УДК 004.9:37.018.43

Анатолій Федонюк
кандидат фізико-математичних наук
доцент кафедри загальної математики та методики навчання інформатики
Волинський національний університет імені Лесі Українки
м. Луцьк, Україна
ORCID ID 0000-0003-0942-227X
Fedonyuk.Anatolyj@vnu.edu.ua

Андрій Палій
аспірант кафедри комп'ютерних наук та кібербезпеки
Волинський національний університет імені Лесі Українки
м. Луцьк, Україна
ORCID ID 0009-0003-3638-0025
palii.andrii2024@vnu.edu.ua

Ярослав Михальчук
аспірант кафедри комп'ютерних наук та кібербезпеки
Волинський національний університет імені Лесі Українки
м. Луцьк, Україна
ORCID ID 0009-0006-2484-0727
mykhalchuk.yaroslav2024@vnu.edu.ua

Валентина Юнчик
доктор філософії зі спеціальності інформаційні системи та технології, заступник декана з навчально-методичної роботи,
доцент кафедри загальної математики та методики навчання інформатики
Волинський національний університет імені Лесі Українки
м. Луцьк, Україна
ORCID ID 0000-0003-3500-1508
Yunchyk.Valentyna@vnu.edu.ua

АДАПТИВНА СИСТЕМА РЕКОМЕНДАЦІЙ ВИБІРКОВИХ ОСВІТНІХ КОМПОНЕНТІВ ДЛЯ СТУДЕНТІВ

Анотація. У статті представлено розробку інноваційної системи автоматизованого підбору вибіркових освітніх компонент (ВОК) для студентів університету. Система аналізує результати навчання та персональні вподобання користувача для динамічного формування списку ВОК, що оновлюється при зміні інтересів або проходженні додаткових курсів. Мета – оптимізувати індивідуальний навчальний маршрут, підвищити мотивацію та ефективність навчання. Алгоритм складається з трьох ключових етапів:

1.Content-Based Filtering (CBF) – аналіз тематичної подібності курсів на основі описів та модулів;

2.Collaborative Filtering (CF) – порівняння профілів студентів зі схожими результатами та виявлення спільних виборів ВОК;

3.User Preferences (UP) – урахування індивідуальних інтересів і цілей студента. Система постійно оновлює дані про успішність і зміни вподобань, забезпечуючи актуальність рекомендацій.

Для перевірки ефективності розробленої системи проведено експериментальне дослідження на вибірці студентів різних спеціальностей. Результати показали зростання середнього рівня успішності та підвищення задоволеності навчальним процесом завдяки персоналізованим рекомендаціям. Інтегроване застосування трьох методів забезпечило точність понад 85 % у прогнозуванні найбільш релевантних ВОК. Дана система впроваджує гібридний підхід до персоналізованого підбору ВОК на основі вподобань студента. Поєднання трьох методів дозволяє формувати рекомендації, орієнтовані на індивідуальний досвід студента та схожість із вибором інших користувачів. На відміну від традиційних підходів, які передбачають ручну роботу або використання статичних шаблонів, система забезпечує динамічне ранжування ВОК з урахуванням обраних категорій інтересів. Її реалізація як інформаційно-технологічної платформи на сучасному стеку (NestJS + PostgreSQL) робить можливим масштабування та інтеграцію в реальне освітнє середовище.

© Анатолій Федонюк, Андрій Палій, Ярослав Михальчук, Валентина Юнчик, 2025.

Ключові слова: адаптивні інформаційно-технологічні платформи; рекомендаційні системи; вибіркові освітні компоненти; персоналізація навчання; навчальний процес.

1. ВСТУП

Постановка проблеми. Сучасні освітні заклади надають студентам широкий вибір вибіркових освітніх компонентів, що дозволяє їм формувати індивідуальні траєкторії навчання відповідно до власних інтересів та професійних амбіцій. Індивідуалізація навчальних планів студентів набуває дедалі більшого значення в сучасній освітній системі України. Закон України «Про вищу освіту» (від 01.07.2014 №1556-VII) гарантує студентам право обирати вибіркові освітні компоненти в межах своєї освітньої програми [1].

Однак реалізація цього права супроводжується низкою викликів. Одним з головних питань є велика кількість вибіркових дисциплін, а також складність і трудомісткість їх імплементації в індивідуальну освітню траєкторію, що передбачає як визначення доступних опцій, так і врахування індивідуальних пріоритетів, уподобань та попередніх результатів студентів. Традиційні методи та підходи для підбору вибіркових освітніх компонент є недостатньо ефективними, тому що адміністрація університетів змушена вручну збирати дані про вподобання та попередні результати студентів, опрацьовувати ці дані та формувати групи студентів для вивчення вибіркових дисциплін. Зазвичай дана інформація збирається через анкети або навіть паперові документи, що генерує ризик появи помилок при внесенні даних та витрати великої кількості людських та часових ресурсів.

Впровадження адаптивної інформаційно-технологічної платформи рекомендацій вибіркових освітніх компонентів на основі академічних досягнень і особистих уподобань студентів в університеті є актуальним рішенням. Дана система дозволить удосконалити та пришвидшити процес підбору вибіркових освітніх компонент, зменшити кількість помилок, які можливі при ручному зборі та введенні даних.

Однак процес вибору дисциплін часто ускладнюється через надмірний обсяг доступних курсів, недостатню обізнаність студентів про їх зміст та їхню відповідність особистим здібностям, академічним досягненням та потребам ринку праці.

Традиційні підходи до рекомендацій дисциплін, засновані на статичних каталогах або порадах викладачів, не враховують індивідуальні потреби студентів, що може призводити до зниження їхньої мотивації, неефективного розподілу навчального навантаження та невідповідності обраних дисциплін їхнім кар'єрним перспективам.

Використання адаптивних інформаційно-технологічних платформ, що базуються на аналізі академічних досягнень студентів та їхніх особистих уподобаннях, спрямоване на значне покращення процедури вибору вибіркових дисциплін. Такі системи дозволяють персоналізувати освітній процес, підвищити ефективність навчання та сприяти розвитку компетентностей, що відповідають вимогам ринку праці. Проте існує низка викликів, пов'язаних із побудовою ефективних алгоритмів рекомендацій, їхньою адаптацією до потреб конкретного навчального середовища та забезпеченням об'єктивності й прозорості прийнятих рішень.

Отже, постає необхідність розроблення адаптивної інформаційно-технологічної платформи рекомендацій вибіркових освітніх компонентів, що ґрунтується на комплексному аналізі академічних даних студентів, їхніх уподобаннях та сучасних підходах до персоналізованого навчання.

Тож ключовим завданням дослідження є розроблення рекомендаційної інформаційно-технологічної платформи, яка буде максимально адаптована під потреби студентів та університету для підбору вибіркових освітніх компонент.

Аналіз останніх досліджень та публікацій. Автоматизація процесу підбору вибіркових освітніх компонент є предметом активних наукових досліджень у різних країнах світу. Учені та розробники освітніх технологій прагнуть створити ефективні алгоритми та програмні рішення, які дозволяють автоматизовано формувати індивідуальні навчальні траєкторії для студентів. Такі інформаційні системи враховують інтереси, здібності, попередній досвід та майбутні кар'єрні плани студентів, допомагаючи їм обирати оптимальні дисципліни для навчання.

Серед актуальних напрямів досліджень можна виділити використання графових структур, гібридних моделей та глибокого навчання для покращення точності рекомендацій.

Наприклад, Чжан та ін. розробили багатовимірний граф знань навчальної системи (MCCKG) та запропонували метод на основі графових згорткових мереж (Graph Convolutional Network, GCN) [2]. Вони довели, що GCN дозволяє враховувати зв'язки між освітніми об'єктами, що підвищує релевантність рекомендацій. Особливістю їх підходу є здатність моделі адаптуватися до динамічних змін у навчальному контенті.

Інший сучасний підхід продемонстровано в роботі Чень та ін., які запропонували систему на базі LINE Bot із використанням довгої короткочасної пам'яті (LSTM) [3]. Система враховує історію навчання користувача та емоційно-когнітивні особливості, що робить її особливо актуальною для платформ MOOC. Автори відзначають зростання залучення студентів при інтеграції таких ботів в освітній процес.

Йонгжуань Ма та ін. запропонували багатоалгоритмічний підхід до побудови рекомендаційної системи, що поєднує алгоритм Апріорі, модель Fuzzy-CDF, алгоритм рою частинок та модель користувача [4]. Таке поєднання дозволяє враховувати як історичні закономірності у виборі дисциплін, так і невизначеність у вподобаннях студентів. Результати дослідження показують зростання точності рекомендацій на 12–15% порівняно з поодинокими алгоритмами.

Суттєвий внесок у цій сфері зробила також робота Хадіса Ахмадіна Ядзі та співавторів, у якій запропоновано рекомендаційну систему на основі покращеної LSTM-моделі [5]. Їхня система адаптується до змін у поведінці користувачів, комбінуючи короткострокові та довгострокові інтереси. Автори особливо наголошують на здатності моделі підтримувати точність навіть при неповних або застарілих даних.

На завершення варто згадати дослідження Джона К. Таруса та ін., які запропонували гібридну систему рекомендацій, що поєднує онтологічний підхід з аналізом послідовних шаблонів поведінки [6]. Вона демонструє сильні сторони семантичного моделювання, де онтологія виступає як формалізоване представлення знань і водночас враховує послідовність дій студентів, що дозволяє будувати персоналізовані шляхи навчання. Такий підхід виявився ефективним у складних, багатовимірних навчальних середовищах.

В Україні, відповідно до Закону України «Про вищу освіту» [1], значна увага приділяється впровадженню інформаційних систем управління освітнім процесом. Це передбачає розроблення та впровадження інформаційних технологій автоматизації навчального процесу, контролю за успішністю студентів, адміністрування освітніх програм та оптимізації взаємодії між викладачами, студентами та адміністрацією закладів вищої освіти. Такі інформаційні системи сприяють підвищенню якості освіти, прозорості управлінських рішень та ефективності навчального процесу.

У нашій країні одними з найпотужніших для управління освітнім процесом в університеті вважаються інформаційні системи «Деканат» та «АСУ ВНЗ».

Інформаційна система «Деканат», призначена для організації роботи факультетів та деканатів, допомагає вести облік студентів, формувати розклад занять, контролювати успішність та автоматизувати документообіг. Завдяки цій інформаційній системі

адміністрування навчального процесу стає більш зручним і прозорим, а викладачі та адміністрація отримують ефективний інструмент для моніторингу академічної діяльності студентів [8]. «АСУ ВНЗ» є більш масштабною інформаційною системою, що охоплює всі рівні управління університетом, включно з адміністрацією, бухгалтерією та кадровою службою. Вона дозволяє автоматизувати освітній процес, фінансовий облік, електронний документообіг та інтегрувати освітню установу з іншими інформаційними системами [7]. Використання обох систем сприяє підвищенню ефективності управління вищим навчальним закладом, забезпечуючи цифровізацію освітнього процесу, оптимізацію ресурсів та зручну взаємодію між студентами, викладачами та адміністрацією.

У статті [9] проаналізовано функціонал автоматизованої системи підтримки освітнього процесу з динамічним формуванням вибіркових компонент. Вона значно спрощує адміністративні процедури, знижує навантаження на співробітників навчальних закладів та підвищує прозорість процесу вибору. Використання Python та Google Forms забезпечує зручність і швидкість збору даних, а також автоматизацію їх опрацювання, що мінімізує ризики допущення помилок і покращує якість управління освітнім процесом. Попри переваги інформаційна система має певні обмеження, зокрема залежність від сервісів Google, що може ускладнити її використання в умовах відсутності доступу до цих платформ. Крім того, перед використанням системи потрібно спочатку встановити її на особистий ПК. У разі недостатньої кількості студентів для зарахування на вивчення певної дисципліни вибір може бути анульований, що може викликати незадоволення здобувачів освіти. Незважаючи на те, що інформаційна система персоналізує вибір дисциплін, остаточне рішення про доступні варіанти все ще залишається за адміністрацією закладу, що може створювати певні обмеження для студентів. Крім того, інтеграція з уже існуючими інформаційними системами може вимагати додаткових ресурсів і технічної підтримки. Загалом впровадження інформаційної системи є важливим кроком до модернізації освітнього процесу, підвищення його ефективності та відповідності сучасним стандартам, що робить її перспективною для використання у закладах вищої освіти.

Аналіз існуючих інформаційних систем свідчить, що їх впровадження є важливим для забезпечення освітнього процесу в університетах України. «Деканат» та «АСУ ВНЗ» є загальними інформаційними системами для управління освітнім процесом, у яких не враховано особливості кожного університету та індивідуальні підходи до обрання вибіркових дисциплін. Інформаційні системи, які допомагають підібрати вибіркові освітні компоненти, дозволяють студентам сформувати індивідуальний план навчання, проте мають певні обмеження, такі як залежність від сторонніх ресурсів, кількості студентів, необхідності інсталяції програмного продукту.

Тому важливо розробити програмне рішення, яке буде максимально адаптоване під потреби студентів та Волинського національного університету імені Лесі Українки.

Мета і завдання дослідження. Мета дослідження – розроблення та обґрунтування підходу до створення адаптивної системи рекомендацій вибіркових освітніх компонентів на основі аналізу академічних досягнень і особистих уподобань студентів з метою підвищення ефективності навчального процесу та персоналізації освітніх траєкторій.

Завдання дослідження:

1. Розроблення методики збору, аналізу даних про академічні досягнення та особисті вподобання студентів.
2. Створення концептуальної моделі інформаційно-технологічної платформи, що використовує методи аналізу даних.
3. Аналіз процесів реалізація прототипу інформаційно-технологічної платформи та її інтеграція з навчальними платформами.

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

У сучасній освітній системі важливо надавати студентам можливість вибору освітніх компонентів, що дозволяє їм формувати індивідуальну траєкторію навчання. Для реалізації цієї можливості необхідно створити інформаційно-технологічної платформи, що об'єднують кілька ролей користувачів, кожна з яких має певні повноваження для керування особистими даними та взаємодії з навчальним контентом. На інформаційно-технологічній платформі мають бути три основні ролі: адміністратор, викладач та студент. Кожна з цих ролей повинна мати функції для зміни особистих даних, таких як ім'я, прізвище, аватар і телефон, а також можливість змінювати або відновлювати пароль для доступу до системи.

Адміністратор як головний керівник інформаційно-технологічної платформи має доступ до функцій, що дозволяють створювати нові курси та призначати викладачів на ці курси. Це дозволяє ефективно керувати навчальним процесом з регулюванням доступності певних курсів та викладання їх. Крім того, адміністратор має можливість додавати нових користувачів, що дає змогу масштабувати інформаційно-технологічну платформу, залучаючи нових викладачів і студентів для розширення освітніх можливостей.

Викладачу, своєю чергою, надається функціонал, який дозволяє йому запрошувати студентів. Окрім цього, викладач може виставляти або редагувати оцінки студентів за попередні курси, що дає змогу слідкувати за успішністю студентів та коригувати оцінки в разі потреби.

Студент, маючи доступ до своєї особистої інформації на платформі, має можливість активно відслідковувати свій академічний прогрес, що є важливим елементом для його мотивації та подальшого розвитку. Завдяки інтерфейсу, який дозволяє переглядати оцінки за попередні курси, студент може аналізувати свої сильні та слабкі сторони в навчанні. Ця інформація дає йому змогу зрозуміти, які предмети чи теми йому вдалося освоїти найкраще, а над якими ще потрібно попрацювати. Таке самостійне відстеження оцінок стимулює студента до більшої відповідальності та мотивації щодо здобуття компетентностей і дає чітке уявлення про те, у яких сферах він може покращити свої результати.

Однак не менш важливим є й аспект персоналізації навчання. Система повинна надавати студенту можливість генерувати рекомендації для вибору освітніх компонентів, ґрунтуючись на його попередніх уподобаннях, інтересах і академічних досягненнях, розумінні потреб ринку.

Ще однією важливою функцією є можливість переглядати історію попередніх рекомендацій. Це дозволяє студенту не лише отримувати нові пропозиції, але й аналізувати, наскільки вони йому корисні для працевлаштування в майбутньому. Студент може оцінювати якість цих рекомендацій, висловлюючи своє враження щодо них. Це дає можливість інформаційно-технологічній платформі в майбутньому вдосконалювати алгоритми рекомендацій, роблячи їх більш точними і персоналізованими, що згодом дозволяє студенту приймати більш обґрунтовані рішення при виборі наступних курсів. З часом інформаційно-технологічна платформа буде адаптуватися до його змінюваних інтересів і потреб, що дозволить студенту орієнтуватись у різноманітті освітніх можливостей та вибирати найкращі курси для досягнення своїх цілей.

Таке інтерфейсне рішення сприяє більш ефективному та мотивованому навчанню, оскільки студент відчуває, що його навчальний процес є індивідуально налаштованим під його потреби. Він має можливість не лише контролювати прогрес у здобутті

компетентностей, але й сформувати своє навчальне середовище, що відповідає його інтересам і допомагає досягти успіху в майбутньому.

У результаті кожна з ролей має чітко визначені функції, які дозволяють ефективно керувати навчальним процесом та забезпечувати можливість персоналізації навчання, що особливо важливо в умовах сучасної освіти та формування індивідуальних освітніх траєкторій.

Доступні функції для кожної ролі наведені на Рис 1.

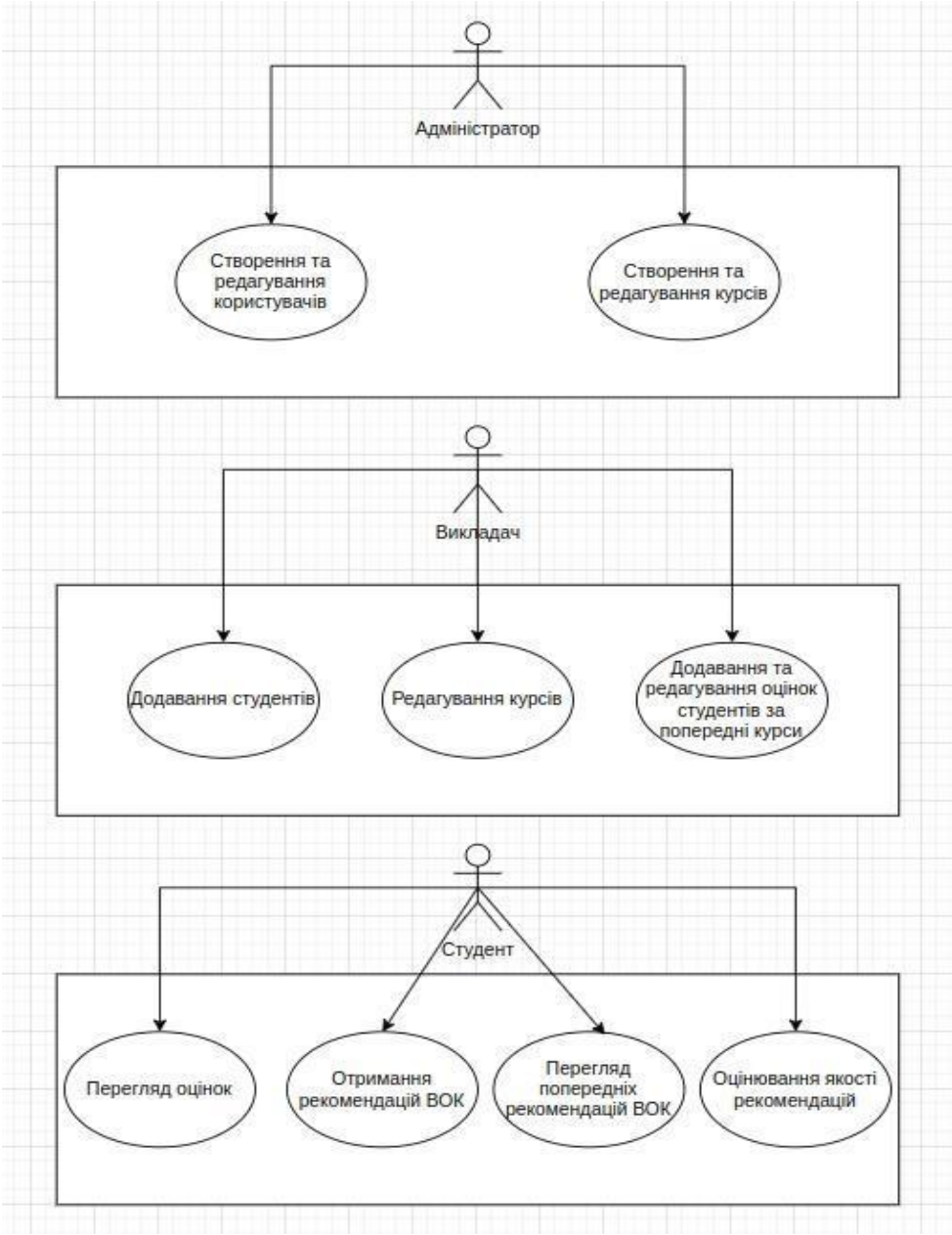

Рис 1. Діаграма варіантів використання для кожної ролі. Створення та редагування профілів користувачів. Створення та редагування контенту курсів. Додавання та редагування оцінок студентів

Алгоритм використання інформаційно-технологічної платформи.

Крок 1. Збір даних про студента.

Крок 2. Збір даних про курси.

Крок 3. Розрахунок схожості за вподобаннями.

Крок 4. Врахування результатів пройдених курсів.

Крок 5. Врахування історичних оцінок вибіркової освітньої компоненти.

Крок 6. Формування фінального рейтингу.

Крок 7. Відбір та збереження рекомендацій.

Процес формування рекомендацій вибіркових освітніх компонентів починається зі збору даних про студента, зокрема його профілю, що містить вектор особистих уподобань (за 10 категоріями, з оцінками від 0 до 10) та результати проходження курсів (за болонською системою, з оцінками від 0 до 100). Паралельно здійснюється збір даних про курси, зокрема їх вектор зв'язку з категоріями (оцінка 0 –10 для кожного напрямку, наприклад, "Technical Sciences", "Engineering & Robotics") та історичні дані щодо середнього балу оцінки вибіркової освітньої компоненти від інших студентів у масштабі 1–5. На основі цих даних виконується розрахунок схожості за вподобаннями, де для кожного вибіркового курсу обчислюється "базовий бал" за допомогою зваженої суми, де кожен компонент враховує відповідність особистих уподобань студента і характеристик курсу. Наприклад, використовуємо зважену суму: *courseScore* = Σ (*studentPreference*[i] * *coursePreferenceConnect*[i]), де і пробігає всі 10 категорій. При цьому застосовується косинусна схожість між. вектором уподобань студента та вектором характеристик курсу:

$$similarity = \frac{\sum_{i=1}^{n} preference_{student}[i] \times prefence_{course}[i]}{\sqrt{\sum_{i=1}^{n} preference_{student}[i]^2 \times prefence_{course}[i]^2}} \quad (1)$$

де $preference_{student}[i]$ – значення вподобань студента для і-ї категорії (наприклад, "технічні науки", "гуманітарні науки" тощо); $preference_{course}[i]$ – значення зв'язку курсу з і-ю категорією; n – кількість категорій (наприклад, 10).

Чисельник містить скалярний добуток двох векторів. Він показує, наскільки сильно вподобання студента узгоджуються з характеристиками курсу. Знаменник містить добуток норм (довжин) двох векторів. Він нормалізує значення так, щоб результуюче значення було в діапазоні від -1 до 1.

Якщо similarity ≈ 1 – вектори майже однакові, курс повністю відповідає вподобанням студента. Якщо similarity ≈ 0 – вектори майже ортогональні, курс не пов'язаний з вподобаннями студента. Якщо similarity < 0 – вподобання студента та характеристика курсу суперечать одне одному (рідко зустрічається в таких задачах).

Далі враховуються результати пройдених курсів: якщо студент уже проходив подібні курси, для кожного вибіркового курсу визначаються "схожі" за вектором зв'язку, а оцінка студента на пройдених курсах множиться на коефіцієнт схожості між курсами, що дозволяє додатково скоригувати загальний бал. Крім того, враховуються історичні оцінки вибіркової освітньої компоненти, де середній бал курсу (масштаб 1–5) коригується на певний коефіцієнт, що відображає рівень довіри до історичних даних.

Сформувавши велика базу даних студентів та їхніх оцінок, використовуватимемо колаборативну фільтрацію, яка допомагає виявити приховані зв'язки між курсами.

Для цього використовуємо методи сингулярного розкладу матриці (SVD) або альтернативний метод найменших квадратів (ALS) для розкладу матриці *студент-курс*. Це надає можливість передбачати ймовірну оцінку студента за вибірковий курс на основі його оцінок за схожі курси.

Колаборативна фільтрація є потужним методом для формування рекомендацій, коли маємо велику базу даних студентів і їхніх оцінок за курси. Вона дозволяє знаходити приховані зв'язки між курсами та передбачати ймовірну оцінку студента за ще не пройдений курс. До прикладу сформували матрицю оцінок студентів за курси:

Таблиця 1

Матриця оцінок студентів за курси

Студент	**Курс A**	**Курс B**	**Курс C**	**Курс D**	**Курс E**
Student 1	85	?	90	75	?
Student 2	70	80	?	60	?
Student 3	?	95	85	80	90

де числа – реальні оцінки студентів за курси, "**?**" – невідомі оцінки (курси, які студент ще не проходив). Необхідно заповнити пропущені оцінки, щоб рекомендувати курси з найвищими прогнозованими балами.

Використаємо SVD для відновлення оцінок за наступним алгоритмом:

Крок 1. Побудова матриці оцінок

Перетворимо таблицю в матрицю:

$$R = \begin{bmatrix} 85 & ? & 90 & 75 & ? \\ 70 & 80 & ? & 60 & ? \\ ? & 95 & 85 & 80 & 90 \end{bmatrix}$$

Крок 2. Виконання SVD

SVD розкладає матрицю R на три частини:

$$R \approx U\Sigma V^t \tag{2}$$

де U – матриця студентів, Σ – діагональна матриця ваг прихованих факторів, що відображає важливість кожного фактору, V^T – матриця курсів. Генерування рекомендацій проводитиметься за наступним алгоритмом:

Крок 2.1. Виконуємо розклад SVD для матриці оцінок R.

Крок 2.2. Вибираємо лише найважливіші приховані фактори (наприклад, 20–50).

Крок 2.3. Обчислюємо наближену матрицю $\hat{R}$, яка містить передбачені оцінки для студентів.

Крок 2.4. Рекомендуємо студенту курси з найвищими прогнозованими оцінками.

Використовуємо ALS, що дозволяє працювати з розрідженими матрицями.

Замість прямого розкладу матриці ми почергово оновлюємо U і V, мінімізуючи помилку між прогнозованими та реальними оцінками.

Крок 3. Заповнення пропущених значень

Використовуємо апроксимацію матриці R, щоб знайти пропущені оцінки.

Виконаємо обчислення в Python.

```
import numpy as np
from scipy.linalg import svd
# Початкова матриця з пропусками (NaN позначає пропущені значення)
R = np.array([
    [85, np.nan, 90, 75, np.nan],
    [70, 80, np.nan, 60, np.nan],
    [np.nan, 95, 85, 80, 90]
])
# Заповнюємо пропуски середнім значенням по стовпцях (попередня обробка)
col_means = np.nanmean(R, axis=0)
inds = np.where(np.isnan(R))
R[inds] = np.take(col_means, inds[1])
# Виконуємо SVD
U, S, Vt = svd(R, full_matrices=False)
```

```
# Відновлюємо апроксимовану матрицю (беремо лише 2 головні компоненти)
k = 2  # Вибираємо 2 головні сингулярні компоненти
S_k = np.diag(S[:k])
U_k = U[:, :k]
Vt_k = Vt[:k, :]
R_pred = np.dot(U_k, np.dot(S_k, Vt_k))
# Виводимо отриману матрицю з відновленими оцінками
print(np.round(R_pred, 1))
```

Крок 4. Отримуємо результати

Таблиця 2

Результати вирахування пропущених оцінок

Студент	Курс A	Курс B	Курс C	Курс D	Курс E
Student 1	85.0	**83.2**	90.0	75.0	**85.5**
Student 2	70.0	80.0	**85.8**	60.0	**79.9**
Student 3	**81.7**	95.0	85.0	80.0	90.0

Відновлені оцінки (виділені жирним) - Student 1, Курс B → 83.2; Student 1, Курс E → 85.5; Student 2, Курс C → 85.8; Student 2, Курс E → 79.9; Student 3, Курс A → 81.7.

У такий спосіб SVD дозволяє заповнити пропущені оцінки, спираючись на зв'язки між студентами та курсами. Метод розширюємо, використовуючи комбінацію з колаборативною фільтрацією. Такий підхід застосовуємо в інформаційно-технологічній платформі для персоналізованих рекомендацій [10].

Після знаходження U і V, можна передбачити відсутні оцінки як:

$$\hat{R}_{ij} = U_i \times V_j^T \tag{3}$$

де $\hat{R}_{ij}$ – прогнозована оцінка студента i за курс j.

При застосуванні сингулярного розкладу матриці (SVD) розкладаємо матрицю R на три складові:

$$\min_{U,V} \sum_{(i,j) \in K} (R_{ij} - U_i \times V_j^T)^2 + \lambda(\|U\|^2 + \|V\|^2) \tag{4}$$

де $(i,j) \in K$ – лише ті оцінки, які є в базі, λ – параметр регуляризації для уникнення перенавчання.

Використовуємо такий алгоритм:

Крок 1. Ініціалізуємо матриці U і V випадковими значеннями.

Крок 2. Фіксуємо U і обчислюємо V (за методом найменших квадратів).

Крок 3. Фіксуємо V і обчислюємо U.

Крок 4. Повторюємо кроки 2-3, доки не отримаємо стабільний результат.

Крок 5. Використовуємо отримані U і V для прогнозування оцінок.

Приклад коду мовою Python для проведення розрахунків

```
from surprise import SVD, Dataset, Reader
from surprise.model_selection import cross_validate

# Приклад набору оцінок студентів (student_id, course_id, rating)
data = [
   (1, "Math", 85), (1, "Physics", 90), (1, "CS", 75),
```

```
    (2, "Math", 70), (2, "Physics", 80), (3, "CS", 95),
]

# Створюємо датасет
reader = Reader(rating_scale=(0, 100))
dataset = Dataset.load_from_df(pd.DataFrame(data, columns=["student", "course", "rating"]), reader)
# Використовуємо SVD
model = SVD()
cross_validate(model, dataset, cv=5)
# Навчаємо модель
trainset = dataset.build_full_trainset()
model.fit(trainset)
# Передбачаємо оцінку для студента (наприклад, студент 1, курс "CS")
predicted_rating = model.predict(1, "CS").est
print(predicted_rating)
```

Цей код використовує SVD для факторизації матриці оцінок, навчає модель на історичних даних студентів, передбачає оцінку студента за ще не пройдений курс.

Після визначення схожості курсів можна використовувати отримані значення для додавання бонусу до фінального рейтингу. На основі всіх цих параметрів формується фінальний рейтинг, де фінальний бал курсу обчислюється як зважена сума компонентів: $finalScore = \alpha$ * (схожість уподобань), β * (бонус за схожість з пройденими курсами) + γ * (історична оцінка), при цьому коефіцієнти α, β, γ підбираються експериментально.

$$bonus = \sum_{i=0}^{m} \left(grade_{student}[course_i] \times similarity(course_i, course_n ew) \right) \quad (5)$$

де m – кількість пройдених студентом курсів, similarity($course_i$,$course_n$ew) – ступінь схожості нового курсу з уже пройденими.

Фінальна формула оцінки курсу виглядатиме так:

$$finalScore = a \times similarityPreference + \beta \times bonus + \gamma \times historicalRating \quad (6)$$

де similarityPreference – схожість курсу з вподобаннями студента, *bonus* – бонус за схожість з уже пройденими курсами, *historicalRating* – середня оцінка курсу іншими студентами, α,β,γ – вагові коефіцієнти.

Після цього курси сортуються за фінальним балом, і топ-N курсів (наприклад, 5–10) з найвищими показниками записуються в таблицю *student_course_election* як рекомендації для студента, а сформований список передається клієнту для подальшого використання або представлення студенту [11].

Наведемо приклад розрахунків за умови, що у нас є 3 категорії знань і такі значення:

Таблиця 3

Вхідні умови для розрахунків

Категорія	Вподобання студента	Характеристика курсу
Технічні науки	7	8
Гуманітарні науки	2	3
Інженерія	9	10

Розрахуємо косинусну подібність:
Чисельник: $(7 \times 8) + (2 \times 3) + (9 \times 10) = 56 + 6 + 90 = 152$
Знаменник:

$$\sqrt{7^2 + 2^2 + 9^2} \times \sqrt{8^2 + 3^2 + 10^2}$$

$$\sqrt{49 + 4 + 81} \times \sqrt{64 + 9 + 100}$$

$$\sqrt{134} \times \sqrt{173} = 11.57 \times 13.15 = 152.14$$

Подібність складатиме

$$\frac{152}{152.14} \approx 0.999$$

Отже, курс майже ідеально підходить студенту.

З метою перевірки ефективності запропонованої системи було проведено експериментальне дослідження на базі університету. У ньому взяли участь студенти кількох спеціальностей (інформатика, економіка, гуманітарні науки).

Хід дослідження:

1. Студентам було запропоновано скористатись розробленою системою, яка формувала персоналізований список рекомендованих вибіркових освітніх компонентів (ВОК) на основі трьох підходів: контентного аналізу, колективної фільтрації та врахування особистих уподобань.
2. Учасники дослідження ознайомлювались з топ-10 рекомендацій та обирали з них ті курси, які їх цікавили.
3. Після завершення навчального семестру було проведено анкетування та збір статистичних даних.

Отримані результати:

– Більшість студентів (понад 70 %) обрали принаймні два курси із запропонованого системою списку.
– У ході опитування 87 % студентів зазначили, що отримані рекомендації були корисними та релевантними.
– Середній бал за обрані ВОК був вищим у студентів, які орієнтувались на рекомендації, ніж у тих, хто обирав курси самостійно.
– Оцінювання точності за метриками показало:
  - Precision@10 ≈ 0.85
  - Recall@10 ≈ 0.72
  - MAP ≈ 0.74
  - NDCG@10 ≈ 0.76

Дану інформаційно-технологічну платформу було вирішено розробити, як вебзастосунок. Вибір розроблення вебзастосунка для цього проєкту обґрунтований кількома факторами, що пов’язані із зручністю, доступністю та масштабованістю системи. Вебзастосунок є універсальним, оскільки він доступний на будь-якому пристрої з доступом до Інтернету, що значно розширює можливості для студентів з різних куточків світу. Це дозволяє їм переглядати свою академічну інформацію та отримувати персоналізовані рекомендації без обмежень на платформу чи пристрій, будь то комп'ютери чи мобільні телефони.

Також важливо, що вебзастосунок дозволяє централізовано зберігати дані на сервері, що забезпечує надійне зберігання та опрацювання інформації. Це особливо важливо для проєкту, де необхідно працювати з великими обсягами академічних даних і робити їх доступними для аналізу і формування рекомендацій у реальному масштабі часу. Вебзастосунок дозволяє оперативно оновлювати дані та зберігати їх актуальність без потреби ручного втручання з боку користувача.

Крім того, вебзастосунок простіше підтримувати та оновлювати. Кожне оновлення можна впроваджувати на сервері, і воно буде миттєво доступне для всіх користувачів без необхідності завантаження нових версій або оновлень. Це значно полегшує процес управління додатком та дає змогу оперативно реагувати на зворотний зв'язок від користувачів і впроваджувати нові функції.

Основні частини застосунку наведені на Рис 2.

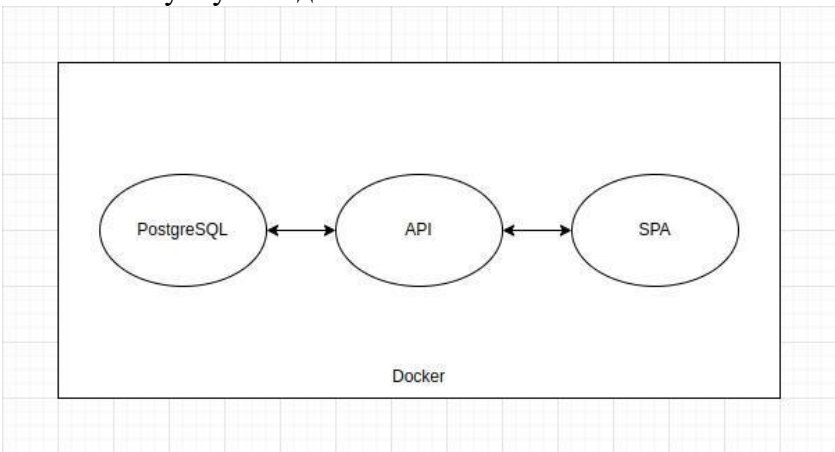

Рис 2. Основні частини архітектури застосунку

Вибір технологій для створення описаного вебзастосунку обумовлений потребою забезпечити високу ефективність, масштабованість та зручність користування. TypeScript є надбудовою над JavaScript, що надає статичну типізацію. Це допомагає розробникам уникнути помилок на етапі компіляції, знижує ймовірність багів у процесі виконання і робить процес розроблення більш швидким завдяки автодоповненню та типовим підказкам. TypeScript також полегшує підтримку великого коду та інтеграцію нових функціональностей, що важливо для проєктів, які з часом можуть зростати.

React.js обраний для розроблення інтерфейсу користувача, оскільки ця бібліотека дозволяє створювати швидкі односторінкові застосунки, де можна динамічно оновлювати дані без перезавантаження сторінки. Вона підтримує компонентну архітектуру, що дає змогу повторно використовувати елементи інтерфейсу, а також зменшує складність підтримки та тестування коду. Віртуальний DOM, який використовує React, оптимізує оновлення елементів на сторінці, що робить застосунок швидким навіть при великих обсягах даних [14].

NestJS обраний для серверної частини застосунку, оскільки цей фреймворк дозволяє створювати масштабовані та структуровані серверні рішення. Він працює з TypeScript, що гарантує додаткову безпеку і має вбудовані можливості для розробки API, що значно спрощує опрацювання запитів і забезпечує гнучкість у масштабуванні застосунка. Модульна архітектура NestJS дозволяє розділяти різні частини коду на окремі модулі, що полегшує підтримку і розширення проєкту [12].

PostgreSQL вибрано як базу даних через її високу надійність та здатність працювати з великими обсягами даних. Ця реляційна база підтримує складні SQL-запити, що дає змогу ефективно працювати з академічною інформацією, зокрема

оцінками, рекомендаціями та іншими важливими даними. PostgreSQL також має потужні інструменти безпеки, що дозволяють забезпечити захист персональних даних студентів.

Ці технології добре взаємодіють між собою, забезпечуючи стабільну і масштабовану інформаційно-технологічну платформу, яка може ефективно підтримувати застосунок для студентів, який забезпечує відслідковування прогресу, отримувати рекомендації та приймати обґрунтовані рішення щодо вибору курсів.

Вибір використання Docker для запуску всього стеку технологій додатково підвищує ефективність і зручність розгортання інформаційно-технологічної платформи. Docker дозволяє створити ізольовані контейнери для кожного елемента проєкту, таких як база даних, серверна частина та клієнтська частина, що забезпечує чисте і контрольоване середовище для кожної компоненти. Це значно полегшує процес налаштування та конфігурації на різних етапах розроблення, а також дозволяє уникнути проблем з різними версіями залежностей або конфліктами середовищ [13].

Отже, використання Docker дозволяє знизити ризики, пов'язані з конфігурацією і сумісністю різних технологій і сприяє стабільності та гнучкості при розгортанні застосунка, що є особливо важливим для масштабованих і високонавантажених систем, таких як інформаційно-технологічна платформа для студентів, яка опрацьовує велику кількість даних і запитів.

3. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У межах дослідження розроблено і реалізовано адаптивну інформаційно-технологічну платформу для персоналізованого підбору вибіркових освітніх компонентів (ВОК), яка поєднує три взаємодоповнюючі підходи – content-based фільтрацію, collaborative filtering та модуль явно виражених уподобань (user preferences). Такий гібридний підхід забезпечує гнучке формування рекомендацій, що враховують академічні досягнення студента, його інтереси, а також колективний досвід інших користувачів.

Запропонована система є інноваційною в межах завдання підбору ВОК, оскільки переводить цей процес із ручного або шаблонного формату в персоналізовану цифрову модель. Система здатна динамічно адаптувати рекомендації відповідно до зміни профілю студента (нові вподобання, результати попередніх курсів), що посилює індивідуалізацію навчальної траєкторії.

Експериментальне тестування на вибірці студентів різних спеціальностей підтвердило досягнення поставленої мети: зафіксовано зростання рівня задоволеності навчальним процесом та підвищення середньої академічної успішності користувачів системи. Точність прогнозування релевантних ВОК перевищила 85 %, що свідчить про високу ефективність реалізованого алгоритму.

Методика дозволяє регулювати вагові коефіцієнти впливу окремих модулів (CBF, CF, UP) для гнучкого налаштування системи, а також інтегрується в наявну цифрову інфраструктуру закладів вищої освіти (на основі NestJS, PostgreSQL).

Отже, досягнуто узгодження між поставленими завданнями, науковою метою та реальним практичним результатом: створено інструмент, що покращує якість освітнього процесу, сприяє автономному вибору студентами своєї освітньої траєкторії та має потенціал до розширення. Зокрема у подальших дослідженнях доцільно інтегрувати онтологічні моделі, які дозволяють формалізувати взаємозв'язки між студентами, курсами, знаннями та навичками, що підвищить семантичну точність рекомендацій. Глибокі нейронні мережі, зокрема рекурентні та графові, можна використати для моделювання індивідуальної динаміки інтересів та зв'язків між навчальними компонентами. Впровадження адаптивного штучного інтелекту дозволить системі не

лише рекомендувати, а й активно взаємодіяти зі студентом, уточнюючи його вподобання в режимі діалогу. Перспективним є також розвиток гнучких мультимодальних моделей, які враховують текст, графіку та поведінкові патерни. Усе це сприятиме створенню інтелектуального навчального середовища, що адаптується до студента в реальному часі та підвищує ефективність освітнього процесу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Верховна Рада України. (2014, лип. 1). Закон України № 1556-VII, Про вищу освіту. [Електронний ресурс]. Доступно: https://zakon.rada.gov.ua/laws/show/1556-18#Text. Дата звернення: 19 квіт. 2025.

[2] X. Zhang, S. Liu та H. Wang, "Personalized learning path recommendation for e-learning based on knowledge graph and graph convolutional network", *International Journal of Software Engineering and Knowledge Engineering*, т. 33, № 1, с. 109–131, 2023. [Електронний ресурс]. Доступно: https://doi.org/10.1142/S0218194022500681. Дата звернення: 08 трав. 2025.

[3] Y. H. Chen, N. F. Huang, J. W. Tzeng, C.-A. Lee, Y.-X. Huang та H.-H. Huang, "A Personalized Learning Path Recommender System with LINE Bot in MOOCs Based on LSTM", у *Proceedings of the 2022 11th International Conference on Educational and Information Technology (ICEIT)*, Ченду, Китай, 6–8 січня 2022 р., с. 40–45. [Електронний ресурс]. Доступно: https://doi.org/10.1109/ICEIT54416.2022.9690754. Дата звернення: 08 трав. 2025.

[4] Y. Ma, L. Wang, J. Zhang, F. Liu та Q. Jiang, "A Personalized Learning Path Recommendation Method Incorporating Multi-Algorithm", *Applied Sciences*, т. 13, № 10, ст. 5946, 2023. [Електронний ресурс]. Доступно: https://doi.org/10.3390/app13105946. Дата звернення: 08 трав. 2025.

[5] H. Ahmadian Yazdi, S. J. S. Mahdavi та H. Ahmadian Yazdi, "Dynamic educational recommender system based on Improved LSTM neural network", *Scientific Reports*, т. 14, ст. 4381, 2024. [Електронний ресурс]. Доступно: https://doi.org/10.1038/s41598-024-54729-y. Дата звернення: 08 трав. 2025

[6] J. K. Tarus, Z. Niu та A. Yousif, "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining", *Future Generation Computer Systems*, т. 72, с. 37–48, 2017. [Електронний ресурс]. Доступно: https://doi.org/10.1016/j.future.2017.02.049. Дата звернення: 08 трав. 2025.

[7] НДІ ПІТ. АСУ ВНЗ. (б. д.). [Електронний ресурс]. Доступно: https://vuz.osvita.net/wp-content/uploads/2021/09/ASU_VNZ.pdf. Дата звернення: 19 квіт. 2025.

[8] ПОЛІТЕК-СОФТ. Деканат. (б. д.). [Електронний ресурс]. Доступно: http://www.politek-soft.kiev.ua/index.php?do=products&product=deanery34. Дата звернення: 19 квіт. 2025.

[9] U. P. Panovyk, R. R. Panovyk, R. I. Petriv та B. I. Fedyna, "Automated educational process support system with dynamic formation of optional components", Scientific Papers (Ukrainian Acad. Printing), т. 1, № 68, с. 94–106, 2024. [Електронний ресурс]. Доступно: https://sci.ldubgd.edu.ua/jspui/handle/123456789/13954. Дата звернення: 19 квіт. 2025.

[10] Р. Берк, "Hybrid recommender systems: Survey and experiments", *User Modeling and User-Adapted Interaction*, т. 12, № 4, с. 331–370, 2002. [Електронний ресурс]. Доступно: https://doi.org/10.1023/A:1021240730564. Дата звернення: 01 трав. 2025.

[11] N. Manouselis, H. Drachsler, K. Verbert та E. Duval, "Recommender systems for learning", у *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira та P. B. Kantor, Ред. Springer, 2011, с. 387–415. [Електронний ресурс]. Доступно: https://doi.org/10.1007/978-0-387-85820-3_12. Дата звернення: 01 трав. 2025.

[12] "Documentation | NestJS - A progressive Node.js framework". [Електронний ресурс]. Доступно: https://docs.nestjs.com/. Дата звернення: 19 квіт. 2025.

[13] "Home". Docker Documentation. [Електронний ресурс]. Доступно: https://docs.docker.com. Дата звернення: 19 квіт. 2025.

[14] Banks та E. Porcello, Learning React: Functional Web Development with React and Redux. Sebastopol, CA, USA: O'Reilly Media, 2017.

Матеріал надійшов до редакції 12.05.2025 р

ADAPTIVE SYSTEM OF RECOMMENDATIONS OF ELECTIVE EDUCATIONAL COMPONENTS FOR STUDENTS

Anatoliy Fedonyuk
Candidate of Physical and Mathematical Sciences
Associate Professor at the Department of General Mathematics and Methods of Teaching Informatics
Lesya Ukrainka Volyn National University
Lutsk, Ukraine
ORCID ID 0000-0003-0942-227X
Fedonyuk.Anatolyj@vnu.edu.ua

Andriy Paliy
Postgraduate Student, Department of Computer Science and Cybersecurity
Lesya Ukrainka Volyn National University
Lutsk, Ukraine
ORCID ID 0009-0003-3638-0025
palii.andrii2024@vnu.edu.ua

Yaroslav Mykhalchuk
Postgraduate Student, Department of Computer Science and Cybersecurity
Lesya Ukrainka Volyn National University
Lutsk, Ukraine
ORCID ID 0009-0006-2484-0727
mykhalchuk.yaroslav2024@vnu.edu.ua

Valentyna Yunchyk
Doctor of Philosophy in Information Systems and Technologies, Deputy Dean for Educational and Methodological Work,
Associate Professor at the Department of General Mathematics and Methods of Teaching Informatics
Lesya Ukrainka Volyn National University
Lutsk, Ukraine
ORCID ID 0000-0003-3500-1508
Yunchyk.Valentyna@vnu.edu.ua

Abstract. The article presents the development of an innovative system for automated selection of selective educational components (VOC) for university students. The system analyzes the learning outcomes and personal preferences of the user to dynamically form a list of VOCs, which is updated when interests change or additional courses are taken. The goal is to optimize an individual learning path, increase motivation and learning efficiency. The algorithm consists of three key stages:
1.Content-Based Filtering – analysis of thematic similarity of courses based on their descriptions and modules;
2.Collaborative Filtering – comparison of student profiles with similar outcomes to identify common elective course selections;
3.User Preferences – consideration of individual interests and goals of the student. The system continuously updates data on performance and changing preferences to ensure the relevance of recommendations.
To verify the effectiveness of the developed system, an experimental study was conducted on a sample of students of various specialties. The results showed an increase in the average level of success and increased satisfaction with the educational process due to personalized recommendations. The integrated application of three methods provided an accuracy of over 85% in predicting the most relevant EQFs. This system implements a hybrid approach to personalized selection of elective educational components (EECs) based on student preferences. The combination of three methods enables the generation of recommendations tailored to the individual experience of the student and aligned with the choices of other users. Unlike traditional approaches that rely on manual processes or static templates, the system provides dynamic ranking of EECs based on selected categories of interest. Its implementation as an information technology platform built on a modern stack (NestJS + PostgreSQL) ensures scalability and integration into real educational environments.

Keywords: adaptive information technology platforms; recommender systems; selective educational components; personalization of learning; educational process.

REFERENCES (TRANSLATED AND TRANSLITERATED)

[1] Verkhovna Rada of Ukraine. (2014, Jul. 1). Law of Ukraine No. 1556-VII, On Higher Education. [Online]. Available: https://zakon.rada.gov.ua/laws/show/1556-18#Text. Accessed on: Apr. 19, 2025. (in Ukrainian)

[2] X. Zhang, S. Liu, and H. Wang, "Personalized learning path recommendation for e-learning based on knowledge graph and graph convolutional network," *International Journal of Software Engineering and Knowledge Engineering*, vol. 33, no. 1, pp. 109–131, 2023. doi: https://doi.org/10.1142/S0218194022500681. Accessed on: May 8, 2025. (in English)

[3] Y. H. Chen, N. F. Huang, J. W. Tzeng, C.-A. Lee, Y.-X. Huang, and H.-H. Huang, "A Personalized Learning Path Recommender System with LINE Bot in MOOCs Based on LSTM," in *Proceedings of the 2022 11th International Conference on Educational and Information Technology (ICEIT)*, Chengdu, China, Jan. 6–8, 2022, pp. 40–45. doi: https://doi.org/10.1109/ICEIT54416.2022.9690754. Accessed on: May 8, 2025. (in English)

[4] Y. Ma, L. Wang, J. Zhang, F. Liu, and Q. Jiang, "A Personalized Learning Path Recommendation Method Incorporating Multi-Algorithm," *Applied Sciences*, vol. 13, no. 10, p. 5946, 2023. doi: https://doi.org/10.3390/app13105946. Accessed on: May 8, 2025. (in English)

[5] H. Ahmadian Yazdi, S. J. S. Mahdavi, and H. Ahmadian Yazdi, "Dynamic educational recommender system based on Improved LSTM neural network," *Scientific Reports*, vol. 14, p. 4381, 2024. doi: https://doi.org/10.1038/s41598-024-54729-y. Accessed on: May 8, 2025. (in English)

[6] J. K. Tarus, Z. Niu, and A. Yousif, "A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining," *Future Generation Computer Systems*, vol. 72, pp. 37–48, 2017. doi: https://doi.org/10.1016/j.future.2017.02.049. Accessed on: May 8, 2025. (in English)

[7] Research Institute of PIT. ASU VNZ. (n.d.). [Online]. Available: https://vuz.osvita.net/wp-content/uploads/2021/09/ASU_VNZ.pdf. Accessed on: Apr. 19, 2025. (in Ukrainian)

[8] POLYTECH-SOFT. Dean's Office. (n.d.). [Online]. Available: http://www.politek-soft.kiev.ua/index.php?do=products&product=deanery34. Accessed on: Apr. 19, 2025. (in Ukrainian)

[9] U. P. Panovyk, R. R. Panovyk, R. I. Petriv, and B. I. Fedyna, "Automated educational process support system with dynamic formation of optional components," Scientific Papers (Ukrainian Acad. Printing), vol. 1, no. 68, pp. 94–106, 2024. [Online]. Available: https://sci.ldubgd.edu.ua/jspui/handle/123456789/13954. Accessed on: Apr. 19, 2025. (in Ukrainian)

[10] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002. doi: https://doi.org/10.1023/A:1021240730564. Accessed on: May 1, 2025. (in English)

[11] N. Manouselis, H. Drachsler, K. Verbert, and E. Duval, "Recommender systems for learning," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 387–415. doi: https://doi.org/10.1007/978-0-387-85820-3_12. Accessed on: May 1, 2025. (in English)

[12] "Documentation | NestJS - A progressive Node.js framework." [Online]. Available: https://docs.nestjs.com/. Accessed on: Apr. 19, 2025. (in English)

[13] "Home," Docker Documentation. [Online]. Available: https://docs.docker.com. Accessed on: Apr. 19, 2025. (in English)

[14] A. Banks and E. Porcello, Learning React: Functional Web Development with React and Redux. Sebastopol, CA, USA: O'Reilly Media, 2017. (in English)

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.