СТЕК ТЕХНОЛОГІЙ

В теперішніх умовах пріоритетною платформою для розробки є веб. Адже з розвитком телекомунікації та розповсюдженням інтернету, все більше і більше людей переходять до використання додатків у своєму веб-браузері. Відповідно і ринок потребує нових продуктів.

Головною мовою розробки цього проекту є мова - **C#**, але головною особливістю є написання на такому архітектурному шаблоні програмування як **MVC**.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: **модель даних**, **вигляд** (інтерфейс користувача) та **модуль керування**. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Чому саме **MVC**? У MVC чітко розділена логіка програми, а це, в свою чергу, на багато спрощує роботу над великими проектами. Переваги цього методу:

1. Швидкий процес розробки, підтримує швидке і паралельний розвиток.
2. З MVC один програміст може працювати над виставою, а інший може працювати над контролером для створення бізнес-логіки.
3. Додаток, розроблене з його застосуванням, в три рази швидше, ніж додаток, розроблене з іншими шаблонами розробки.
4. Можливість надання декількох видів.
5. У MVC можна створювати кілька подань.
6. Копіювання дублікатів дуже обмежена, оскільки воно відокремлює дані і логіку від дисплея.
7. Підтримка асинхронної технології, яка допомагає розробникам розробляти швидко завантажувати додаток.
8. Модифікація не впливає на всю модель, тому що частина моделі не залежить від частини переглядів. Тому будь-які зміни в Моделі не впливатимуть на всю архітектуру.
9. Шаблон NET MVC повертає дані без застосування будь-якого форматування, тому одні й ті ж компоненти можуть використовуватися і викликатися для використання з будь-яким інтерфейсом.
10. За допомогою цієї платформи дуже легко розробляти URL-адреси, оптимізовані для SEO, для отримання більшої кількості відвідувань з певної програми.

Також переваги **C#**:

1. Ефективна побудова складних ієрархій класів з можливістю їх зручної модифікації. Роботу класів в ієрархії можна змінювати шляхом додавання нових успадкованих класів у потрібному місці ієрархії;
2. Повторне використання раніше написаного коду з подальшою його модифікацією під виконувану задачу. У свою чергу, новостворений код також може використовуватись на нижчих ієрархіях класів;
3. Зручність у супроводі (доповненні) програмного коду шляхом введення нових класів з новими можливостями;
4. Зменшення кількості логічних помилок при розробці складних програмних систем. Повторно використовуваний код частіше тестується, а, отже, менша ймовірність наявності в ньому помилок;
5. Полегшення в узгодженні різних частин програмного коду шляхом використання інтерфейсів. Якщо два класи успадковані від спільного нащадка, поведінка цих класів буде однаковою у всіх випадках. Це твердження виходить з вимоги, що схожі об’єкти повинні мати схожу поведінку. Саме використання інтерфейсів зумовлює схожу поведінку об’єктів;
6. Створення бібліотек коду, які можна використовувати та доповнювати власними розробками;
7. Можливість реалізовувати відомі шаблони проектування для побудови гнучкого коду, який не змінює попередні розробки;
8. Використання переваг поліморфізму неможливе без спадковості. Завдяки поліморфізму забезпечується принцип: один інтерфейс – декілька реалізацій;
9. Забезпечення дослідного програмування (швидкого макетування). Таке програмування застосовується у випадках, коли цілі та вимоги до програмної системи на початку є розпливчасті. Спочатку створюється макет структури, потім цей макет поетапно удосконалюється завдяки успадкуванню попереднього. Процес триває до отримання бажаного кінцевого результату;
10. Краще розуміння структури програмної системи програмістом завдяки природному представленню механізму спадковості. Якщо при побудові складних ієрархій пробувати використовувати інші принципи, то це може значно ускладнити розуміння всієї задачі і призведе до збільшення кількості помилок.