

1. Гра “Камінь-Ножиці-Папір”

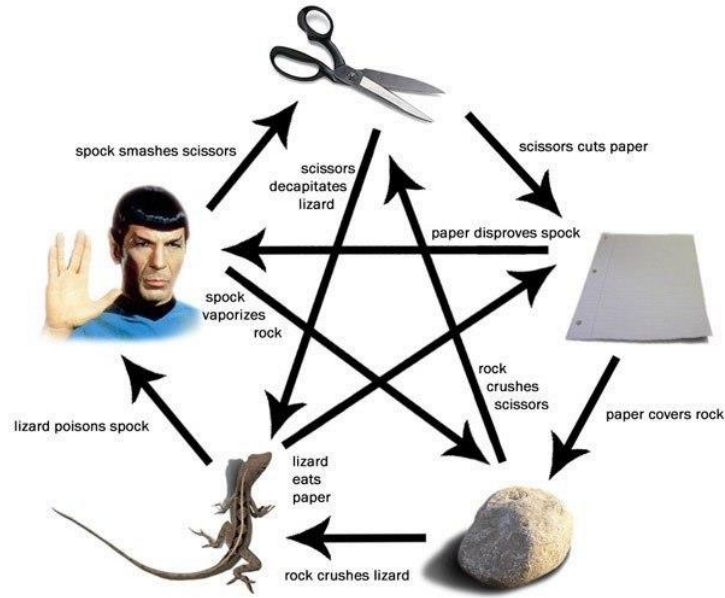
- а. Створити прототип гри камінь/ножиці/папір
 - i. гравець створює нову гру та робить “ставку”(камінь/ножиці/папір)
 - ii. інший гравець робить свою “ставку”(камінь/ножиці/папір)



- iii. відбувається порівняння ставок по схемі
- iv. обох гравців інформують про результат
- б. Інтерфейс користувача повинен містити
 - .перелік доступних ігор
 - i.історію всіх розіграшів
 - ii.вибір випадкової гри
 - iii.випадковий вибір результатів зіграної гри
 - iv.статистику: доступні ігри, зіграні ігри, результат розіграшів, найкращий гравець, найгірший гравець
 - с. Інтеграція з соціальними мережами
 - .Додати вхід через популярні соціальні мережі
 - 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 - 2. Google (<https://developers.google.com/identity/>)

2. Гра “Камінь-Ножиці-Папір-Ящірка-Спок”

- а. Створити прототип гри камінь/ножиці/папір/ящірка/спок
 - i.гравець створює нову гру та робить “ставку”(камінь/ножиці/папір/ящірка/спок)
 - ii.інший гравець робить свою “ставку”(камінь/ножиці/папір/ящірка/спок)
 - iii.Відбувається порівняння ставок по схемі



- iv. Обою гравців інформують про результат
- d. Інтерфейс користувача повинен містити
 - iv. перелік доступних ігор
- v. Історія всіх розіграшів
- vi. Вибір випадкової гри
- vii. Випадковий вибір зіграної гри
- viii. Статистику: доступні ігри, зіграні ігри, результат розіграшів, найкращий гравець, найгірший гравець
 - e. Інтеграція з соціальними мережами
- .Додати вхід через популярні соціальні мережі
 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 2. Google (<https://developers.google.com/identity/>)

3. Байкар-технар

- a. Розробити сервіс створення казок на кшталт гри "Нісенітниця" на аркуші паперу, де кожна фраза загортається і він передається далі
 - i. казка заповнюється по реченню
 - ii. один користувач може додати тільки одне речення в одну казку підряд
 - iii. у вікні додавання речення казкар бачить тільки попереднє речення і кількість речень всього
 - iv. у казці має бути не менше 10 речень
 - v. казка вважається завершеною коли це вказує казкар поставивши відповідну відмітку після додавання не менше 10 речень
- b. Інтерфейс користувача повинен містити
 - .сторінку, де кожен казкар може переглянути завершені казки у які він додав текст. До завершення казки він не знає у створенні якої він приймав участь
 - i. перелік доступних незавершених казок
 - ii. перелік завершених казок, відображення повної казки
 - iii. випадкове відображення завершеної казки
 - iv. випадковий вибір незавершеної казки

v. статистику: незавершені казки, завершені казки, розподіл казок по кількості речень, найбільш активний автор, найменш активний автор

c. Інтеграція з соціальними мережами

.Додати вхід через популярні соціальні мережі

1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
2. Google (<https://developers.google.com/identity/>)

4. Дай списати прогу!

a. Розробити сервіс поширення текстів, аналог [PasteBin](#), [Fedora Paste](#)

i. Автор тексту отримує унікальне посилання прямого доступу яке він може передати іншому користувачу

ii. Автор може переглянути всі розміщені тексти

iii. Автор може задати "час життя" тексту

iv. Після закінчення "часу життя" текст перестає бути доступний всім, але залишається в системі

b. Інтерфейс користувача повинен містити

.перелік доступних текстів

i. випадкове відображення тексту

ii. випадкове відображення тексту який перестане бути доступним менше ніж за певний час (годину, добу, тиждень)

iii. підсвітку синтаксису популярних мов програмування (із автовизначенням мови)

iv. статистику: скільки всього було текстів, скільки доступно зараз, скільки перестане бути доступними через певний час (добу, годину)

c. Інтеграція з соціальними мережами

.Додати вхід через популярні соціальні мережі

1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
2. Google (<https://developers.google.com/identity/>)

5. Не вгадав

a. Створити сервіс генерації випадкової інформації на кшталт [Random.org](#)

i. генерація випадкового числа

ii. генерація послідовності чисел із заданого діапазону

iii. вибір випадкового слова з вказаного тексту

iv. генерація паролів

v. генерація наборів паролів

b. Інтерфейс користувача повинен містити

.історію запитів

i. випадкове відображення запиту з історії

ii. відображення всіх запитів певного користувача

iii. статистику: скільки всього запитів, скільки запитів кожного типу, скільки і яких запитів зробив певний користувач

c. Інтеграція з соціальними мережами

.Додати вхід через популярні соціальні мережі

1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
2. Google (<https://developers.google.com/identity/>)

6. У кого коротше?

- a. Створити сервіс скорочення посилань як-от [goo.gl](#), [bit.ly](#), тощо
- i.кожен користувач може переглянути перелік своїх посилань
- ii.користувач може задати термін дії скороченого посилання
- iii.кожен користувач може переглянути статистику використання посилань
- b. Інтерфейс користувача повинен містити
 - .відображення всіх скорочених посилань
 - i.Відображення посилань які зникнуть через певний період часу(годину, добу, тиждень)
 - ii.Статистику: скільки було всього скорочень, скільки скорочень, скільки посилань перестане бути доступними через певний час(добу, годину)
- c. Інтеграція з соціальними мережами
- .Додати вхід через популярні соціальні мережі
 - 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 - 2. Google (<https://developers.google.com/identity/>)

7. Скільки зустріч не плануй, все одно матимеш



- a. Створити сервіс спільного планування подій на кшталт [Doodle](#).
Інтерфейс створення планування події має надавати можливість:
 - i.вказати опис події, набір дат та період часу протягом якого можна провести зустріч.
 - ii.задати період протягом якого можливе проведення голосування
 - iii.згенерувати посилання для передачі іншим можливим учасникам
 - iv.по завершенню обрати час та дату які набрали найбільшу кількість голосів
- b. Інтерфейс користувача повинен містити
 - .Відображення переліку всіх створених подій
 - i.Випадково обрати подію
 - ii.відобразити найбільш популярну подію
 - iii.відобразити найменш популярну подію
- iv.Статистику: скільки було всього створено подій для планування, скільки всього користувачів проголосувало
- c. Інтеграція з соціальними мережами
- .Додати вхід через популярні соціальні мережі
 - 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 - 2. Google (<https://developers.google.com/identity/>)

8. Базар -- народна платформа

- a. Створити чат на кшталт [e-chat](#) чи [Lycos Chat](#)
- i.зробити можливість створення чату для 2 і більше користувачів
- ii.після створення чату для доступу до нього генерується унікальне посилання
- iii.якщо користувач створює
 - 1. приватний чат то до нього можна підключитись по унікальному посиланню і з введенням кодового слова
 - 2. публічний чат то до нього можна підключитись будь-хто маючи тільки кодове слово
- iv.зберігати всю історію всіх чатів

- b. Інтерфейс користувача повинен містити
 - .відображення переліку всіх чатів з можливістю перегляду конкретного
- i. функцію випадково обрати чат
- ii. функцію відобразити найбільш популярний чат
- iii. функцію відобразити найменш популярний чат
- iv. статистику: скільки було всього створено чатів, скільки приватних та скільки публічних, скільки всього повідомлень
- c. Інтеграція з соціальними мережами
 - .Додати вхід через популярні соціальні мережі
 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 2. Google (<https://developers.google.com/identity/>)

9. “Накрути собі Місс” або “Сам собі Містер”

(Ми їх не обирали, це все боти і накрутка голосів!)

- a. Створити систему голосування
 - i. створити сторінку голосування з вибором між декількома кандидатами
 - ii. створити декілька сторінок вибору Місс/Містер факультету РЕКС, або якогось іншого (кандидатур можна обрати з конкурсів попередніх років)
 - iii. створити інтерфейс для додавання власної категорії голосування, наприклад “Місс конгеніальність ФРЕКС 2017-2018”
 - iv. Користувач може проголосувати тільки один раз в одній категорії!
- b. Інтерфейс користувача повинен містити
 - .відображення всіх категорій
 - i. вибір випадкової категорії
 - ii. відображення кандидата з найбільшою кількістю голосів
 - iii. відображення кандидата з найменшою кількістю голосів
 - iv. статистику: скільки було всього створено категорій, скільки всього проголосувало, у якій категорії найбільше голосів
- c. Інтеграція з соціальними мережами
 - .Додати вхід через популярні соціальні мережі
 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 2. Google (<https://developers.google.com/identity/>)
 3. Instagram (<https://www.instagram.com/developer/>)

10. Петиціарна служба

(Скажемо “Ні!” комерційні забудові навколо КНУ!)

- a. Створити систему збору підписів для петицій
 - i. розробити інтерфейс для створення петиції із вибором категорії
 - ii. після створення петиції користувач отримує посилання для поширення петиції
 - iii. після створення петиції є 2 тижні для збору підписів
 - iv. потрібно зберігати список всіх хто підписав петицію
 - v. для розгляду петиція повинна набрати 200 голосів
- b. Інтерфейс користувача повинен містити
 - .відображення всіх петицій
 - i. відображення всіх петицій певного користувача
 - ii. вибір випадкової категорії

- iii. відображення петицій яка зібрала достатньо підписів для відображення
- iv. відображення петицій які не зібрали достатньо підписів і закінчився термін збору підписів
- v. статистику: скільки було всього створено петицій, скільки всього зібрано підписів
- с. Інтеграція з соціальними мережами
 - .Додати вхід через популярні соціальні мережі
 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 2. Google (<https://developers.google.com/identity/>)

11. Хапай-файлопередай

- а. Розробити сервіс поширення фалів на кшталт [MEGA](#), [RGhost](#)
- i. автор файлу отримує унікальне посилання прямого доступу, яке він може передати іншому користувачу
- ii. автор може переглянути всі свої розміщені файли
- iii. автор може задати “час життя” файлу
- iv. після закінчення “часу життя” файл перестає бути доступний всім, але залишається в системі
- б. Інтерфейс користувача повинен містити
 - .перелік доступних файлів
 - i. випадкове відображення файлу
 - ii. випадкове відображення файлу, який перестане бути доступним менше ніж за певний час(годину, добу, тиждень)
 - iii. статистику: скільки всього було завантажено файлів, скільки доступно зараз, скільки перестане бути доступними через певний час (добу, годину)
 - с. Інтеграція з соціальними мережами
 - .Додати вхід через популярні соціальні мережі
 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 2. Google (<https://developers.google.com/identity/>)

12. Ліквідація безграмотності

- а. Розробити сервіс перевірки правопису типу [Suggest](#), [Grammarly](#) (використовуючи готову бібліотеку, наприклад [aspell](#))
- i. на кожен запит перевірки написання слова або слів автор отримує унікальне посилання прямого доступу, яке він може передати іншому користувачу
- ii. при створенні запиту автор може вибрати мову (українська, англійська)
- iii. автор може переглянути всі свої розміщені запити та результати їх перевірки
- iv. автор може задати “час життя” запиту в системі
- v. після закінчення “часу життя” запит перестає бути доступний всім, але залишається в системі
- б. Інтерфейс користувача повинен містити
 - .перелік поданих запитів
 - i. випадкове відображення запиту
 - ii. випадкове відображення запиту який перестане бути доступним менше ніж за певний час (годину, добу, тиждень)
 - iii. відображення найдовшого запиту, що не містить орфографічних помилок
 - iv. статистику: скільки всього було подано запитів, скільки доступно зараз, скільки перестане бути доступними через певний час (добу, годину)
 - с. Інтеграція з соціальними мережами

.Додати вхід через популярні соціальні мережі

1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
2. Google (<https://developers.google.com/identity/>)

13. ShIT Happens

a. Розробити сервіс поширення коротких історій з голосуванням на кшталт [УкрБаш](#)

i. автор історії отримує унікальне посилання прямого доступу яке він може передати іншому користувачу

ii. автор може переглянути всі свої розміщені історії

iii. автор може задати "час життя" історії

iv. після закінчення "часу життя" історія перестає бути доступною всім, але залишається в системі

b. Інтерфейс користувача повинен містити

.перелік доступних історій, впорядкованих за часом створення або рейтингом

i. випадкове відображення історії

ii. можливість проголосувати за історію (лайкнути)

iii. випадкове відображення історії яка перестане бути доступною менше ніж за певний час (годину, добу, тиждень)

iv. статистику: скільки всього було подано історій, скільки доступно зараз, скільки перестане бути доступними через певний час (добу, годину)

c. Інтеграція з соціальними мережами

.Додати вхід через популярні соціальні мережі

1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
2. Google (<https://developers.google.com/identity/>)

14. Поговори з Америкою

a. Розробити сервіс відображення точного часу відповідно до обраного місця

i. користувач має можливість створити сторінку події, на якій буде відображатись точний час настання події для однієї або декількох локацій (часових поясів)

ii. користувач отримує унікальне посилання прямого доступу до таких сторінок, яке він може передати іншому користувачу

iii. користувач може переглянути всі свої створені сторінки та змінювати часовий пояс під час перегляду, автор може зберегти ці налаштування події

iv. користувач може задати "час життя" сторінки

v. після закінчення "часу життя" сторінка перестає бути доступний всім, але залишається в системі

b. Інтерфейс користувача повинен містити

.перелік доступних сторінок

i. випадкове відображення сторінки

ii. випадкове відображення сторінки яка перестане бути доступною менше ніж за певний час(годину, добу, тиждень)

iii. статистику: скільки всього було створено сторінок, скільки доступно зараз, скільки перестане бути доступними через певний час(добу, годину)

c. Інтеграція з соціальними мережами

.Додати вхід через популярні соціальні мережі

1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
2. Google (<https://developers.google.com/identity/>)

15. Тест на ВІЛ (Високошвидкісний Інтернет Літає?)

- a. Розробити сервіс тестування швидкості передачі файлів на зразок [SpeedTest](#)
 - i. користувач може виконати тест швидкості передачі даних між сервером і власним браузером в обох напрямках.
 - ii. розмір файлу для тестування можна задати
 - iii. по закінченню тесту користувач отримує унікальне посилання прямого доступу до результатів яке він може передати іншому користувачу
 - iv. користувач може переглянути історію своїх тестів
 - v. автор може задати "час життя" результату тесту
 - vi. після закінчення "часу життя" результат перестає бути доступний всім, але залишається в системі
- b. Інтерфейс користувача повинен містити
 - i. перелік доступних серверів для тестування (хоча б 2)
 - ii. запуск тесту через випадковий сервер
 - iii. випадкове відображення результату тестування, який перестане бути доступним менше ніж за певний час (годину, добу, тиждень)
 - iii. статистику: скільки всього було здійснено тестів, скільки результатів доступно зараз, скільки перестане бути доступними через певний час (добу, годину), середня та максимальна швидкість доступу за певний час (добу, годину)
- c. Інтеграція з соціальними мережами
 - . Додати вхід через популярні соціальні мережі
 - 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 - 2. Google (<https://developers.google.com/identity/>)

16. Таємний (безпо-)радник

- a. Розробити сервіс поширення пліток про широко відомих у вузьких колах людей
 - i. користувач в особистому кабінеті може створити чутку про одного із обраних персонажів або запропонувати свого персонажа
 - ii. автор може переглянути всі свої розміщені плітки, змінювати та видаляти їх
 - iii. автор може задати "час життя" своїх чуток
 - iv. після закінчення "часу життя" інформація перестає бути доступна всім, але залишається в системі
- b. Інтерфейс користувача повинен містити
 - i. перелік розпущених чуток
 - i. відображення усіх чуток про певного персонажа (анонімно!)
 - ii. випадкове відображення чуток, яка перестане бути доступною менше ніж за певний час (годину, добу, тиждень)
 - iii. статистику: скільки всього було запущено пліток, скільки доступно зараз, скільки різних персонажів у системі, рейтинг персонажів за кількістю пліток за певний час (добу, годину, увесь час існування сервісу)
- c. Інтеграція з соціальними мережами
 - . Додати вхід через популярні соціальні мережі
 - 1. Facebook (<https://developers.facebook.com/docs/facebook-login>)
 - 2. Google (<https://developers.google.com/identity/>)