COMPUTER SYSTEMS AND NETWORKS
Vol. 4, No. 1, 2022

DOI: https://doi.org/10.23939/csn2022.01.078
УДК 004.031.6

ПРИНЦИПИ ПОБУДОВИ ВЕБЗАСТОСУНКУ ДЛЯ МОНІТОРИНГУ НАВИЧОК ПРАЦІВНИКІВ

В. В. Кульчицький[1], Н. Я. Павич[2], Я. С. Парамуд[1]

Національний університет "Львівська політехніка",
[1]кафедра електронних обчислювальних машин,
[2]кафедра програмного забезпечення
E-mail: volodymyr.kulchytskyi.mkiks.2021@lpnu/ua; npavych@gmail.com; yaroslav.s.paramud@lpnu.ua

Узагальнюючи означення поняття "навички" із оглянутих літературних джерел визначаємо, що: *навички* – це поєднання знань і вмінь, які людина отримала протягом життя.

Для покращення навичок потрібно вибирати певні інструменти (тренінги, семінари, вебінари, онлайн-курси, курси підвищення кваліфікації, читання спеціальної літератури тощо). Важливе значення у виборі інструментів і досягненні результатів мають особистісні якості та установки молоді.

В рамках цього дослідження було спроєктовано та розроблено принципи побудови вебзастосунку для моніторингу навичок працівників. Вебресурс має зручний користувацький інтерфейс, можливість огляду карти навичок працівника, залишання коментарів до певного навика працівника, дає змогу користувачам створювати і записуватися на існуючі навчальні тренінги, формування сертифікатів, що підтверджують володіння навиком, перегляд статистики.

На відміну від інших аналогів програм для моніторингу навичок, у цій системі передбачено взаємодію трьох типів користувачів: менеджер, ментор та працівник.

До опцій менеджера входять такі: створення шаблону володіння технологією, управління персоналом, управління навиками, управління сертифікатами, управління сповіщеннями;

До опцій ментора входять такі: управління навиками, управління тренінгами, управління сповіщеннями;

До опцій працівника входять такі: управління навиками, управління тренінгами, управління сповіщеннями, управління сертифікатами, перегляд статистики, перегляд рекомендацій.

Ключові слова: принципи побудови вебзастосунку, моніторинг, навик, бази даних, працівник, ментор, менеджер, сертифікат.

Вступ

На багатьох підприємствах та фірмах офіційно та неофіційно відбуваються процеси щодо оцінки навичок працівників. І це допустимо – така людська ідентичність. Звично, що оцінки ці не завжди є правильними. І якщо один із учнів є улюбленцем вчительки в школі – це не дуже приємно для інших учнів. Такий приклад не критичний. Інший приклад, коли один із працівників є фаворитом керівництва підприємства, особливо коли обраний не має навичок для ефективної роботи. Це небезпека для розвитку кар'єри більш талановитих працівників. Кожний із випадків неправильного оцінювання навичок працівників створює небажані стосунки у колективах підприємств та фірм, призводить до погіршення ефективності роботи.

Правильна система перевірки навичок працівників допомагає найкраще усунути негативні проблеми в колективі, оскільки виконується не із зацікавленості конкретного члена органу управління чи працівника, а в інтересах цілої організації. Це найважливіше, що має забезпечувати така система. Оцінка навичок працівників допоможе зберегти власне організацію від найбільш поширених "непорозумінь". Коли все виконано згідно норм і правил, а вони чіткі і прозорі, їх усі знають, всі поінформовані завчасно, тоді багато питань стають простішими і конфлікти не трапляються або мінімізовані. Додатково варто зазначити інший аспект. У багатьох випадках перевірка компетентності стає позитивним зворотнім відгуком для працівників. Це спосіб їх заохотити до якісної роботи та показати, що управління компанії звертає увагу на їхні здібності. Це також спосіб заохотити співробітника до підвищення кваліфікації та дати йому змогу навчатися, позбуватись прогалин у навиках, або розвиватися у схожих напрямках.

Огляд літературних джерел

На етапі планування методів моніторингу навичок працівників керівник проєкту (в більшості випадків – менеджер з управління персоналом) повинен дати відповідь на велику кількість питань. По-перше, які ресурси потрібні. Включаючи розробку та друкування форм для атестації, чи опитування в інтернеті, приєднання консультантів, якщо необхідно, додаткову літературу, навчання асесорів при проведенні асесмент-центру та ін. В цьому випадку менеджер повинен дати чітку оцінку, які ресурси він має, і на що він може розраховувати в подальшому [1]. Потрібно додатково вказати, що у великих організаціях при оцінці працівників часто приймають неефективні рішення, використовують значно складніші і вартісні технології, засоби, ніж це дійсно необхідно для досягнення визначеної цілі [2]. Тому спочатку потрібно спроєктувати систему для моніторингу навичок та описати її основний функціонал, що можна зробити за допомоги мови UML [3]. Доцільно використовувати найбільш поширені: діаграму прецедентів, діаграму послідовності та діаграму класів [4].

Ще одне питання, яке потрібно вирішити заздалегідь – яким чином буде опрацьована інформація щодо навичок, як вона буде перетворюватись, у якому вигляді зберігатися. Якщо про це не подбати заздалегідь, то після оцінки менеджер з управління персоналом буде перевантажений великою кількістю інформації та аналітичних даних [5]. Отже, варто створити уніфіковані формати, принаймні, наскільки це можливо для різних категорій осіб, а великим фірмам потрібно організувати безпаперову – комп'ютерну обробку даних [6]. Для збереження такої інформації використовуються реляційні бази даних [7]. Виконання автоматизованої взаємодії з базою даних можна здійснити функціоналом збережених процедур і тригерів [8]. Також потрібно визначити, в якому вигляді результати будуть показані працівникам: в таблицях, діаграмах, графіках, інших візуальних представленнях [9]. Одним із засобів, які можуть в цьому допомогти, є фреймворк Angular, що використовується для розробки односторінкових вебдодатків та написаний на мові TypeScript [10].

Вебзастосунок можна вважати доцільним і актуальним порівняно з іншими схожими системами, оскільки основною цільовою аудиторією додатку будуть працівники та менеджери з управління персоналом, що допоможе їм автоматизовано опрацьовувати і зберігати результати перевірки кваліфікації працівників. Такі засоби значно підвищать ефективність роботи менеджерів і мінімізують кількість паперової роботи.

Постановка завдання

Завданням цього дослідження є розроблення принципів побудови вебзастосунку для моніторингу навичок працівників, який забезпечить ефективну роботу менеджера з управління персоналом, дасть змогу працівникам переглядати свої навики, сертифікати (отримати можливість роздрукувати сформований сертифікат), забезпечить доступні тренінги для навчання (можливість доєднатися до одного з них) та шаблони спеціальностей (для порівняння вивчених навичок із певним рівнем фаховості).

Однією із важливих вимог до розроблюваної системи є правильна і швидка взаємодія між різними типами користувачів. Оскільки більшість схожих програм не передбачають цього, то потрібно забезпечити швидке опрацювання даних, правильно спроєктувати базу даних і взаємодію між компонентами вебзастосунку, що відповідають за функціонал для різних користувачів.

Ще однією важливою вимогою є раціональне використання системою ресурсів сервера. Так як вебзастосунком можуть користуватись декілька користувачів одночасно і система буде для кожного користувача формувати окрему карту навичок, потрібно оптимізувати обробку даних на сервері, щоб ресурси в основному використовувались саме для графічної роботи.

Результати досліджень

Відомі інструментальні засоби, які використовуються для реалізації програмної системи від етапу аналізу до створення програмного коду. Окремо виділяють так звані CASE-засоби верхнього рівня (upper CASE tools) і CASE-засоби нижнього рівня (lower CASE tools) [3, 9, 10].

Серед основних проблем використання CASE-засобів верхнього рівня виділяють проблеми їх адаптації під конкретні проєкти, оскільки вони жорстко регламентують процес розробки і не дають можливості організувати роботу на рівні окремих елементів проєкту. Альтернативою їм може стати використання CASE-засоби нижнього рівня, але їх використання спричиняє інші проблеми – труднощі в організації взаємодії між командами, що працюють над різними елементами проєкту.

Засобом, що дозволяє об’єднати ці підходи, є уніфікована мова об’єктно-орієнтованого моделювання (Unified Modeling Language – UML). До переваг мови UML можна віднести різноманітні інструментальні засоби, які підтримують життєвий цикл програми та дають змогу налаштувати і відобразити специфіку діяльності розробників різних елементів проєкту. Діаграма прецедентів наведена на рис. 1. На діаграмі Use case зображено таких акторів: менеджер персоналу, ментор, працівник і система моніторингу навичок.

Сценарій використання застосунку менеджером персоналу є такий:

- менеджер з управління персоналом авторизується в системі;
- менеджер може або змінити дані свого профілю, або переглянути опції, які доступні менеджеру персоналу;
- налаштування менеджера на такі опції: створення шаблону володіння технологією, управління персоналом, управління навиками, управління сертифікатами, управління сповіщеннями;
- налаштування опцій загального управління персоналом: зміна ролі користувача та видалення користувача;
- налаштування опції управління навиками для менеджера, що включає перегляд карт навичок усіх працівників, перегляд матриці навичок працівників та додавання навички працівнику;
- налаштування процедури додавання навичок, що включає такі опції як встановлення метрик навичок, встановлення основних розділів, що включаються в навичку та встановлення характеристик навички;
- налаштування режиму управління сертифікатами для менеджера, шо включає перегляд запитів на сертифікат, підтвердження формування сертифікатасертифіката та скасування формування сертифікатасертифіката;
- налаштування опції управління сповіщеннями, що є спільною для всіх типів користувачів та включає створення сповіщень і перегляд сповіщень.

Сценарій використання програми ментором є таким:

- ментор авторизується в системі;
- ментор може або змінити дані свого профілю, або переглянути опції, які йому доступні;
- до опцій ментора входять: управління навиками, управління тренінгами, управління сповіщеннями;
- опція управління навиками для ментора включає перегляд карти навичок своїх учнів, перегляд їхньої матриці навичок та написання коментарів до навичок;
- опція управління тренінгами є основною для ментора та включає створення нових тренінгів, перегляд учасників власних тренінгів та створення рекомендацій для тренінгів, що будуть відображатися на профільному вікні програми для звичайних працівників.

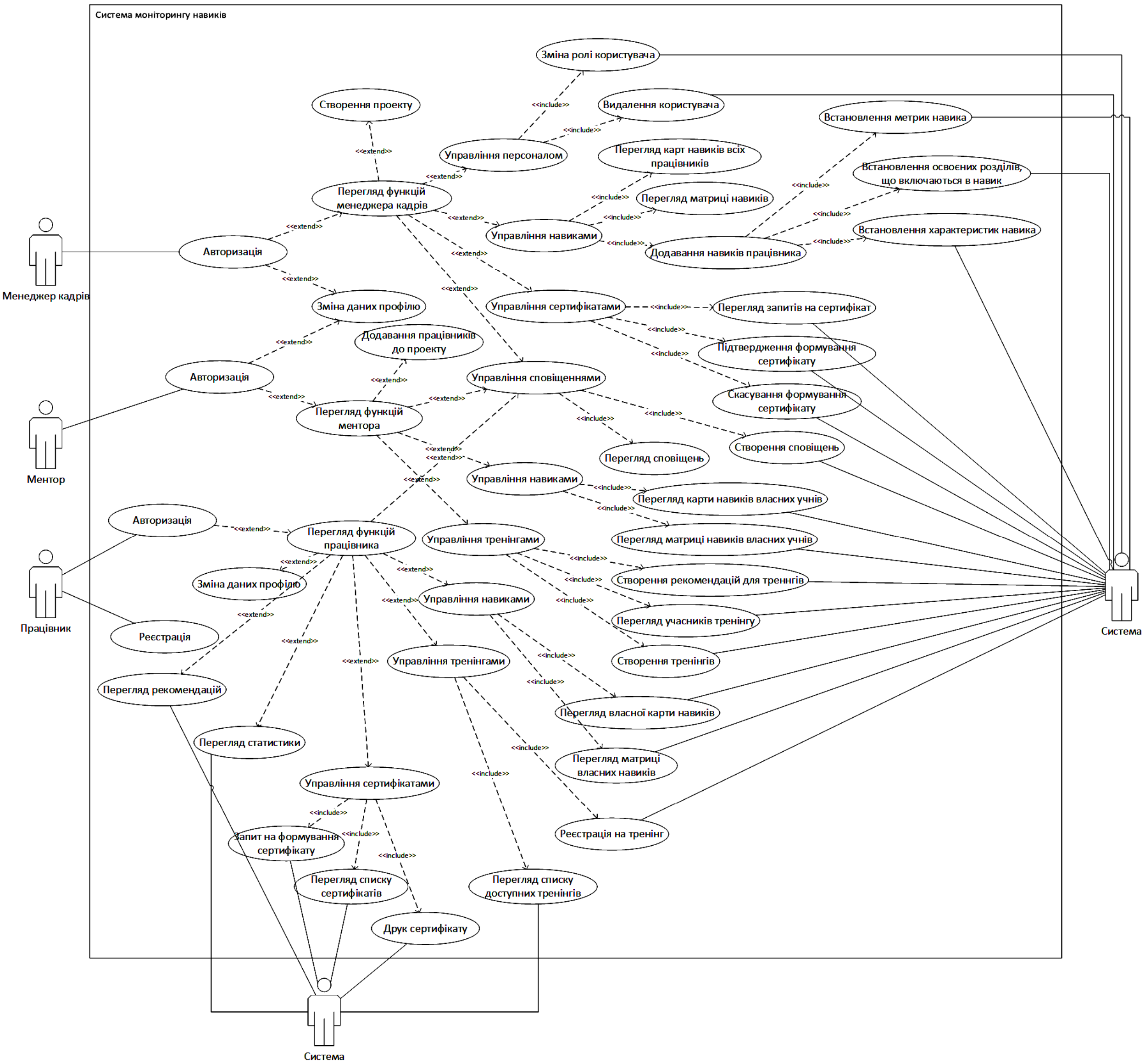

Рис. 1. Діаграма прецедентів

Сценарій використання програми працівником є таким:

- працівник може зареєструватися в системі, після чого менеджер з управління персоналом може змінити його роль у системі;
- працівник може авторизуватись у системі;
- працівник може або змінити дані свого профілю, або переглянути опції, які йому доступні;
- налаштування опцій працівника, до яких входять такі: управління навичками, управління тренінгами, управління сповіщеннями, управління сертифікатами, перегляд статистики, перегляд рекомендацій;
- опція управління навичками для працівника включає перегляд карти власних навичок, перегляд власної матриці навичок;
- опція управління тренінгами для працівника включає перегляд списку доступних тренінгів та реєстрацію на тренінг.
- опція управління сертифікатами для працівника включає перегляд списку власних сертифікатів, створення запиту на формування нового сертифікатасертифіката та формування документу сертифікатасертифіката в форматі .pdf для подальшого друку.

Проєктування бази даних застосунку має такі особливості. Створення бази даних слід починати з проєктування її структури. У результаті проєктування має бути визначена структура бази, тобто склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці визначається складом стовпців, їхньою послідовністю, типом даних кожного стовпця та їхнім розміром, а також ключем таблиці. Процес проєктування можна здійснювати двома підходами. За першого підходу спочатку визначають основні задачі, для розв'язання яких створюється база, та потреби цих задач у даних. За другого підходу визначають предметну область (сферу), здійснюють аналіз її даних і встановлюють типові об'єкти предметної області. Найбільш раціональним підходом проєктування бази даних є поєднання обох підходів.

Процес проєктування бази даних поділяється на етапи, кожний з яких передбачає виконання певних дій. Перший етап – розробка інформаційно-логічної моделі даних предметної області, що базується на описі предметної області, отриманому в результаті її обстеження. На цьому етапі спочатку визначають склад і структуру даних предметної області, які мають міститись у базі даних та забезпечувати виконання запитів, задач і застосувань користувача. Подальше визначення структурних зв'язків між об'єктами дозволяє побудувати інформаційно-логічну модель.

Другий етап – визначення логічної структури бази даних. Для реляційної бази даних цей етап є значною мірою формальним, оскільки інформаційно-логічна модель відображається в структурі реляційної бази даних адекватно.

Наступний етап – конструювання таблиць бази даних, яке здійснюється засобами СУБД, та узгодження їх із замовником. Структура таблиць бази даних задається за допомогою засобів опису (конструювання) таблиць у СУБД із цілковитою відповідністю інформаційним об'єктам.

Модель бази даних передбачає визначення особливостей основних таблиць.

Сутність Employee відповідає за працівників, що будуть мати власні навики. Таблиця містить таку інформацію:

- Id_Employee – унікальний номер (ідентифікатор), призначений кожному працівнику;
- FirstName – імя працівника;
- LastName – прізвище працівника;
- Password – пароль працівника;
- Role – роль працівника;
- DevLevel – посада працівника;
- Experience – досвід працівника;
- ID_Mentor – номер ментора, що відповідає за працівника (посилання на таблицю Employer);
- ID_Company – номер компанії, в якій працює працівник (посилання на таблицю Company).

Сутність UserSkill відповідає за працівників, що будуть мати власні навики. Таблиця містить таку інформацію:

- Id_UserSkill – унікальний номер (ідентифікатор), призначений кожній навичці;
- StartDate – початок вивчення навички;
- EndDate – кінцева дата вивчення навички;
- SkillName – назва навички;
- SkillLevel – рівень володіння навичкою;
- Id_Category – номер категорії, до якої належить навичка (посилання на таблицю Category);
- Id_Employee – номер працівника, до якого прив'язана навичка (посилання на таблицю Employee).

Сутність Training відповідає за тренінги, що можуть створювати ментор та менеджер з управління персоналом. Таблиця містить таку інформацію:

- Id_Training – унікальний номер (ідентифікатор), призначений кожному тренінгу;
- StartDate – дата початку тренінгу;
- EndDate – дата закінчення тренінгу;

- TrainingTitle – назва тренінгу;
- Description – опис тренінгу;
- Id_Category – номер категорії, до якої належить тренінг (посилання на таблицю Category);
- Id_Coach – номер ментора або менеджера, до якого прив'язаний тренінг (посилання на таблицю Employer).

Сутність Certificate відповідає за сертифікати, що будуть створюватися для підтвердження навичок. Таблиця містить таку інформацію:

- Id_Certificate – унікальний номер (ідентифікатор), призначений кожному сертифіката сертифіката;
- DateOfIssue – дата видачі сертифікатасертифіката;
- ExpiryDate – кінцева дата дії сертифіката;
- CertificateTitle – назва сертифіката;
- SkillLevel – рівень володіння навиком у сертифікаті;
- Id_Recipient – номер отримувача сертифіката (посилання на таблицю Employee);
- Id_Publisher – номер ментора або менеджера, що затвердив сертифікат (посилання на таблицю Employer);
- Id_UserSkill – номер навика, для якого створений сертифікат (посилання на таблицю UserSkill).

Сутність SkillTemplate відповідає за шаблон спеціальності для працівників. Таблиця містить таку інформацію:

- Id_SkillTemplate – унікальний номер (ідентифікатор), призначений кожному шаблону спеціальності;
- TemplateTitle – назва шаблону;
- Description – опис шаблону;
- AverageSalary – середня зарплата.

Сутність Requirement відповідає за вимоги, що прикріпляються до шаблону. Таблиця містить таку інформацію:

- Id_Requirement – унікальний номер (ідентифікатор), призначений кожній вимозі;
- ReqTitle – назва вимоги;
- ReqLevel – рівень вимоги;
- Id_SkillTemplate – номер шаблону, в якому використовується ця вимога (посилання на таблицю SkillTemplate).

Інші зв'язки у базі даних такі:

- Зовнішній ключ IdMentor у таблиці Employee посилається на таблицю Employer. Зовнішній ключ IdCompany посилається на таблицю Company.
- Зовнішній ключ IdCompany у таблиці Employer посилається на таблицю Company. Зовнішній ключ IdUserSkill у таблиці Certificate посилається на таблицю UserSkill. Зовнішній ключ IdRecipient посилається на таблицю Employee. Зовнішній ключ IdPublisher посилається на таблицю Employer.
- Зовнішній ключ IdEmployee у таблиці Statistic посилається на таблицю Employee. Зовнішній ключ IdTraining у таблиці TrainingMember посилається на таблицю Training. Зовнішній ключ IdMember посилається на таблицю Employee.
- Зовнішній ключ IdTraining у таблиці Recommendation посилається на таблицю Training. Зовнішній ключ IdEmployee посилається на таблицю Employee. Зовнішній ключ IdCoach у таблиці Training посилається на таблицю Employer. Зовнішній ключ IdCategory у таблиці Training посилається на таблицю Category.
- Зовнішній ключ IdEmployee у таблиці UserSkill посилається на таблицю Employee. Зовнішній ключ IdCategory у таблиці UserSkill посилається на таблицю Category.
- Зовнішній ключ IdUserSkill у таблиці SkillMetric посилається на таблицю UserSkill.

Запропонована модель бази даних приведена на рис. 2.

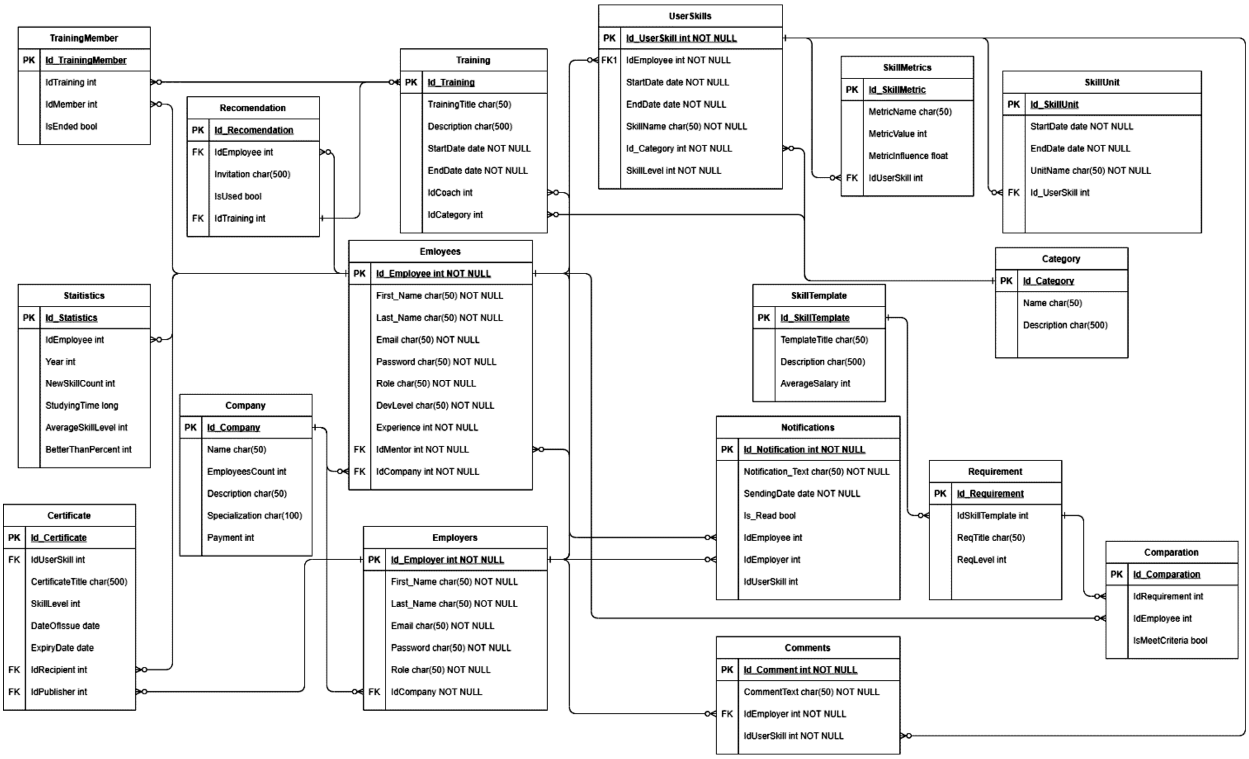

Рис. 2. Модель бази даних

Принципи побудови серверної частини. На першому етапі доцільно створити класи, що представляють таблиці з бази даних. Відповідний список представлений на рис. 3.

Base	Initial commit
Category.cs	Add new additional tables
Certificate.cs	Add training and management
Comment.cs	Bugs commit
Company.cs	Add new additional tables
Comparation.cs	Generate data in db
Employee.cs	Generate data in db
Employer.cs	Add new additional tables
Notification.cs	Bugs commit
Recommendation.cs	Add training and management
Requirement.cs	Generate data in db
SkillMetric.cs	Add training and management
SkillTemplate.cs	Generate data in db
SkillUnit.cs	Add training and management
Statistics.cs	Add new additional tables
Training.cs	Add training and management
TrainingMember.cs	Add new additional tables
UserSkill.cs	Add training and management

Рис. 3. Список класів для представлення таблиць бази даних

Для зберігання списків усіх моделей та обробки даних за допомогою базових функцій роботи з базою даних найкраще використати шаблон репозиторію.

Для кожної моделі доцільно використати свій клас репозиторію, який містить посилання на контекст бази даних та функції додавання, видалення, оновлення, вибірки одного елемента або масиву елементів.

Доступ до всіх таблиць доцільно здійснювати з одного місця, використати клас UnitOfWork, в якому зберігаються всі об'єкти репозиторіїв, та функція збереження змін в базі даних.

Управління більш складною логікою взаємодії системи із базою даних для кожної моделі можна здійснювати оригінальним класом сервісу.

Надання можливості клієнтам взаємодіяти із сервером можна здійснювати оригінальними класами контролерів, що реалізують взаємодію за протоколом HTTP такими основними методами, як GET, POST, PUT, DELETE.

Принципи побудови клієнтської частини. На першому етапі необхідно створити класи для даних, які будуть отримуватися від сервера. Для взаємодії з контролерами сервера було створено класи сервісів, які містять функції для таких типів HTTP запитів, як GET, POST, PUT, DELETE.

В системі присутні 3 класи користувачів: працівник, ментор, менеджер з управління персоналом. Ментор та менеджер виконують частково схожі управлінські функції, що надає можливість створити в клієнтській частині додатка спільні для обох компоненти користувацької взаємодії. Працівник отримує власні компоненти, що обмежені у функціоналі порівняно із ментором та менеджером.

Особливості побудови та роботи вебзастосунку розглянемо на прикладі побудови вебзастосунку за розглянутими принципами. При вході в систему користувача зустрічає вікно авторизації. Для того, щоб пройти авторизацію, потрібно ввести логін, пароль та натиснути відповідну кнопку.

Після авторизації користувачу відкриється вікно з профільними даними та доступними опціями системи залежно від типу користувача, що зображено на рис. 4. Для користувача типу "працівник" на цьому вікні додатково відображається статистика та рекомендації тренінгів.

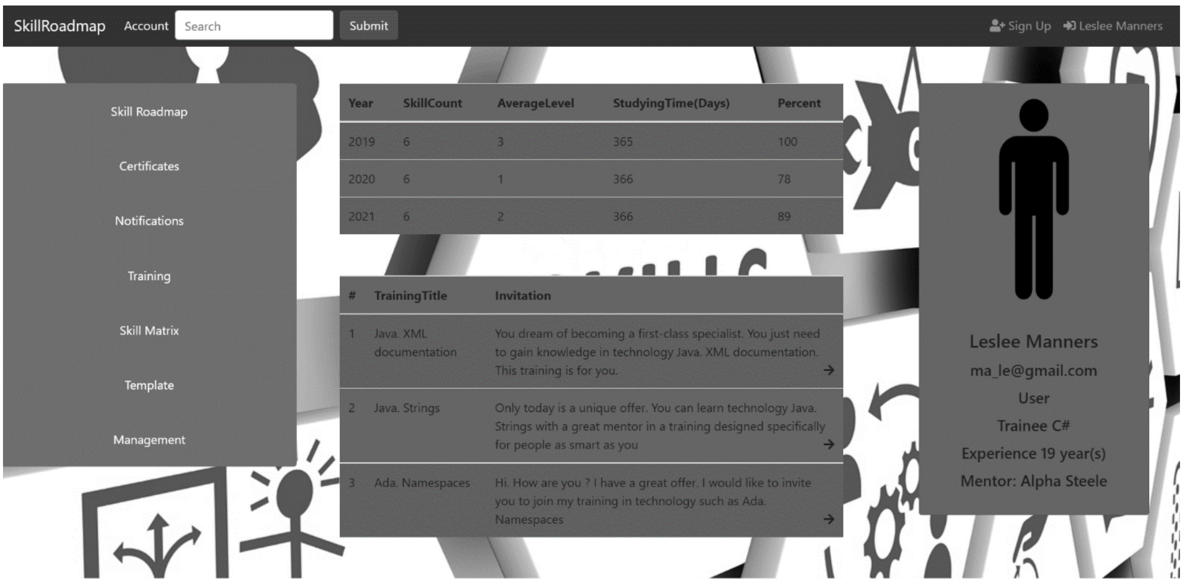

Рис. 4. Вікно профілю користувача

Якщо в систему зайшов користувач типу "Працівник", то йому фактично доступні лише функції перегляду даних, запис на тренінги та друк сертифікатів. Для прикладу розглянемо вікно із переглядом навичок на рис. 5.

В програмі доступна можливість сортувати навички за іменем, датою початку вивчення навички, датою закінчення вивчення, рівнем навички, категорією. Також доступний фільтр за категорією та за рівнем володіння навичкою.

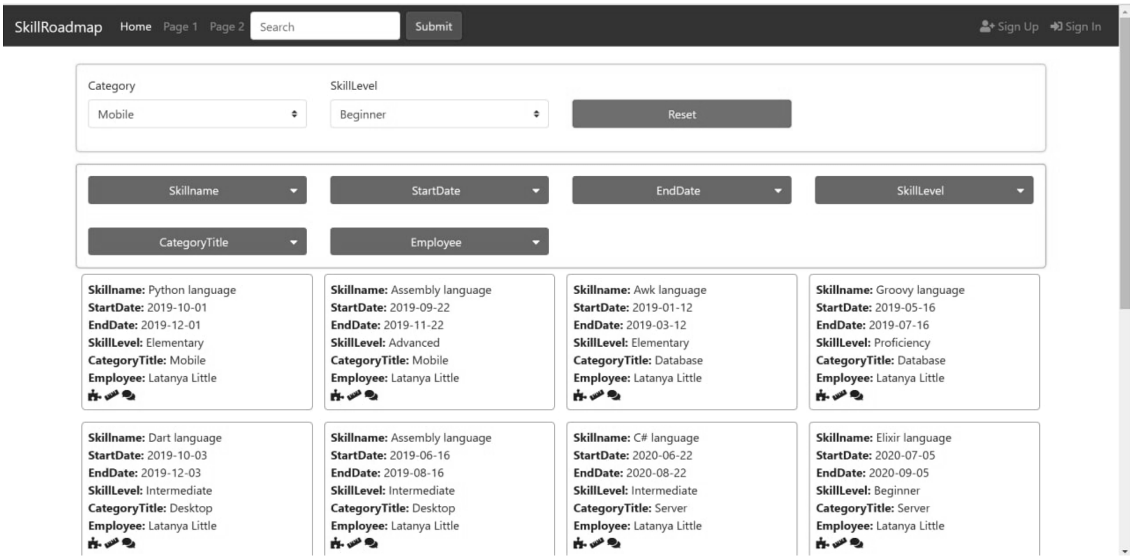

Рис. 5. Вікно навичок користувача

Для кожної навички доступні кнопки перегляду розділів, з якими було ознайомлення, перегляду метрик оцінки вивчення, перегляд коментарів. Для прикладу, на рис. 6. обравши перегляд розділів, відкриється нова таблиця уже з іншими можливостями сортування.

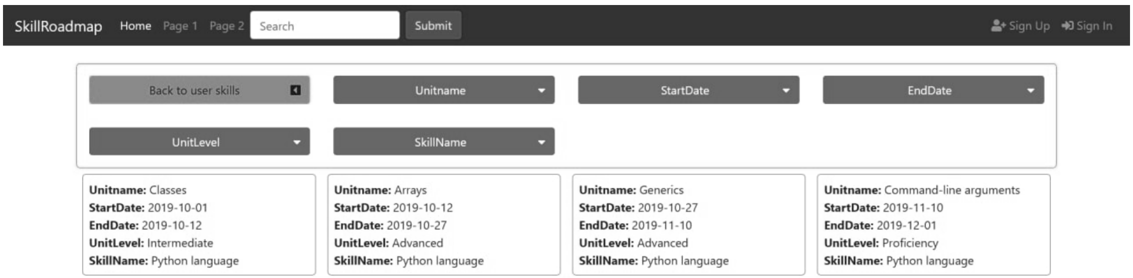

Рис. 6. Вікно розділів для навичок

Із профільної сторінки працівник може перейти на сторінку вибраного тренінгу, натиснувши на одну з рекомендацій, зображених у початковому вікні. На сторінці тренінгу користувач може стати його учасником, що зображено на рис. 7.

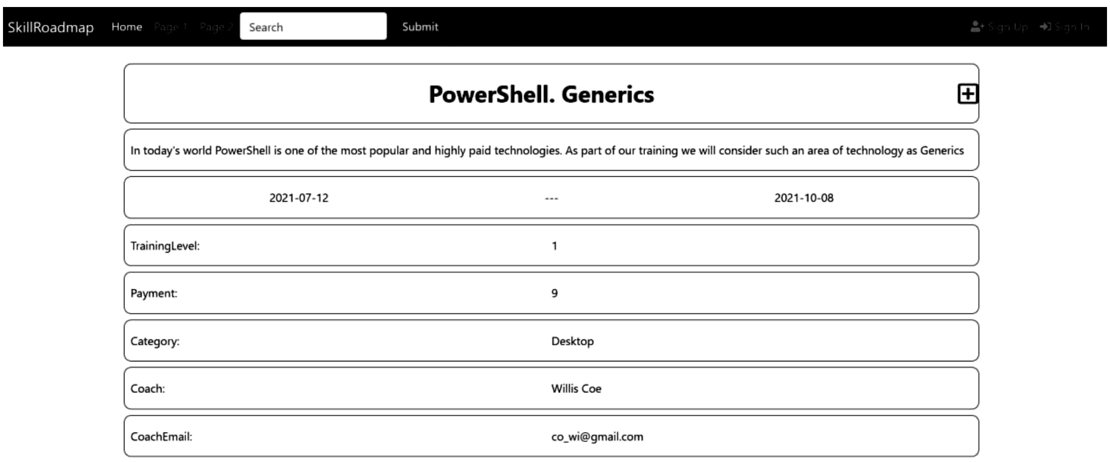

Рис. 7. Вікно інформації про тренінг

Перейшовши до вікна сертифікатів, що зображене на рис. 8, працівник може переглянути їх а також сформувати pdf-документ сертифіката, з можливістю друку в майбутньому.

Рис. 8. Сформований сертифікат у вигляді pdf-файлу

Важливою сторінкою з погляду візуального зображення є сторінка карти навичок працівника, на якій відображаються навички різним кольором за рівнем володіння, а також при наведенні на навичку відкривається додаткова інформація про неї. Ця сторінка зображена на рис. 9.

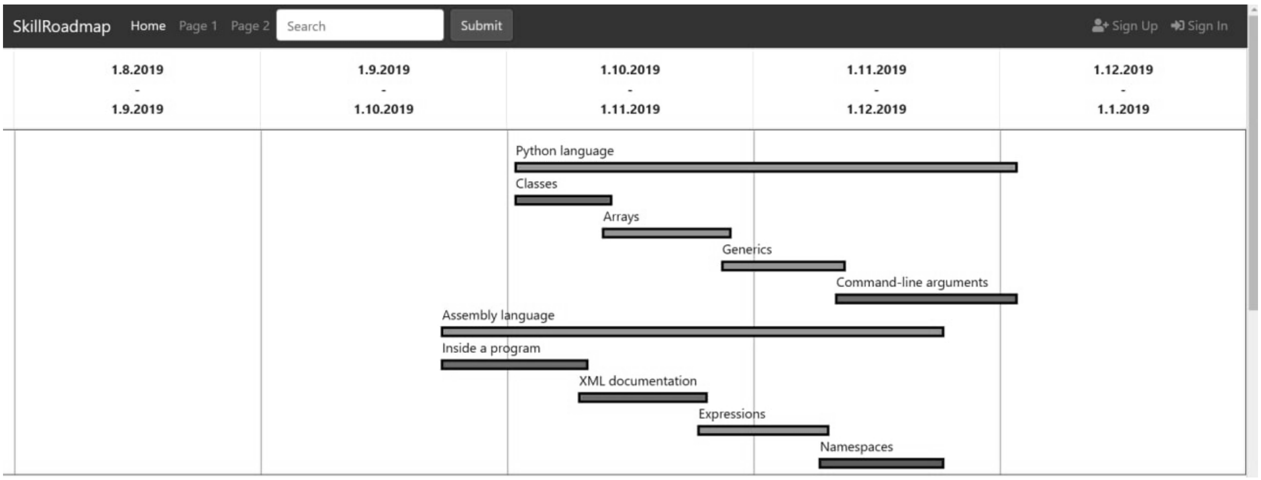

Рис. 9. Вікно карти навичок користувача

Користувачі типу "Ментор" та "Менеджер з управління персоналом" мають деякі окремі функції, проте основна їхня відмінність від "працівника" – це можливість додавати дані в систему за допомогою форм, створених окремо для навичок, тренінгів, шаблонів і т. д. Для прикладу обрано форму створення навички, що зображена на рис. 10.

Після заповнення форми і збереження навички користувач може побачити цю навичку в матриці навичок або на карті навичок працівника, відфільтрувавши навички за роками. Для того, щоб інформація про навичку була повною, також потрібно додати розділи, з яких складається навичка та метрики оцінки вивчення.

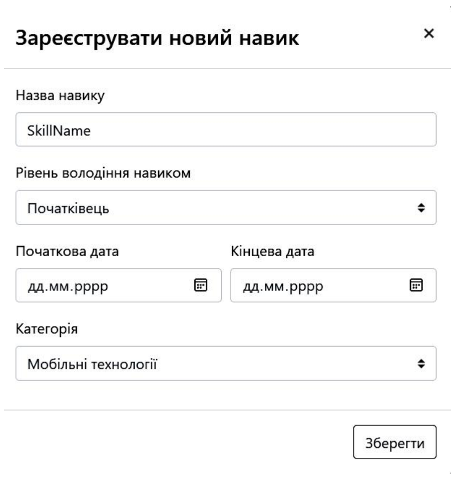

Рис. 10. Форма створення реєстраційного вікна

Рис. 11. Результати тестування навантаженості системи у Visual Studio

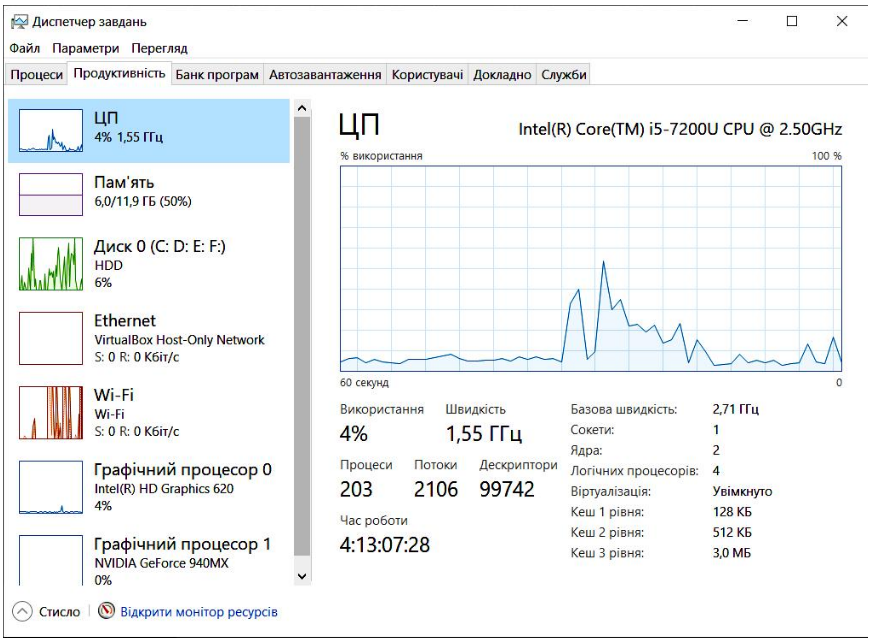

Рис. 12. Результати тестування навантаженості системи в Task Manager

Тестування продуктивності. При тестуванні продуктивності перевірено використання ресурсів серверної частини програми із використанням засобів Visual Studio, що продемонстровано на рис. 11.

Продуктивність клієнтської частини системи перевірено за допомогою Task Manager. Основним показниками, за якими велося спостереження, були використання процесора та оперативної пам'яті. Результат цього тестування показано на рис 12.

За результатами тестування визначено, що розроблений вебзастосунок працює швидше і використовує менше ресурсів комп'ютера, а саме до 150 МБ оперативної пам'яті та до 15 % можливостей процесора, що є приблизно на 10–15 % кращим показником за інші схожі застосунки для моніторингу навичок працівників. Це відбувається за рахунок ефективно підібраних принципів побудови цього застосунку, що дозволило системі швидше обробляти інформацію.

Висновки

За результатами досліджень окреслено мету та завдання створення застосунку, створено специфікацію вимог, описано архітектуру майбутнього додатку, спроєктована модель бази даних та поведінка системи, а також визначені головні сторінки майбутнього вебдодатку.

У результаті дослідження розроблено та досліджено принципи побудови вебзастосунку для моніторингу навичок працівників. При цьому потрібно виконати такі основні дії:

- обрати правильний метод збереження даних в базі даних, визначити інструменти, що будуть використовуватись для організації збереження та доступу до даних;
- спроєктувати простий для користувача інтерфейс основних сторінок вебаплікації, опираючись на функції та характеристику системи;
- провести реалізацію програмних рішень спроєктованих заздалегідь, визначено основні класи системи та написано програмний код відповідно до запланованих архітектурних шаблонів;
- описати роботу програми, її дії за різної поведінки користувача;
- проаналізувати вебзастосунки, які є частково схожими за функціоналом, описати їхні позитивні та негативні сторони, визначити переваги розроблюваної системи.

Окрім цього, окреслено мету та завдання створення застосунку, створено специфікацію вимог, описано архітектуру майбутнього додатку, спроєктовано модель бази даних та поведінку системи, а також визначено головні сторінки майбутнього вебдодатку.

Розглянуто особливості практичної реалізації системи. Описано основні класи, використовувані для реалізації серверної частини, такі як моделі, сервіси, контролери. Для клієнтської частини описано основні компоненти системи для відображення користувачеві. Проведено функціональне тестування, тестування інтерфейсу та тестування продуктивності системи. За результатами тестування визначено, що вебзастосунок використовує до 100 МБ оперативної пам'яті та до 15 % можливостей процесора середньої продуктивності.

Список літератури

1. Johnsen J. D. Technical Skill Assessment – Measuring Staff Capabilities and Optimizing Training Investments. [Electronic resource]. DOI: https://doi.org/10.3997/2214-4609-pdb.6.B06 (accessed: 27 September 2022).

2. Matviishyn Y. G. Strategic human resource management. Lviv, 2011. – 200 p.

3. Holt J. Process modelling using the UML. [Electronic resource]. – DOI: https://doi.org/10.1049/ic:20000366 (accessed: 27 September 2022).

4. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. USA, 2011. – 192 p.

5. Pukhovska L. P. Professional development of companies' personnel. Kyiv, 2015. – 175 p.

6. Ben-Gan I. Microsoft SQL Server 2012 T-SQL Fundamentals. USA, 2012. – 287 p.

7. Witold Litwin. SQL for Stored and Inherited Relations. [Electronic resource]. – DOI: https://doi.org/10.5220/0007676700370048 (accessed: 27 September 2022).

8. Furkan Mendi A. Hybrid Database Design Combination of Blockchain And Central Database. [Electronic resource]. – DOI: https://doi.org/10.1109/ismsit.2019.8932763 (accessed: 27 September 2022).

9. Hajian M. Progressive Web Apps with Angular. [Electronic resource]. – URL: https://doi.org/10.1007/978-1-4842-4448-7 (accessed: 27 September 2022).

10. Freeman A. Pro Angular. USA, 2017. – 322 p.

PRINCIPLES OF DEVELOPING A WEB APPLICATION FOR MONITORING EMPLOYEE SKILLS

V. Kulchytskyi[1], N. Pavych[2], Y. Paramud[1]

Lviv Polytechnic National University,
[1] Computer Engineering Department,
[2] Software Department

Summarizing the definition of the concept of "skills" from the reviewed literary sources, we determine that: skills are a combination of knowledge and skills that a person has acquired during his life. To improve skills, you need to choose certain tools (trainings, seminars, webinars, online courses, refresher courses, reading special literature, etc.). Personal qualities and attitudes of young people are important in choosing tools and achieving results. As part of this study, the principles of building a web application for monitoring the skills of employees were designed and developed. The web resource has a convenient user interface, the ability to view an employee's skill map, leave comments on a specific employee skill, allows users to create and sign up for existing training courses, generate certificates confirming mastery of a skill, and view statistics. Unlike other analogs of programs for monitoring skills, this system provides for the interaction of three types of users: manager, mentor, and employee. The manager's options include: creating a technology ownership template, personnel management, skills management, certificate management, notification management; Mentor options include: skill management, training management, notification management; Employee options include the following: skill management, training management, notification management, certificate management, view statistics, view recommendations.

Keywords: principles of building a web application, monitoring, skill, databases, employee, mentor, manager, certificate.