|  |  |
| --- | --- |
| Назва програмної системи | Fair Online Judge |
| Призначення системи. | Головна суть цієї системи це порівняння k текстових файлів зі спільною тематикою (рішень задач, певних рефератів, інших документів) на близькість одне з одним, представлення цієї близькості у зрозумілому для користувача вигляді, вибір серед отриманого графіку двох файлів для відображення їх фактичної різниці.  **У базовому вигляді** це буде веб-сервер чи local-app, що буде орієнтований на performance та зручний UI:   * Юзер має змогу завантажити k файлів чи надати посилання на Cloud Drive, з якого можна буде отримати файли; * Далі користувач побачить діаграму схожості файлів один між одним як 2D графік з нодами. На цьому графіку користувач може вибирати ноди, що відповідають певним файлам, та отримувати інформацію щодо їх вмісту – це назва файлу та формат в базовому вигляді, окрім цього це можуть бути метадані (кількість слів/символів, найбільша/найменша/медіана схожості, кількість схожих файлів з заданим найбільшим відхиленням); * На діаграмі схожості юзер може вибрати дві ноди та отримати порівняння контенту двох файлів з виділенням змінених/схожих/різних частин.   **За бажанням та наявністю ресурсів** можливе розширення системи на наступну інфраструктуру: Система неявно надає дві ролі користувачам – publisher, submitter.  Кожен користувач може бути і паблішером і сабмітером, але функціонал у ролей відрізняється:   * Publisher створює пости/руми/завдання, на які submitter, що має ссилку та доступ може засилати рішення. Пости/руми/завдання складаються з назви і опису, publisher може переглядати всі заслані завдання та відхиляти вибрані, також встановлювати порог на найбільш допустиму схожість рішення. Publisher бачить схожість усіх засланих задач, може вибрати k найбільш схожих до заданого, також бачить аккаунти відправників в відповідних нодах. * Submitter має лише посилання, на яке може засилати рішення при наявності відповідних прав. Submitter може засилати не більше одного рішення за заданим посиланням, але може редагувати раніше відправлене.   При наявності доступу submitter може бачити схожість свого рішення з іншими, кількість пройдених тестів у випадку використання коду як рішення. |
| Порівняння з існуючими аналогічними системами.  Ефект від її впровадження | AFAIK, немає аналогів з порівнянням k (k > 2) рішень один між одним.  Ефектом впровадження для розробників є поповнене портфоліо, отримані знання в темах веб-програмування, машинного навчання.  Ефектом впровадження для клієнтів є надання рішення для проблему пошуку схожості серед групи файлів, що буде використовуваним в наступних випадках:  Група [X-Z] – група людей, що вибрана за певними інтересами.   * Ця програма буде корисною для групи X, бо планується викладення коду на відкритий github репозиторій, що надасть змогу розібратися в поняттях nlp, теплових карт, машинного навчання, веб-програмування і т.д. * Ця програма буде корисною для групи Y, бо часто виникає потреба у розумінні унікальності запропонованої роботи у випадку наявності багатьох робіт за спільною тематикою. |
| Загальна характеристика задачі, що вирішується | * Пошук близькості між двома текстовими файлами на даний момент має два основних підходи до рішення - TF-IDF, Word Embedding; * Приблизне або точне відображення точок з заданими попарно функціями відстані, що може бути виконане через теплову карту, деяку поверхню, чи в певній мірі t-sne/pca; * Пошук різниці між двома файлами через жадні diff алгоритми та інші алгоритми схожості; * Побудова архітектури веб-серверу з описаним вище функціоналом. |
| Мотивація вибору задачі. | * Бажання вирішення проблеми справедливого оцінювання робіт шляхом візуалізації подібності рішень; * Особиста зацікавленість у дослідницькій роботі, яка б використала знання та навички, отримані з розробки цього проекту; * Відсутність задовільних рішень задачі автоматизації оцінювання, враховуючи наявність плагіату, та бажання сприяти розв’язку цієї задачі; * Бажання почати кар’єру та розвиток в прогресивному напрямку машинного навчання. |
| Аналіз здійсненності. Необхідні для виконання проекту ресурси. Основний ризик для проекту | Складність полягає у плануванні розробки проекту, тобто рішенні підзадач зазначених у відповідних пуктах, проектуванні архітектури, опануванні фронт-енд частини.  Ризиком є можлива недостатність часу для виконання бажаного обсягу роботи, що призведе до урізання функціоналу, вибрана ітеративна модель, що може сповільнити процес розробки, відсутність розробника, що на достатньому рівні вміє робити Front-End. |
| Модель виробничого циклу. Обгрунтування вибору. Відмінні риси процесу розробки обраної системи | В якості моделі була вибрана ітеративна модель. З вибором такої моделі на кожному кроці буде змога повенутися на попередні для редагування аспектів проекту. |