

Movie 4U

Dokumentacja projektu

Autorzy:	Numer Indeksu
Piotr Kozak	173960
Robert Korus	173899
Dawid Holko	173601

Spis treści

1. Repozytorium	3
Adres repozytorium	3
Opis repozytorium	3
Opis poszczególnych branch'y	3
Znane problemy przy pobieraniu aplikacji z repozytorium	4
2. Działająca aplikacja.....	6
Adres gotowej i działającej aplikacji	6
Wygląd gotowej aplikacji.....	6
Domyślne dane logowania:.....	6
Uwagi	7
3. Opis zależności w kodzie aplikacji	7
4. Nerozwiazane problemy	8

1. Repozytorium

Adres repozytorium

<https://github.com/pepe67/ProjectPP5>

Opis repozytorium

W Repozytorium projektu jest dostępnych 7 branch'y, które ukazują postęp prac nad projektem. Branche na liście w repozytorium są ułożone alfabetycznie, lecz prawidłowa ich kolejność wygląda następująco:

1. Master
2. FOSUB&Template
3. Categories
4. Movies
5. Cart_And_Comments
6. Orders
7. AdminPanel

Opis poszczególnych branch'y

1. Master – branch początkowy zawierający jedynie czystą instalację Symfony
2. FOSUB&Template – Dodane zostały formularze logowania, rejestracji i obsługa kont poprzez „wtyczkę” FOSUserBundle oraz został wdrożony zakładany wygląd strony.
3. Categories – Dodane zostały kategorie (klasa *Category*, formularze i ich widoki), do których są przypisywane filmy utworzone w kolejnym branchu. Zaimplementowano wyświetlanie nazw kategorii w menu. Wstępnie dodano wyświetlanie filmów z kategorii (filmów brak, jednak wyświetlają się dane domyślne z szablonu)
4. Movies – Dodano klasę która odpowiedzialna jest za obsługę filmów. Tabela *Category* została połączona z *Movie* relacją jeden-do-wielu. Zakładamy więc, że każdy film ma jedną kategorię, a kategoria może mieć wiele filmów. Dokończono wyświetlanie filmów z kategorii, oraz zaimplementowano widok pojedynczego filmu.

5. `Cart_And_Comments` – Dodano obsługę koszyka, działającego w oparciu o sesje. Filmy dodawane do koszyka zapisywane są w sesji. Użytkownik może wyświetlić swój koszyk, dodać przedmiot i go usunąć. Przeliczana jest również wartość koszyka.
Dodano obsługę komentarzy. Formularz dodawania komentarza oraz komentarze, wyświetlane są w zakładce Opinie użytkowników w widoku filmu. Komentarze po dodaniu wymagają akceptacji przez administratora.
6. `Orders` – Dodano obsługę zamówień. Użytkownik generuje zamówienie poprzez dodanie filmów do koszyka i kliknięciu w przycisk „złóż zamówienie” w koszyku. Niestety brak implementacji płatności, dlatego testowo został udostępniony link w „Moich zamówieniach” do „akceptacji” płatności. Użytkownik po poprawnym opłaceniu zamówienia ma możliwość oglądać film (link Oglądaj – klikalny, lecz brak implementacji odtwarzania filmu). Dodano również streowany zwiastun w widoku filmu.
7. `AdminPanel` – Branch jest próbą zebrania wszystkich linków które powinny być dostępne jedynie dla administratora, tj. dodawanie, edycja i usuwanie filmów, akceptacja komentarzy, dodawanie edycja i usuwanie kategorii. Zmieniono routingu tak, by powyższe czynności wymagały uprawnień administratora tj. {adres aplikacji} /admin/ {czynność} . Zgodnie z ustawieniem w pliku `app/config/security.yml` adresy {adres aplikacji}/admin wymagają roli administratora. Próba wejścia np. w edycję filmu skutkuje przekierowaniem do formularza logowania.

Znane problemy przy pobieraniu aplikacji z repozytorium

1. Są pewne pozostałości z drugiego Brancha, które zostały nadpisane w późniejszym czasie. Próby usunięcia tych pozostałości skutkowały ponownym utworzeniem tych plików. Po pobraniu aplikacji z repozytorium należy usunąć wszystkie katalogi z katalogu **`app/Resources/FosUserBundle/views`** . Należy pamiętać, by zostawić plik **`layout.html.twig`**.
2. Plik `parameters.yml` z katalogu `app/config/` nie jest pobierany z repozytorium (z racji wrażliwych danych), dlatego po pobraniu aplikacji plik powinien posiadać następującą strukturę:

```
parameters:
  database_driver: pdo_mysql
  database_host: 127.0.0.1
  database_port: null
  database_name: projekt
  database_user: root
  database_password: root
  mailer_transport: smtp
  mailer_host: 127.0.0.1
  mailer_user: null
  mailer_password: null
  locale: pl
  secret: ThisTokenIsNotSoSecretChangeIt
  dotpay_id: 72890
  dotpay_pin: 1234567887654321
```

2. Działająca aplikacja

Adres gotowej i działającej aplikacji

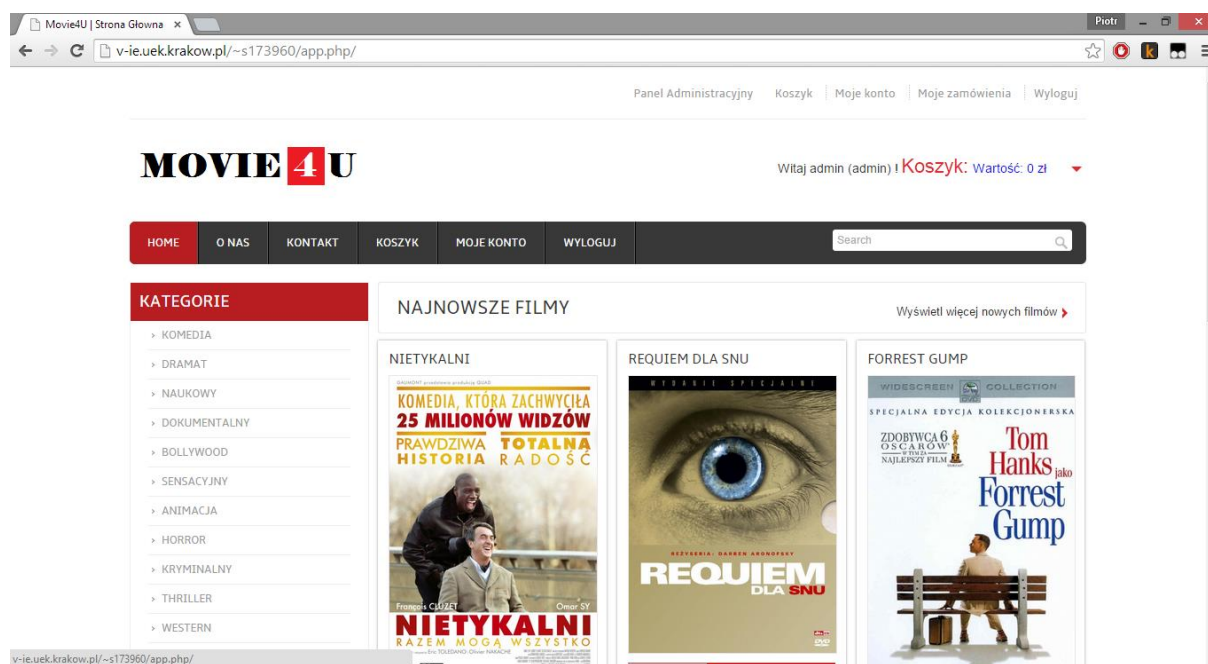
Adres developerski (używa cache generowanego w locie):

http://v-ie.uek.krakow.pl/~s173960/app_dev.php

Adres główny (używa cache wygenerowanego):

<http://v-ie.uek.krakow.pl/~s173960/app.php>

Wygląd gotowej aplikacji



Domyślne dane logowania:

Konto administratora:	Konto użytkownika:
Login: admin	Login: user
Hasło: admin	Hasło: user

Uwagi

W aplikacji nie zbudowaliśmy wszystkich wyświetlanych (aczkolwiek nieobowiązkowych) odnośników. Przykładowo odnośnik **Kontakt** w menu górnym nie odnosi się do niczego. Jest to odnośnik z **#** w atrybucie **href=""**

3. Opis zależności w kodzie aplikacji

- Utworzenie konta administratora jest możliwe po zmianie linii
`$this->roles = array('ROLE_USER');`
na:
`$this->roles = array('ROLE_ADMIN');`
W pliku: *src/ProjectBundle/Entity/User.php*
- Główny plik layout'u (**layout.html.twig**) umieszczony jest w katalogu:
src/ProjectBundle/Resources/views
Odnoszą się do niego wszystkie zdefiniowane i używane widoki w aplikacji.
- W katalogu *src/ProjectBundle/Resources/views/Default* umieszczone są wszystkie widoki, których używa aplikacja do wyświetlenia danych potrzebnych użytkownikowi.
- Pliki widoków w katalogach: *Admin*, *Category*, *Comment*, *Movie* w lokalizacji *src/ProjectBundle/Resources/views/* są używane przez panel administracyjny.
- Pliki widoków w katalogu *MovieOrder* w lokalizacji *src/ProjectBundle/Resources/views/* używane są zarówno przez administratora jak i użytkownika, który składa zamówienie.
- Strona w większości zapytań używa jednego kontrolera *Defaultcontroller.php* w którym zdefiniowano akcje dostępne dla użytkownika. W przypadku przeglądania i składania zamówień używany jest *MovieOrderController.php*
- Pozostałe kontrolery wykorzystywane są przez panel administracyjny.

- Brak implementacji płatności sprawia, że kontroler **PaymentController.php** nie jest używany, jednak zdefiniowano dla niego routingi.
- Pliki css, font, js, oraz pliki graficzne używane w aplikacji zostały umieszczone w katalogu *src/ProjectBundle/Resources/public/*. Każdy typ plików umieszczony jest w osobnym katalogu, by zachować pewien porządek.
- Zamówienia posiadają dwa statusy w bazie danych.
0 oznacza brak płatności i możliwość wywołania formularza płatności
1 oznacza sukces w realizacji zamówienia i możliwość oglądania filmu.

4. Nierozwiązane problemy

- Największym problemem stała się realizacja płatności. O ile strona płatności dotpay otrzymywała wszystkie potrzebne dane, o tyle nie udało nam się zaimplementować zmiany statusu zamówienia. Dlatego też, po zapisaniu zamówienia do bazy danych przyjmuje ono status **0**, użytkownik zostaje przekierowany na stronę z jego zamówieniami.
By zasymulować opłacenie zamówienia na stronie z zamówieniami można ręcznie wywołać zmianę statusu zamówienia poprzez kliknięcie odnośnika **Zapłać bez handlera** pomijając tym samym system dotpay.