

DOKUMENT TECHNICZNY PROJEKTU STRONY INTERNETOWEJ DO ZARZĄDZANIA DOMOWĄ BIBLIOTEKĄ

PRZEDMIOT: PROGRAMOWANIE ZAAWANSOWANE

AUTOR: WOJCIECH JANKOWSKI

1. Tematyka projektu

Tematyką projektu jest stworzenie strony internetowej o architekturze klient – serwer. Projekt zakłada stworzenie aplikacji internetowej do zarządzania domową biblioteką. Strona internetowa docelowo będzie posiadać następujące funkcjonalności:

- Rejestracja nowych użytkowników
- Logowanie użytkowników
- Możliwość przeglądania książek, które użytkownik obecnie czyta
- Możliwość przeglądania dostępnych książek
- Dodawanie nowych książek do bazy
- Dodanie książki do aktualnie czytanych przez użytkownika

2. Opis techniczny projektu

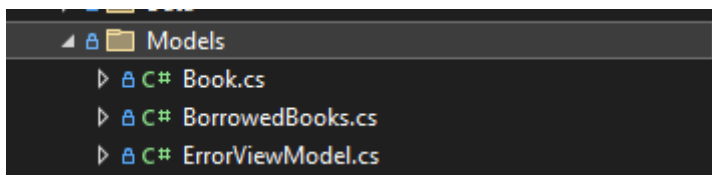
Aplikacja będzie napisana w języku C# oraz w oparciu o projekt MVC (Model – View – Controller). Aplikacja będzie tak naprawdę stroną internetową, która będzie wymagała zalogowania się użytkownika na stronie aby mógł on z niej skorzystać. Dane o użytkownikach takie jak np. imię, nazwa, email, hasło będą przechowywane w tabeli w bazie danych. W taki sam sposób będą przechowywane informacje o książkach dostępnych w bibliotece oraz o książkach, które są obecnie w trakcie czytania przez użytkowników

3. Potencjalne możliwe problemy i zagrożenia

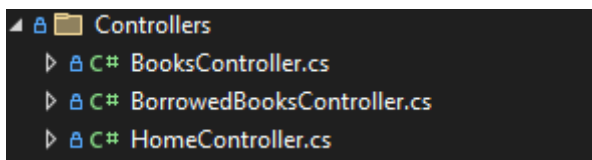
Potencjalnym problem korzystania z aplikacji może być brak dostępu do Internetu ponieważ aplikacja będzie działała na stronie internetowej i bez dostępu do Internetu nie będzie możliwości korzystania z niej. Problematyczne mogą okazać się także błędy jakie popełnią użytkownicy podczas korzystania z aplikacji np. podczas wprowadzania nowych książek do bazy danych mogą popełnić oni literówki albo wprowadzą złe wartości do komórek co spowoduje, że dane w bazie danych będą błędnie zapisane.

4. Implementacja

Aplikacja posiada dwie tabele do gromadzenia danych: w tabeli „Book” aplikacja przechowuje informacje o wszystkich dostępnych książkach. Natomiast tabela „BorrowedBooks” przechowuje informacje o aktualnie wypożyczonych książkach przez użytkownika.



Aplikacja posiada także 2 główne kontrolery, „BooksController” oraz „BorrowedBooksController” które odpowiedzialne są za wyświetlanie książek oraz za obsługę przycisków.



Metoda „Assign” w kontrolerze „BooksController” odpowiada za obsługę przycisku wypożycz. Metoda ta sprawdza najpierw czy książkę nie jest już wypożyczona, jeśli książka jest dostępna do wypożyczenia to dodaje Id książki do tabeli „BorrowedBooks” , Id użytkownika oraz datę wypożyczenia książki.

```
public ActionResult Assign(int Id, string UserId)
{
    using (_context)
    {
        var book = _context.Book.Find(Id);
        if (book.IsBorrowed == true)
        {
            return View("~/Views/Home/Blad.cshtml");
        }
        else
        {
            book.UserId = User.FindFirstValue(ClaimTypes.NameIdentifier); ;
            book.IsBorrowed = true;

            // Dodaj nowy rekord do tabeli "BorrowedBooks"
            var borrowedBook = new BorrowedBooks
            {
                BookId = book.Id,
                UserId = User.FindFirstValue(ClaimTypes.NameIdentifier),
                BorrowDate = DateTime.Now
            };
            _context.BorrowedBooks.Add(borrowedBook);

            _context.SaveChanges();
            return View("~/Views/Home/Confirm.cshtml");
        }
    }
    return RedirectToAction("Index");
}
```

Metoda return odpowiada za obsługę przycisku zwróć. Metoda ta usuwa książkę z tabeli wypożyczonych

```
public ActionResult Return(int Id, string UserId)
{
    using (_context)
    {
        var book = _context.Book.Find(Id);
        if (book.IsBorrowed == false)
        {
            return View("~/Views/Home/Blad1.cshtml");
        }
        else
        {
            book.UserId = UserId;
            book.IsBorrowed = false;

            // Znajdź i usuń wypożyczoną książkę z tabeli "BorrowedBooks"
            var borrowedBook = _context.BorrowedBooks.Where(x => x.BookId == book.Id).FirstOrDefault();
            _context.BorrowedBooks.Remove(borrowedBook);

            _context.SaveChanges();
            return View("~/Views/Home/Confirm1.cshtml");
        }
    }
    return RedirectToAction("Index");
}
```

Drugi kontroler jest odpowiedzialny tylko za wyświetlanie tabeli z wypożyczonymi książkami.

```
namespace HomeLibrary1.Controllers
{
    1 odwołanie
    public class BorrowedBooksController : Controller
    {
        private readonly UserManager<HomeLibrary1User> _userManager;
        private readonly ApplicationDbContext _context;
        Odwołania: 0
        public BorrowedBooksController(ApplicationDbContext context)
        {
            _userManager = _userManager;
            _context = context;
        }
        [Authorize]
        // GET: BorrowedBooks
        Odwołania: 0
        public async Task<IActionResult> Index()
        {
            var applicationDbContext = _context.BorrowedBooks.Include(b => b.Book).Include(b => b.User);
            return View(await applicationDbContext.ToListAsync());
        }
    }
}
```