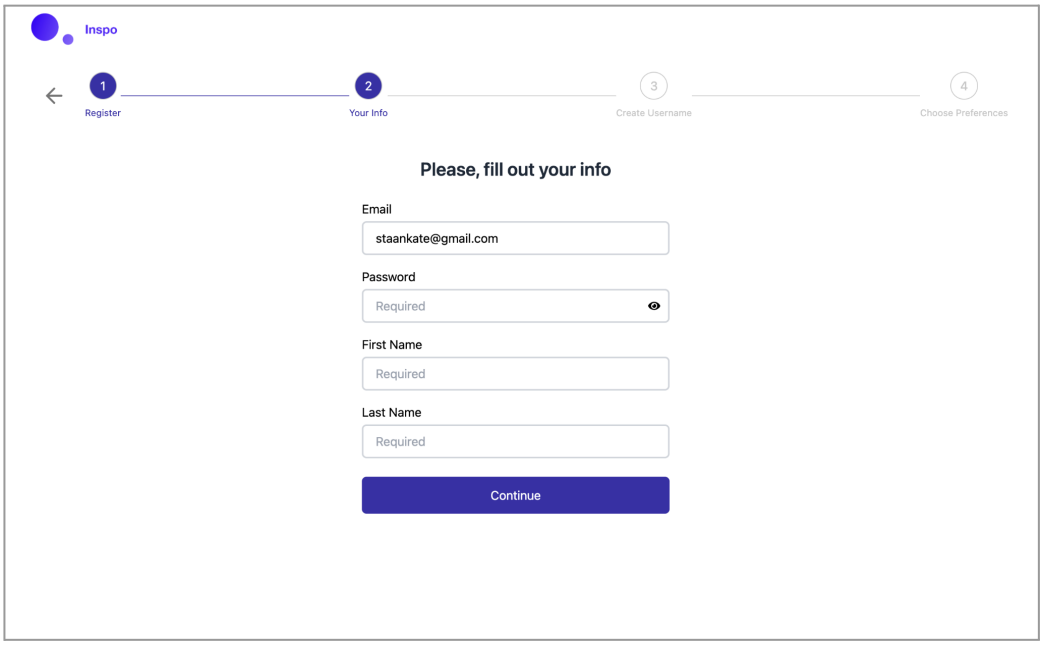

Рис. 3.7 Продовження "мануальної реєстрації"

Введення пароля за налаштуваннями безпеки приховане за замовчуванням, втім, користувач може показати пароль натиснувши іконку ока біля поля введення пароля (рис. 3.8). Таким чином людина може перевірити, чи не допустила вона помилки при введені, а отже *запобігти* її. До того ж система пропонує валідацію пароля задля безпекових цілей та забезпечує зрозумілий *зворотний зв'язок* у вигляді повідомлень про помилки при введенні пароля (рис. 3.9), забезпечуючи *чуйність, інтуїтивність та обробку помилок.*

Рис. 3.8 Приховування пароля

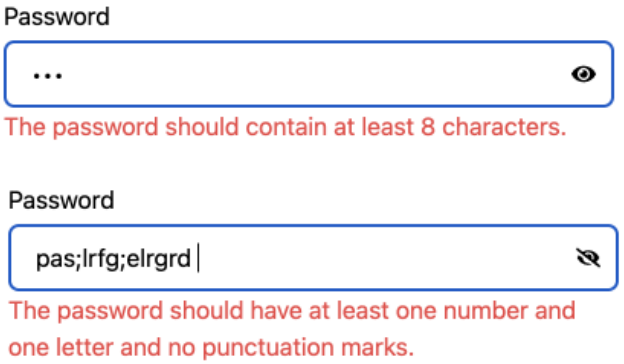

Рис. 3.9 Валідація пароля

Після успішного заповнення полів вводу, користувач натискає кнопку “Continue” та переходить на сторінку створення юзернейму (див. рис. 3.11).

- Сценарій №2: реєстрація з використання Google акаунту

Реєстрація через Google акаунт має такі привілеї як: "попередньо перевірена, пов’язує облікові записи, швидша реєстрація, персоналізований досвід, менша кількість невдалих входів" [14], тому є бажаною опцією користувачів, які не прагнуть вводити та запам’ятовувати логін та пароль для кожної нової реєстрації. Вони більше довіряють знайомій системі, адже Google гарантує впізнаваність, запам’ятовуваність, та безпеку, а тим самим легкість у використанні.

Коли користувач натискає “Continue with Google”, на екрані з’являється вбудований попап від Google (рис. 3.10), що надає користувачеві можливість авторизуватися через бажаний акаунт. Після того, як користувач обрав свій обліковий запис, він адресований на сторінку створення юзернейму (див. рис. 3.11).

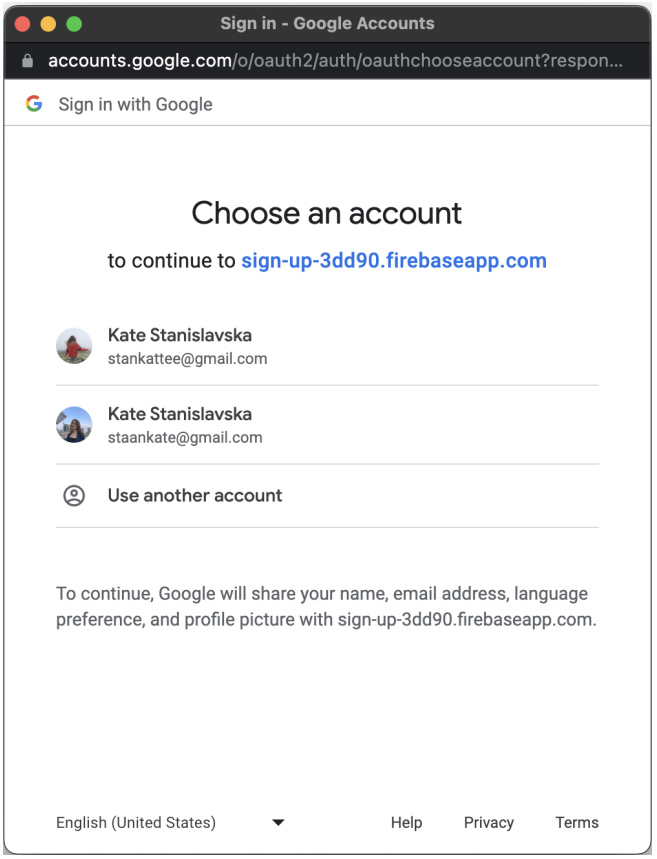

Рис. 3.10 Попап авторизації через Google

На сторінці введення юзернейму (рис. 3.11) користувачеві надаються готові варіанти, що були згенеровані за використання пошти, імені та прізвища користувача. У такий спосіб система орієнтується на потреби користувачів, у яких відсутній час або бажання створювати нікнейм самостійно. Втім, користувач завжди має опцію ввести кастомний юзернейм у поле для введення. При кліці на кнопку "Continue" користувач адресований на останній крок реєстрації (див. рис. 3.12).

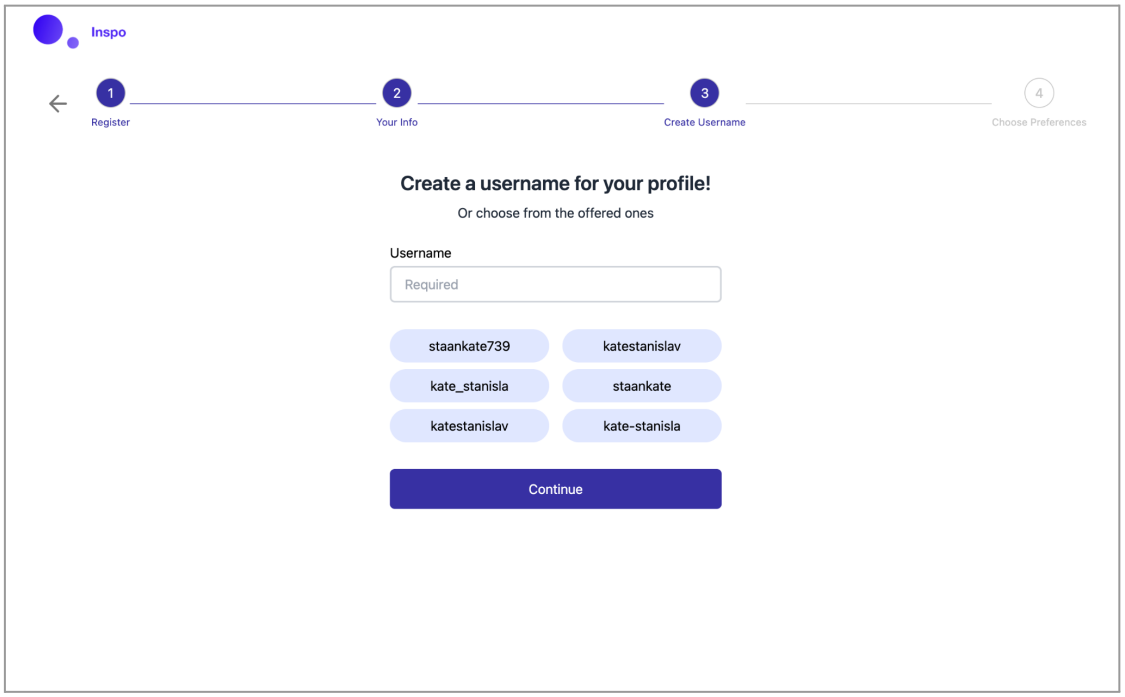

Рис. 3.11 Створення юзернейму

На останньому етапі реєстрації, користувачеві надається можливість обрати набір фотографій із наступних категорій: Подорожі, Спорт, Фотографія, Їжа, Читання, Мода (див. рис. 3.12). Після завершення реєстрації користувач буде бачити контент, який буде спеціально підібраний залежно від його вподобань, що забезпечує більш персоналізований досвід користувача. Однак, якщо користувач не бажає обирати фотографії з переліку, він може пропустити цей крок, натиснувши кнопку "Skip" та завершити реєстрацію.

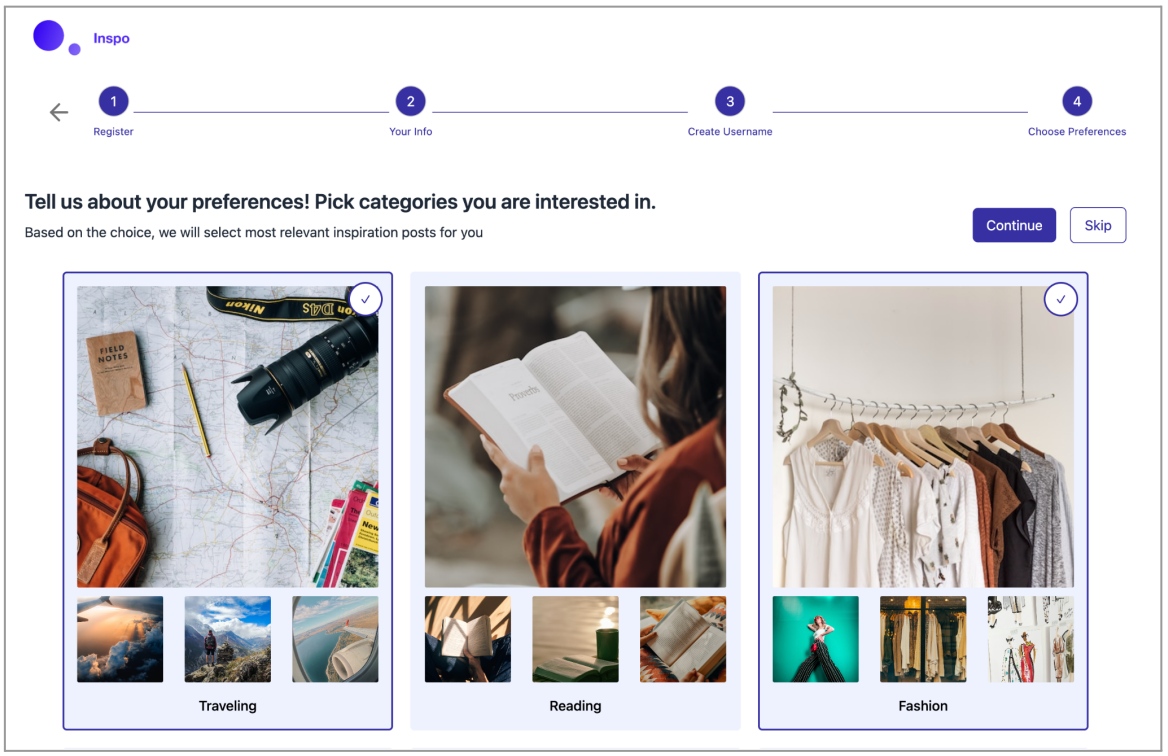

Рис. 3.12 Вибір категорій за вподобанням

Після успішної реєстрації користувач опиняється на домашній сторінці для авторизованих користувачів та може переглядати підібраний для нього контент для референсів (рис. 3.13). У випадку, якщо користувач пропустив крок вибору категорій фотографій, на домашній сторінці для нього підібраний контент з різноманітних категорій.

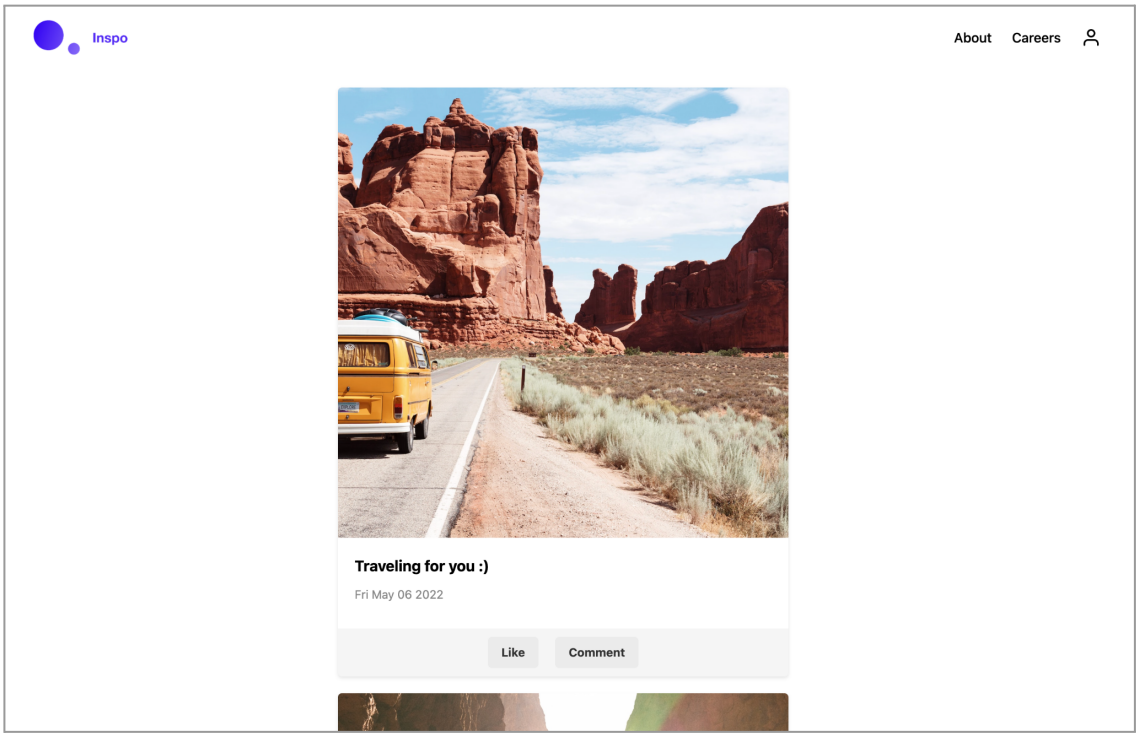

Рис. 3.13 Стрічка постів (домашня сторінка)

Аналізуючи обидва сценарії з точки зору інтерактивності та юзабіліті, можна зробити наступні висновки:

- Інтерактивність: Обидва сценарії надають користувачеві різні варіанти для реєстрації, дозволяючи вибрати найзручніший спосіб. Сценарій №1 (мануальна реєстрація) надає користувачеві більшу гнучкість і контроль над процесом реєстрації, оскільки він може вручну вводити дані й змінювати їх. Сценарій №2 (реєстрація з використанням Google акаунту) пропонує швидшу реєстрацію та забезпечує більшу безпеку, використовуючи попередньо перевірений обліковий запис Google.
- Зручність користування: Обидва сценарії забезпечують юзабіліті шляхом інтуїтивного та доступного інтерфейсу. Система використовує контрастні кольори та індикатори кроків, що полегшують розуміння користувачем процесу реєстрації. Обидва сценарії також надають

зворотний зв'язок у випадку помилок та дозволяють користувачеві їх виправляти.

3.2 Взаємодія другої групи користувачів та додатку для доставки їжі Eatery

Аби досягти своєї *мети* – замовити страву з доставкою – користувачі другої групи мають *намір* розпочати взаємодіяти із додатком. Серед наявних варіантів для початку роботи є реєстрація через електронну пошту (Continue with Email), реєстрація через Google акаунт (Continue with Google), пропуск реєстрації (Skip for now) та вхід в акаунт (Login), рис. 3.14. Варто зазначити, що даний прототип є спрощеною версією застосунку для доставки їжі, а отже у реальному житті опцій для авторизацій може бути більше, наприклад через Facebook або Apple.

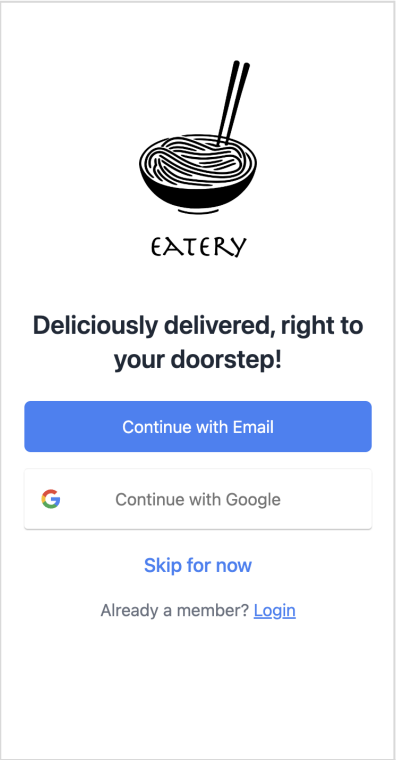

Рис. 3.14 Стартова сторінка

Доступність системи забезпечена належним розташуванням та розміром кнопок для мобільної версії додатку згідно з WCAG 2.0 та іншими W3C/WAI гайдлайнами для мобільної доступності [13]. Система орієнтована на потреби користувачів, тому пропонує декілька опцій реєстрації, аби задовольнити кожен із юзер-сценаріїв.

- Сценарій №1: реєстрація через пошту

При натисканні на “Continue with Email” користувач адресований на сторінку реєстрації яка вимагає заповнення п’яти полів таких як ім’я, прізвище, мобільний телефон, e-mail, та пароль мануально (рис. 3.15). При кожній новій взаємодії користувача і системи, а саме заповнення полів, система переходить на етап обробки, *валідує введені дані,* та *виводить зворотний зв’язок* за необхідності впроваджуючи *обробку помилок* та *чуйність* системи. Також “*вихід*” забезпечується кнопкою “Back” в лівому верхньому кутку екрана, яка переносить користувача на стартову сторінку. При роботі з полями в даному прототипі система також забезпечує *здатність до навчання (learnability)* за допомогою а) конкретизації які саме поля вводу є обов’язковими за додавання плейсхолдера “Required” б) вбудованих підказок браузера для кожного з полів (рис. 3.16). Таким чином користувач має змогу швидко легко опанувати нову для нього систему.

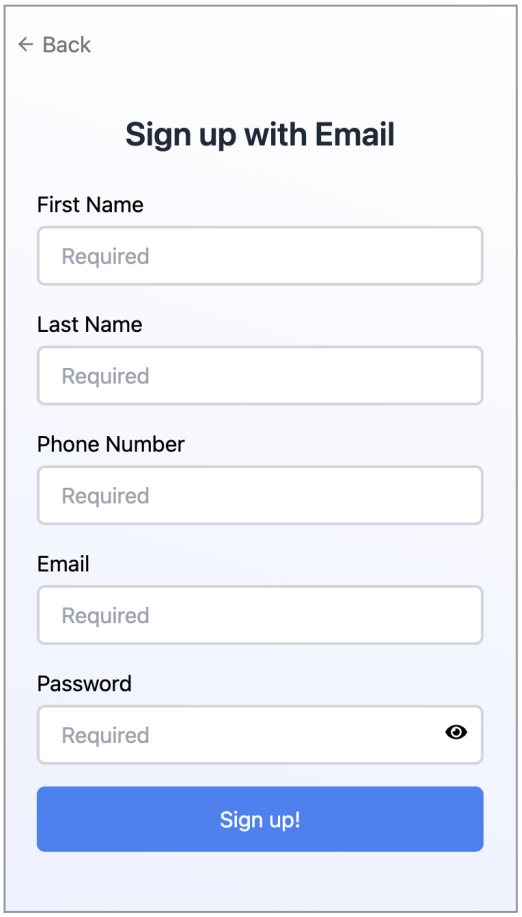

Рис. 3.15 Реєстрація за поштою

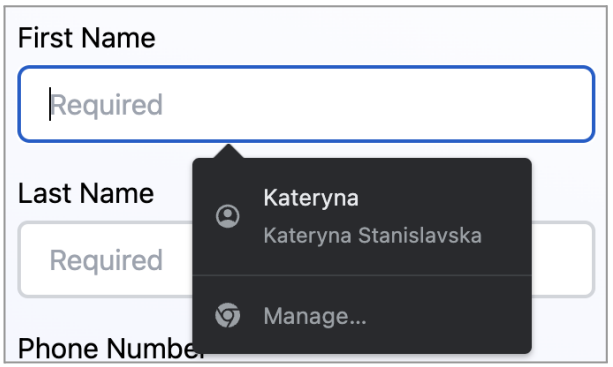

Рис. 3.16 Підказки браузера для полів вводу

Натискаючи кнопку “Sign Up!”, користувач переходить на домашню сторінку, де може обрати страву (див. рис. 3.19).

- Сценарій №2: реєстрація через Google акаунт

Аналогічно до прототипу для соцмережі Inspo, Eatery також відкриває вбудовану авторизацію через Google після натискання на відповідну кнопку (див. рис. 3.10). Далі система адресує користувача на сторінку заповнення персональної інформації, а саме обов’язкового заповнення мобільного телефону (рис. 3.17). Водночас використовуючи принцип *запам’ятовуваності*, система підтягує такі дані як ім’я, прізвище та e-mail з Google авторизації та надає можливість змінити ім’я та прізвище, якщо користувач виявить таку необхідність. У лівому верхньому кутку екрана згідно з вимогами юзабіліті наявна кнопка "Cancel", що скасовує процес реєстрації в разі зміни рішення користувача, забезпечуючи йому *виходи*. При натисканні на неї користувач повертається на стартову сторінку реєстрації (див. рис. 3.14). Натискаючи кнопку “Sign Up!”, користувач переходить на домашню сторінку, де може обрати страву (див. рис. 3.19).

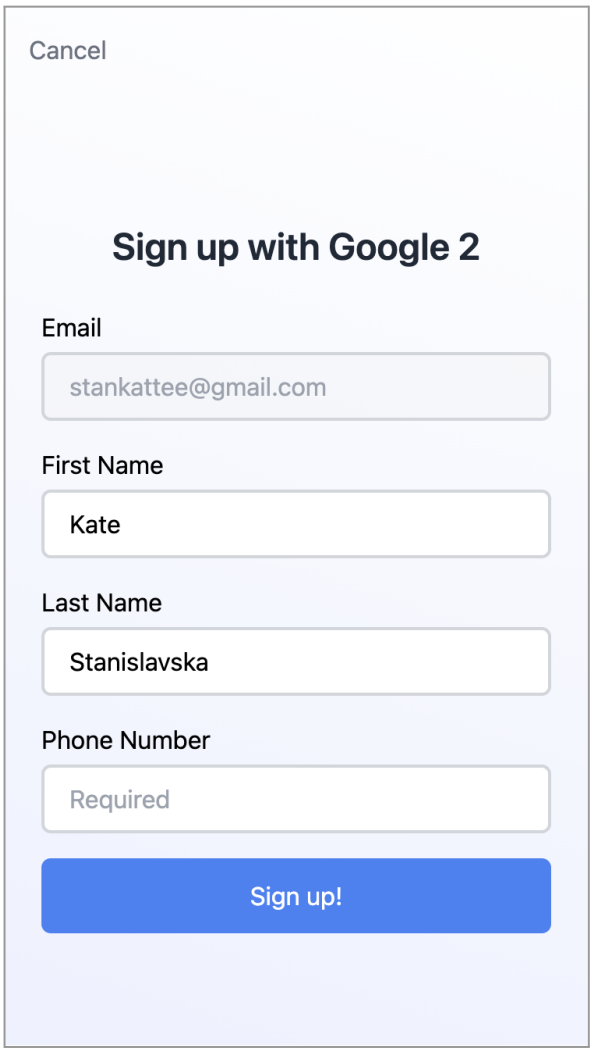

Рис. 3.17 Продовження реєстрації через Google

- Сценарій №3: пропуск первинної реєстрації

Обираючи варіант “Skip for now” (рис. 3.18), користувач має змогу одразу перейти до вибору страви (рис. 3.19). Тобто, система орієнтується на користувачів, у яких є потреба переглянути наявні ресторани та їх пропозиції не витрачаючи часу на первинну реєстрацію, а одразу переходячи до конкретного пошуку.

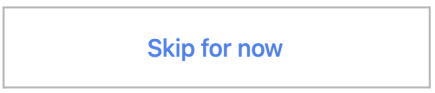

Рис. 3.18 Кнопка пропуску первинної реєстрації

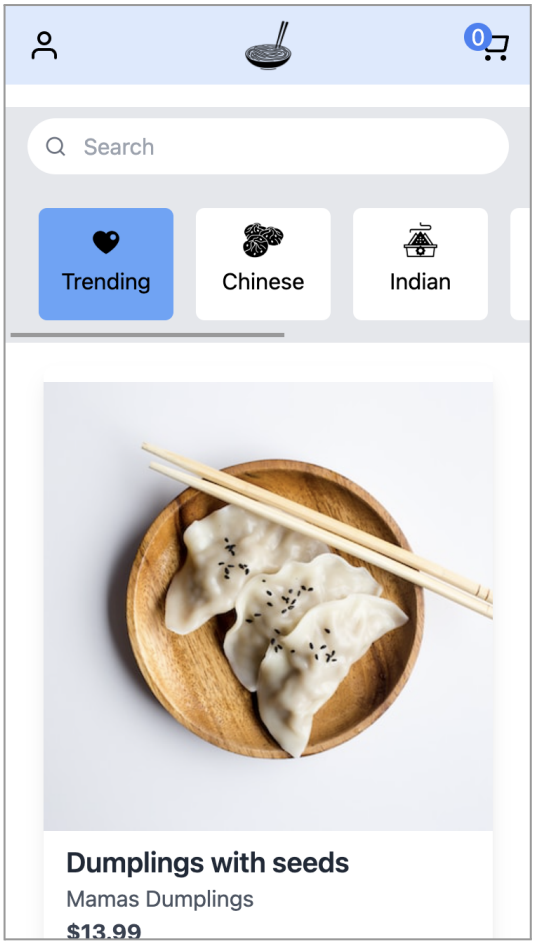

Рис 3.19 Домашня сторінка Eatery

У сценарії №1 та №2, користувач додав би бажані страви в кошик (рис. 3.20) і відразу перейшов би до форми оплати та адресної доставки, натиснувши “Proceed to Checkout”. Натомість у сценарії №3 після кліку на кнопку “Proceed to Checkout” користувачеві доведеться зареєструватись або увійти в акаунт з використанням e-mail або Google, аби оформити замовлення (рис. 3.21).

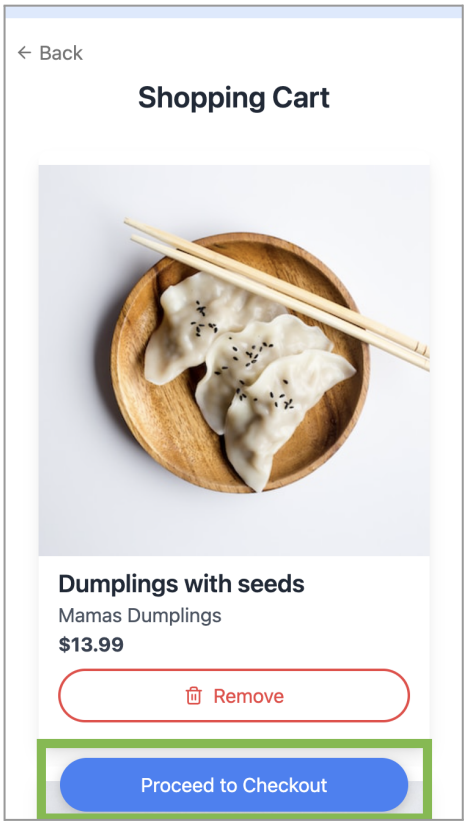

Рис. 3.20 Кошик з кнопкою "Proceed to Checkout"

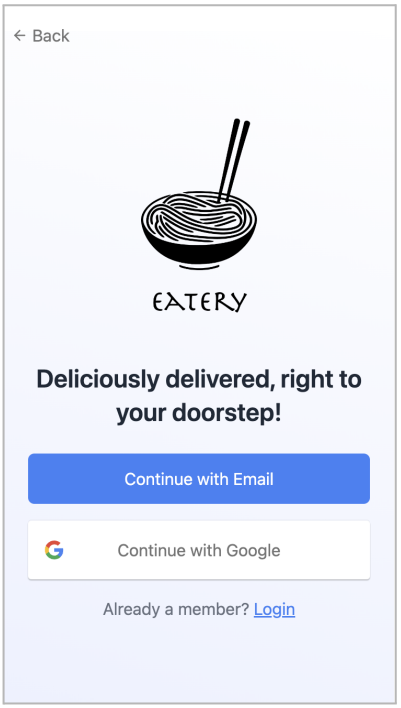

Рис. 3.21 Реєстрація перед оформленням замовлення

Підсумовуючи, у розглянутих сценаріях взаємодії з додатком для замовлення страви з доставкою були представлені три варіанти реєстрації або пропуску реєстрації. Кожен з цих сценаріїв має свої переваги та недоліки.

Сценарій №1 передбачає реєстрацію користувача через електронну пошту. Він забезпечує повний контроль над введеними даними та дозволяє користувачеві вибрати унікальні облікові дані. Однак, цей сценарій вимагає витрати більше часу на заповнення обов'язкових полів.

Сценарій №2 пропонує реєстрацію через Google акаунт, що спрощує процес, оскільки багато користувачів вже мають такий обліковий запис. Він економить час користувача, оскільки деякі особисті дані, такі як ім'я, прізвище та e-mail, автоматично підтягуються з Google авторизації. Однак, для користувачів без Google акаунту цей варіант може бути не зручним або недоступним. Втім в реальних застосунках є й опції авторизації через інші соцмережі, наприклад Facebook, Apple.

Сценарій №3 дозволяє користувачам пропустити первинну реєстрацію та одразу перейти до перегляду доступних страв. Цей варіант зручний для користувачів, які хочуть миттєво ознайомитися з пропозиціями без зобов'язання реєстрації. Однак, при оформленні замовлення їм все одно доведеться пройти реєстрацію або вхід в акаунт, що може створити додаткові перешкоди.

З точки зору зручності використання, сценарій №2, реєстрація через Google акаунт, є найшвидшим і найлегшим для багатьох користувачів, оскільки вони вже мають обліковий запис на даній платформі та можуть довіряти їй більше, аніж застосунку для доставки їжі.

Підсумовуючи розділ, можна зробити висновок, що при розробці різних видів систем, на кшталт реєстрації в соцмережу та реєстрації в додаток для замовлення їжі, необхідно детально продумувати сценарій взаємодії із системою, який буде доречним для цільової групи користувачів. Наприклад, у

застосунку для доставки їжі кроки по типу вибору улюбленої кухні, або детального заповнення профілю, включаючи обов’язковий юзернейм та аватар, під час реєстрації є зайвими, адже клієнти даної системи прагнуть витратити якомога менше часу на речі, які не стосуються замовлення страви. Водночас персоналізований пошук, який базується на виборі категорій фотографій, є доречним під час реєстрації в соцмережу, головним застосуванням якої є пошук референсів.

ВИСНОВКИ

Щодня люди користуються різними видами комп’ютеризованих систем, тому важливо забезпечувати якість даної взаємодії, яка вимірюється її орієнтованістю на користувача.

У роботі розглянуто основні аспекти розробки систем, орієнтованих на користувача та зручних для використання. Для вимірювання зручності застосовують вимоги, такі як здатність до навчання, ефективність використання, запам'ятовуваність, обробка помилок, суб'єктивне задоволення, доступність, інтуїтивність, чуйність та безпека.

Підкреслено індивідуальний підхід до різних типів систем та взаємодії з ними: графічний інтерфейс користувача, командний рядок, обробка природної мови, жести, тактильна відповідь та доповнена/віртуальна реальність, кожен з яких має свої особливості і може включати різні форми реакцій від системи на користувацькі дії.

На прикладі двох прототипів розглянуто різницю підходів до проєктування взаємодії з різними групами користувачів. Визначено специфіку двох різних систем реєстрації з точки зору швидкості та ефективності досягнення мети користувачем. Терміновість здійснення цілі вимагає від системи мінімального залучення користувача у процеси, що не стосуються цієї цілі напряму. Водночас вивід персоналізованих даних від системи вимагає розширеного вводу інформації від користувача.

Дане дослідження в майбутньому варто розширити проведенням кількісного дослідження взаємодії для різних групах користувачів з поточними та іншими варіаціями системи реєстрації задля виявлення критичних точок юзабіліті.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abras C., Maloney-Krichmar D., and Preece J. User-Centered Design, 2004, 445–56
2. Kling R. The Organizational Context of User-Centered Software Designs, MIS Quarterly 1, no. 4, 1977, 41–52
3. Theodor D. Sterling Guidelines for Humanizing Computerized Information Systems: A Report from Stanley House, Communications of the ACM 17, no. 11, 1974, 609–13
4. Bevan N., Kirakowski J., and Maissel J. What Is Usability?, 1991.
5. “ISO 9241-11:2018(En), Ergonomics of Human-System Interaction — Part 11: Usability: Definitions and Concepts,”, https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en.
6. Usability.gov
7. World Leaders in Research-Based User Experience, “Usability 101: Introduction to Usability,” Nielsen Norman Group, https://www.nngroup.com/articles/usability-101-introduction-to-usability/
8. World Leaders in Research-Based User Experience, “10 Usability Heuristics for User Interface Design,” Nielsen Norman Group, https://www.nngroup.com/articles/ten-usability-heuristics/
9. Levy, Steven. "graphical user interface". *Encyclopedia Britannica*, 3 May. 2023, https://www.britannica.com/technology/graphical-user-interface
10. Bass L., Clements P., Kazman R. Software Architecture in Practice, Third Edition [Book], 2013.
11. Donald A. Norman, “Stages and Levels in Human-Machine Interaction,” International Journal of Man-Machine Studies 21, no. 4, 1984
12. “WebAIM: Contrast and Color Accessibility - Understanding WCAG 2 Contrast and Color Requirements,” https://webaim.org/articles/contrast/

13. “Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to Mobile,” February 12, 2015, https://www.w3.org/TR/mobile-accessibility-mapping/
14. Parmar, V., Sanghvi, H. A., Patel, R. H., & Pandya, A. S. A Comprehensive Study on Passwordless Authentication, 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2022, 1266–75
15. Google Authentication with firebase.js https://github.com/fireclint/google-auth-firebase/tree/main/src
16. Blob Animation | Generate Smooth Blob Animation, https://blobanimation.com/