Dokumentacja projektu *Gratia Artis*

1. ***Charakterystyka oprogramowania***

*Gratia Artis* to strona umożliwiająca młodym artystom sprzedaż ich prac bez pośredników. Wystarczy zarejestrować się na stronie, dodać ofertę ze zdjęciem, a każdy użytkownik serwisu będzie w stanie kupić wybrane dzieło sztuki. Podstawowe funkcjonalności aplikacji to:

* Logowanie,
* Zamieszczenie produktu,
* Strona sklepu z widocznymi produktami
* Możliwość dodania produktu do koszyka oraz zakupu
* Kontakt z twórcami
* Strona profilu użytkownika

1. ***Prawa autorskie***

Autorami aplikacji są Joanna Cieszyńska i Katarzyna Kowalska.

Projekt udostępniany jest na licencji zamkniętej – zostaje objęty restrykcjami dotyczącymi używania, kopiowania lub modyfikacji. Repozytorium kodu jest prywatne.

1. ***Specyfikacja wymagań***

Sprzedaż prac często jest nie lada wyzwaniem dla młodych i początkujących artystów. Nawiązanie współpracy z galerią czy serwisem udostępniającym prace na sprzedaż może być problematyczne. Wiąże się to również z koniecznością podziału zysków ze sprzedaży. Istnieje możliwość udostępniania swoich prac na portalach społecznościowych, co pozwala na sprzedaż bez pośredników i łatwy dostęp do sztuki. Jednak w tym przypadku ogromna ilość innego rodzaju informacji na tego typu portalach powoduje, że prace mogą pozostać niezauważone. Jest to problematyczne zarówno dla twórców, jak i miłośników sztuki.

Rozwiązaniem powyższego problemu jest przygotowanie strony internetowej o nazwie Gratia Artis, co w wolnym tłumaczeniu oznacza „dla sztuki”. Ma ona służyć jako okrojony portal społecznościowy, skupiający się wyłącznie na stworzeniu portfolio, udostępnianiu prac oraz ich sprzedaży. Istnienie takiego portalu może zapewnić artystom większą kontrolę nad sprzedażą ich dzieł, oraz platformę do dzielenia się ich sztuką ze światem.

W zakres projektu wchodzą:

* Udostępnianie swoich prac
* Strona profilowa zawierająca portfolio użytkownika
* Strona zawierająca prace wszystkich użytkowników
* Koszyk z możliwością zakupu
  1. ***Wymagania funkcjonalne***

|  |  |  |  |
| --- | --- | --- | --- |
| ***ID*** | ***Nazwa*** | ***Opis*** | ***Priorytet*** |
| U001 | Utworzenie konta | Użytkownik ma możliwość utworzenia konta umożliwiającego mu korzystanie z usług sytemu | 1 |
| U002 | Logowanie | Użytkownik może zalogować się na istniejące konto | 1 |
| U003 | Konfiguracja | Użytkownik ma możliwość konfiguracji swojego konta, w szczególności ustawienia i zmiany hasła | 1 |
| P001 | Profil użytkownika | Użytkownik ma kontrolę nad swoimi pracami wyświetlanymi w profilu | 2 |
| P002 | Udostępnianie prac | Użytkownik może zamieścić swoją pracę na stronie wraz z opisem i ceną | 3 |
| P003 | Przeglądanie prac wszystkich użytkowników | Użytkownik może przeglądać prace zamieszczone przez innych użytkowników | 3 |
| P004 | Kontakt | Użytkownik może skontaktować się z twórcami strony drogą mailową | 2 |
| P005 | FAQ | Strona oferująca odpowiedzi na najczęściej zadawane pytania | 1 |
| S001 | Zakupy | Możliwość zakupu prac dostępnych na stronie | 3 |

* 1. ***Wymagania niefunkcjonalne***

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | ***Nazwa*** | ***Opis*** | ***Priorytet*** |
| N001 | Niezawodność | Strona powinna działać sprawnie przy wysokiej aktywności czy rozdzielczości plików; powinna odpowiadać na zapytania w czasie nie dłuższym niż 5 sekund | 2 |
| N002 | Użyteczność | Strona powinna być łatwa w użyciu, zrozumiała i atrakcyjna dla użytkownika | 2 |
| N003 | Pielęgnowalność | Strona powinna pozwalać na łatwą modyfikację w celu naprawy defektów czy dostosowania do nowych wymagań w celu ułatwienia przyszłego utrzymania lub zmian w środowisku | 2 |
| N004 | Przenaszalność | Strona powinna pozwalać na łatwe przeniesienie z jednego środowiska sprzętowego do innego | 1 |
| N005 | Solidność | Strona powinna oferować jak najmniejsze prawdopodobieństwo utraty danych i krótki czas uruchomienia po awarii | 1 |

1. ***Architektura systemu/oprogramowania.***
   1. ***Architektura rozwoju.***

Oprogramowanie powstało w programie Visual Studio Code przy wykorzystaniu frameworku Ionic pozwalającego na tworzenie aplikacji hybrydowych. Do przechowywania danych dotyczących rejestracji/logowania użytkowników, publikowania obrazów i wyświetlania ich zastosowano Firebase – rozwiązanie Backend as a Service oferujące system uwierzytelniania użytkowników, bazę danych czy przechowywanie plików. Dodatkowo zastosowano biblioteki:

* AngularFire
* Angular/http
* Email Composer, z pluginem Cordova Email Plugin
  1. ***Architektura uruchomieniowa***

Aby uruchomić aplikację niezbędne jest utworzenie w wybranym folderze projektu ionic oraz instalacja bibliotek:

* ionic start <name> <template>
* npm install firebase @angular/fire
* npm install @angular/http@latest
* ionic cordova plugin add cordova-plugin-email-composer
* npm install @ionic-native/email-composer

Następnie należy przenieść pliki znajdujące się w repozytorium projektu do utworzonego folderu, oraz wykonać polecenie ionic serve.Aplikacja uruchomi się w przeglądarce. Do skorzystania z funkcji wysyłania wiadomości niezbędny jest program Android Studio.

1. ***Testy***
   1. ***Scenariusze testów***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Nazwa** | **Warunki wstępne** | **Kroki wykonania** | **Oczekiwany rezultat** |
| U1. | Założenie konta | Brak użytkownika o podanych danych w bazie | -podanie loginu i hasła  -kliknięcie ‘zarejestruj’ | Założenie konta użytkownika, które widoczne jest w bazie, możliwość zalogowania |
| U2. | Zalogowanie się | Uprzednia rejestracja użytkownika w bazie | -podanie loginu i hasła  -kliknięcie ‘zaloguj się’ | Zalogowanie się do konta, dostęp do podstron widocznych tylko dla zalogowanych użytkowników |
| P3. | Wyświetlenie ofert | Zalogowanie się, dostęp do podstrony z ofertami | -Przejście na podstronę  -wyświetlenie obrazów | Możliwość obejrzenia oferowanych produktów |
| S4. | Dodanie obrazu do koszyka | Dostęp do podstrony z ofertami | -Przejście na podstronę  -Naciśnięcie przycisku odpowiadającego za dodanie do koszyka | Obraz dodany do koszyka powinien wyświetlać się na liście po otwarciu koszyka |
| S5. | Zakup obrazu | Dostęp do podstrony, dodanie obrazu do koszyka, wyświetlenie koszyka | -wyświetlenie koszyka  -wybranie ilości | Komunikat informujący użytkownika o dokonaniu zakupu |
| P6. | Kontakt z twórcami | Dostęp do podstrony ‘kontakt’ – zalogowany użytkownik | -podanie adresu oraz treści wiadomości  -naciśnięcie odpowiedniego przycisku | Wiadomość email zostaje wysłana i dostarczona na adres właścicieli strony |
| P7. | Wyświetlenie FAQ | Dostęp do podstrony FAQ | -przejście na podstronę  -wybór pytania | Wybrane pytanie zwraca komunikat z odpowiedzią |
| P8. | Dodanie oferty | Zalogowanie się,  Kontakt z bazą danych | -wybór obrazu z listy plików  -przejście do podstrony z ofertami | Komunikat oznajmiający o powodzeniu,  Dodanie obrazu do bazy, obraz dostępny w zakładce sklepu |
| U9. | Zmiana zdjęcia profilowego | Zalogowanie się, dostęp do zakładki profilu użytkownika | -wybranie zdjęcia profilowego  -wybór nowego obrazu z listy  -zatwierdzenie wyboru hasłem | Zmiana zdjęcia wyświetlanego w profilu użytkownika,  komunikat potwierdzający zmianę |
| U10. | Zmiana danych | Zalogowanie się,  Dostęp do zakładki profilu użytkownika | -podanie nowych danych,  -zatwierdzenie zmiany starym hasłem | Zmiana danych logowania, oraz danych wyświetlanych na profilu |

* 1. ***Sprawozdanie z wykonania scenariuszy testów***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID.** | **Nazwa** | **Warunki wstępne** | **Kroki wykonania** | **Oczekiwany rezultat** |
| U01 | Próba rejestracji ze źle sformułowaną nazwą użytkownika | Istnienie interfejsu rejestracji | Wprowadzenie nazwy użytkownika o nieprawidłowej składni | Komunikat informujący o złym formacie nazwy,  Niepowodzenie rejestracji |
| U02 | Próba rejestracji z hasłem krótszym niż 6 znaków | Istnienie interfejsu rejestracji | Wprowadzenie hasła krótszego niż 6 znaków | Komunikat informujący o wymaganej długości hasła,  Niepowodzenie rejestracji |
| U03 | Próba rejestracji bez nazwy użytkownika | Istnienie interfejsu rejestracji | Pominięcie nazwy użytkownika | Komunikat informujący o złym formatowaniu nazwy, Niepowodzenie |
| U04 | Próba rejestracji bez hasła | Istnienie interfejsu rejestracji | Pominięcie hasła użytkownika | Komunikat informujący o wymaganiach co do długości hasła, Niepowodzenie |
| U05 | Próba zalogowania się jako niezarejestrowany użytkownik | Baza danych przechowująca dane użytkowników, uwierzytelnianie | Logowanie jako nieistniejący użytkownik | Komunikat „użytkownik nie istnieje” |
| U06 | Próba zalogowania się z błędnym hasłem | Baza danych przechowująca dane użytkowników, uwierzytelnianie | Logowanie z nieprawidłowym hasłem | Komunikat informujący o błędnym haśle |
| U07 | Logowanie jako uprzednio zarejestrowany użytkownik | Istnienie interfejsu rejestracji, Baza przechowująca dane użytkowników | Zalogowanie się jako istniejący użytkownik, przy użyciu poprawnych danych | Użytkownik zostaje przeniesiony do systemu |
| U08 | Zmiana danych użytkownika | Istnienie interfejsu rejestracji, Baza przechowująca dane użytkowników | Zalogowanie się, edycja profilu użytkownika, podanie nowego avatara, loginu, hasła | Dane użytkownika zostają zmienione, wyświetlenie komunikatu i przeniesienie na stronę sklepu |
| P09 | Dodanie obrazu | Istnienie połączenia z chmurą i bazą danych | Wybranie obrazu, wysłanie do bazy | Obraz pojawia się w chmurze, wyświetlenie komunikatu o powodzeniu i przeniesienie na stronę sklepu, obraz widoczny na liście |
| S10 | Zakup obrazu | Dostęp do podstrony sklepu, dodanie obrazu do koszyka, wyświetlenie koszyka | -wyświetlenie koszyka  -wybranie ilości | Komunikat informujący użytkownika o dokonaniu zakupu |
| S11 | Dodanie obrazu do koszyka | Dostęp do podstrony sklepu,  Obrazy dodane do bazy danych | -wyświetlenie podstrony sklepu  -naciśnięcie przycisku odpowiadającego za dodanie przedmiotu do koszyka | Zwiększenie się liczby przedmiotów w koszyku, zsumowanie cen |