

## Programowanie w języku Java

Temat projektu:	<b>Biuro podróży</b>
Zespół:	<b>Konrad Michalski Adam Kurczyński</b>
Rok studiów: 2	<b>Grupa: 2ID14A</b>

Prezentowana przez nas aplikacja to biuro podróży. Uruchamiana jest w przeglądarce. Umożliwia nam ona przegląd zdjęć oraz opis najciekawszych miejsc na świecie, które ma w swojej ofercie biuro. Po zdecydowaniu i wyborze odpowiedniego terminu wycieczki, mamy możliwość jej rezerwacji, a w kolejnych krokach kupienia nie więcej niż 10 biletów. Dodatkowo przy pierwszym podsumowaniu naszej oferty jest możliwość edycji wybranych przez nas wycieczek. Aplikacja jest czytelna i intuicyjna w obsłudze. Wszystkie dane prezentowane są w przejrzystej formie. Aplikacja korzysta z bazy danych, która zapisuje dane użytkownika i wszystkie dane dotyczące transakcji.

**Krok 1.** Po uruchomieniu mamy do wyboru dwie opcje, login lub rejestrację. Sprawdzana jest poprawność wprowadzanych danych znajdujących się w bazie danych. Jeżeli użytkownik już istnieje lub hasło jest niepoprawne, nie możemy przejść do kolejnego kroku.

**Krok 2.** Do wyboru mamy wycieczki z 4 krajów, przy każdym mieście znajdują się przyciski, które wyświetlają informacje o miejscu, terminie, ostatni z nich dodaje wycieczkę do listy zarezerwowanych. Po prawej stronie mamy koszt zamówienia.

**Krok 3.** Potwierdzenie zamówienia oraz ostatnie możliwe zmiany w zamówieniu. Aplikacja prosi o więcej danych osobistych osoby oraz kontakt.

**Krok 4.** Wyświetlane zostają informacje na temat osoby składającej zamówienie, wartości zamówienia i bilety do miejsc, które zostały wybrane.

**Krok 5.** Podziękowanie za złożenie zamówienia.

Projekt stworzony jako projekt Maven, z wykorzystaniem środowiska IntelliJ IDEA

W projekcie wykorzystano:

- Język Java, HTML, CSS, JavaScript
- Baza danych SQLite
- Framework Spring MVC, Spring Boot, JUnit
- Biblioteka Spring Web, jQuery, Mockito

### Sposób uruchomienia projektu:

Aby uruchomić projekt, niezbędne będą narzędzia takie jak Apache Maven, Apache Tomcat.

Konfiguracja Tomcat :

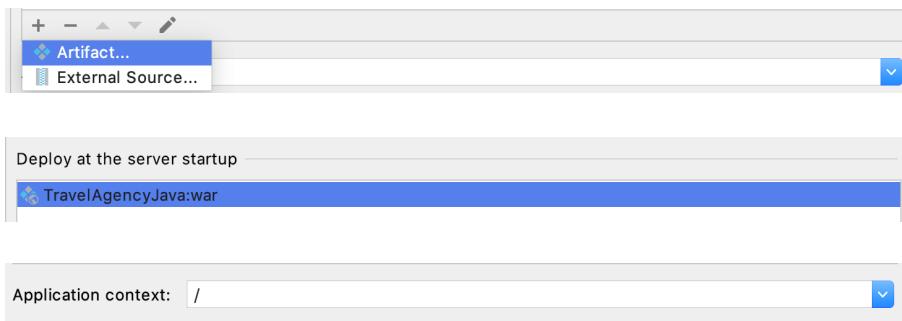
Klikamy Edit Configurations...

Dodajemy nową konfigurację Tomcat Server -> Local

Wybieramy ścieżkę do folderu Tomcat

Po odpowiedniej konfiguracji uzyskamy tą opcję do uruchomienia aplikacji

## Konfiguracja Artifact :



W zakładce Deployment  
Klikamy + i dodajemy  
Artifact...

Wybieramy artifact  
„TravelAgencyJava:war”

W ścieżce pozostawiamy  
znak /

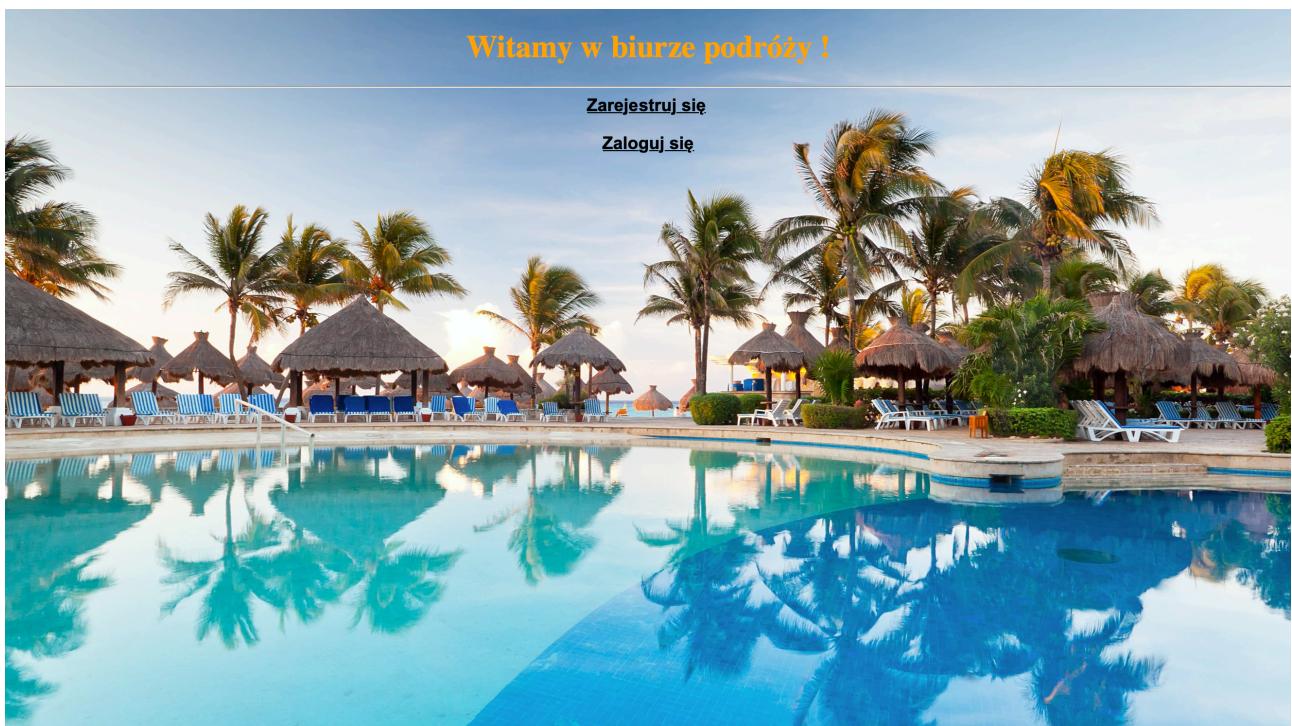
Zatwierdzamy i klikamy Ok.

1. Zmiana ścieżki do bazy danych znajdującej się w klasie SQLite

```
conn = DriverManager.getConnection( url: "jdbc:sqlite:/Users/konradmichalski/Desktop/TravelAgency-master/database.db");
```

2. Projekt uruchamiamy na serwerze Tomcat, po wcześniejszej konfiguracji oraz po dodaniu pliku Artifact.
3. Obsługa projektu wyjaśniona w wyżej wymienionych krokach, graficznie poniżej oraz w pliku video.

Okno po uruchomieniu aplikacji, do wyboru opcja rejestracji i loginu



Rejestracja nowego konta, odrzuca jeśli powtarza się imię lub hasła są niezgodne ze sobą

## Rejestracja

Imię:  Nazwisko:  Hasło:  Potwierdź hasło:

Login, dane sprawdzane w bazie, jeżeli użytkownik nie istnieje brak możliwości przejścia dalej

## Login

Imię:

Hasło:

Główna strona zawierająca informacje dotyczące miejsca, zdjęcia, terminy, koszt wycieczek

### Hiszpania

[Rezerwuj](#) >Informacje< >Termin< Barcelona - 2900 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Madryt - 3200 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Majorka - 4100 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Sewilla - 2050 zł/os.

Kapadocja to dzieło wciąż nieskófczone - wiatr i deszcze nieustannie zmieniają kształt vulkanicznych skał. Najbardziej urozmaicone krajobrazy z fantastycznymi formami skalnymi znajdują się w trójkaście wyznaczonym przez miasta Avanos, Ürgüp i Nevşehir. Turcja Kapadocja. Kapadocja to także krajna balonów.

[Zamawiam!](#)

**Rachunek:**

[Rezerwuj](#) >Informacje< >Termin< Rzym - 3500 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Wenecja - 4200 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Mediolan - 2550 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Florencja - 3300 zł/os.

[Zamawiam!](#)

### Włochy



### Turcja



### Portugalia



Powyżej widzimy opis jednego z miejsc oraz termin wycieczki, ilość dni i koszt, po prawej stronie kilka zarezerwowanych losowych wycieczek oraz koszt całego rachunku, zamawiamy

### Hiszpania

[Rezerwuj](#) >Informacje< >Termin< Barcelona - 2900 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Madryt - 3200 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Majorka - 4100 zł/os.

[Rezerwuj](#) >Informacje< >Termin< Sewilla - 2050 zł/os.

Miasto: Kapadocja  
Liczba dni: 8  
Cena za osobę: 3100  
Data wylotu: 10/06/2020  
Data przylotu: 18/06/2020

[Zamawiam!](#)

**Rachunek:**

Kapadocja : 3100 zł  
Kapadocja : 3100 zł  
Kapadocja : 3100 zł  
Kapadocja : 3100 zł  
Majorka : 4100 zł  
Majorka : 4100 zł  
Majorka : 4100 zł  
Porto : 3400 zł  
Porto : 3400 zł

Całkowity koszt: 31500zł

### Włochy



### Turcja

Lot	Odstęp	Przylot
LOT LO-3860 economy (V) Sekcja lotu Casa Iwants lotu 00:55	25.10.2015 21:05 Wrocław (PL) Strachowice	25.10.2015 22:00 Warszawa (PL) F. Chopin
Lata dobrej dnia 6-400		

### Portugalia

Lot	Odstęp	Przylot
LOT LO-3827 economy (V) Sekcja lotu Casa Iwants lotu 00:55	25.10.2015 22:40 Warszawa (PL) F. Chopin	25.10.2015 23:35 Gdańsk (PL) Rebiechowo
Lata dobrej dnia 6-400		



Dla przykładu wybraliśmy więcej wycieczek, które w tym miejscu usuniemy zostawiając jedynie 4 bilety do Kapadocji

### Zarezerwowane wycieczki:

Kapadocja- 3100 zł	X
Majorka - 4100 zł	X
Majorka - 4100 zł	X
Majorka - 4100 zł	X
Porto - 3400 zł	X
Porto - 3400 zł	X

Podaj dane do faktury :

Email:

Telefon:

[Rezerwuj](#)

Podajemy dane kontaktowe i rezerwujemy

### Zarezerwowane wycieczki:

Kapadocja- 3100 zł	X

Podaj dane do faktury :

Email:  konrad@gmail.com

Telefon:  668302391

[Rezerwuj](#)

Faktura zawiera dane użytkownika, ostateczny koszt zamówienia oraz dane zamówienia

### Dane faktury

Imię : Konrad
Nazwisko : Michalski
Email : konrad@gmail.com
Numer : 668302391
Calkowity koszt : 12400 zł
Data złożenia zamówienia : 12.06.20, 19:51

### Bilety :

Kraj : Turcja
Miasto : Kapadocja
Cena : 3100 zł/1 os.
Liczba dni : 8
Data odlotu : 10/06/2020
Data przylotu : 18/06/2020

Kraj : Turcja
Miasto : Kapadocja
Cena : 3100 zł/1 os.
Liczba dni : 8
Data odlotu : 10/06/2020
Data przylotu : 18/06/2020

Kraj : Turcja
---------------

Na stronie zawarte są również bilety kupionych wycieczek

Cena : 3100 zł/1 os.
Liczba dni : 8
Data odlotu : 10/06/2020
Data przylotu : 18/06/2020

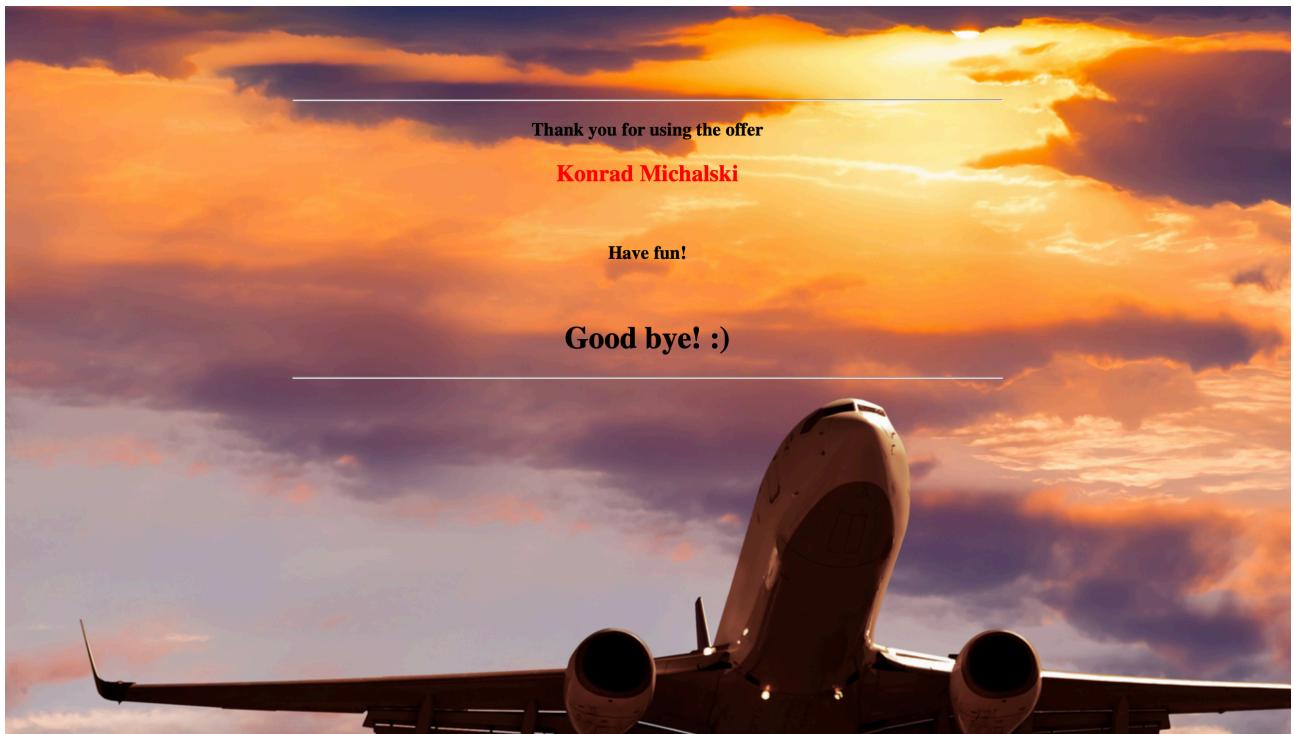
Kraj : Turcja
Miasto : Kapadocja
Cena : 3100 zł/1 os.
Liczba dni : 8
Data odlotu : 10/06/2020
Data przylotu : 18/06/2020

Kraj : Turcja
Miasto : Kapadocja
Cena : 3100 zł/1 os.
Liczba dni : 8
Data odlotu : 10/06/2020
Data przylotu : 18/06/2020

Kraj : Turcja
Miasto : Kapadocja
Cena : 3100 zł/1 os.
Liczba dni : 8
Data odlotu : 10/06/2020
Data przylotu : 18/06/2020

[Dalej](#)

Po przejściu dalej widzimy podziękowania dla zalogowanego użytkownika i kilka miłych słów



Dane w bazie zgadzają się ze złożonym zamówieniem, analizujemy ostatni wiersz

<b>id_order</b>	<b>id_user</b>	<b>email</b>	<b>number</b>	<b>total</b>	<b>date</b>
Filtr	Filtr	Filtr	Filtr	Filtr	Filtr
314	25	adam@gmail.com	684928123	5800	06.06.20, 14:21
315	26	kamil@gmail.com	929201023	5800	07.06.20, 20:50
316	23	konrad@gmail.c...	668302391	12400	12.06.20, 19:34
317	25	konmic98@gma...	123	8200	12.06.20, 19:46
318	23	konrad@gmail.c...	668302391	12400	12.06.20, 19:51

Jak zauważyliśmy id zamówienia ostatniego wiersza wyniosło 318, w poniższej tabeli widzimy 4 zamówienia pod id 318, które odpowiadają wycieczce do Kapadocji w powyższym zamówieniu

<b>id_reservation</b>	<b>id_tours</b>	<b>id_orders</b>
Filtr	Filtr	Filtr
<b>588</b>	<b>11</b>	<b>314</b>
<b>590</b>	<b>1</b>	<b>315</b>
<b>591</b>	<b>1</b>	<b>315</b>
<b>592</b>	<b>10</b>	<b>316</b>
<b>593</b>	<b>10</b>	<b>316</b>
<b>594</b>	<b>10</b>	<b>316</b>
<b>595</b>	<b>10</b>	<b>316</b>
<b>598</b>	<b>3</b>	<b>317</b>
<b>599</b>	<b>3</b>	<b>317</b>
<b>601</b>	<b>10</b>	<b>318</b>
<b>602</b>	<b>10</b>	<b>318</b>
<b>603</b>	<b>10</b>	<b>318</b>
<b>604</b>	<b>10</b>	<b>318</b>

Id wycieczki realizowanego zamówienia to 10, w tabeli pod tym id kryje się Kapadocja

<b>id_tour</b>	<b>country</b>	<b>city</b>	<b>days</b>	<b>price</b>	<b>departure</b>	<b>arrival</b>
Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr
<b>1</b>	<b>Hiszpania</b>	<b>Barcelona</b>	<b>7</b>	<b>2900</b>	<b>01/06/2020</b>	<b>08/06/2020</b>
<b>2</b>	<b>Hiszpania</b>	<b>Madryt</b>	<b>10</b>	<b>3200</b>	<b>02/07/2020</b>	<b>12/07/2020</b>
<b>3</b>	<b>Hiszpania</b>	<b>Majorka</b>	<b>7</b>	<b>4100</b>	<b>05/06/2020</b>	<b>12/06/2020</b>
<b>4</b>	<b>Hiszpania</b>	<b>Sewilla</b>	<b>4</b>	<b>2050</b>	<b>20/07/2020</b>	<b>24/07/2020</b>
<b>5</b>	<b>Włochy</b>	<b>Rzym</b>	<b>7</b>	<b>3500</b>	<b>11/06/2020</b>	<b>18/06/2020</b>
<b>6</b>	<b>Włochy</b>	<b>Wenecja</b>	<b>8</b>	<b>4200</b>	<b>18/06/2020</b>	<b>25/06/2020</b>
<b>7</b>	<b>Włochy</b>	<b>Mediolan</b>	<b>4</b>	<b>2550</b>	<b>19/06/2020</b>	<b>23/06/2020</b>
<b>8</b>	<b>Włochy</b>	<b>Florencja</b>	<b>6</b>	<b>3300</b>	<b>10/06/2020</b>	<b>16/06/2020</b>
<b>9</b>	<b>Turcja</b>	<b>Stambuł</b>	<b>10</b>	<b>3500</b>	<b>21/06/2020</b>	<b>01/07/2020</b>
<b>10</b>	<b>Turcja</b>	<b>Kapadocja</b>	<b>8</b>	<b>3100</b>	<b>10/06/2020</b>	<b>18/06/2020</b>
<b>11</b>	<b>Turcja</b>	<b>Pamukkale</b>	<b>5</b>	<b>2900</b>	<b>22/07/2020</b>	<b>27/07/2020</b>
<b>12</b>	<b>Turcja</b>	<b>Antalya</b>	<b>7</b>	<b>3050</b>	<b>12/07/2020</b>	<b>19/07/2020</b>
<b>13</b>	<b>Portugalia</b>	<b>Lizbona</b>	<b>8</b>	<b>3800</b>	<b>10/06/2020</b>	<b>18/06/2020</b>

Testy jednostkowe z wykorzystaniem frameworka JUnit, wraz z biblioteką Mockito:

-LoginTest, MenuTest, RegistrationTest, SQLiteTest, TestStart

Opis podstawowych funkcjonalności klas i metod:

**Klasa: User** ( zawiera pola dotyczące użytkownika )

**Klasa: Tour** ( zawiera pola dotyczące wycieczek )

**Klasa: Order** ( zawiera pola dotyczące zamówień )

**Klasa: RemovedTours** ( zawiera pola dotyczące usuniętych wycieczek )

**Klasa: Login** ( zawiera obsługę logowania użytkownika )

Metody:

doGetLogin() - wykonuje GET do przekazania adresu strony

doPostLogin() - wykonuje POST do przesłania danych, jeśli się powiedzie ustawiana flaga IsLogged

**Klasa: Registration** ( zawiera obsługę rejestracji użytkownika )

Metody:

doGetRegistration() - wykonuje GET do przekazania adresu strony, jeśli dane poprawne przenosi na kolejną stronę

doPostRegistration() - wykonuje POST do przesłania danych, jeśli nieprawidłowe dane następuje powrót do rejestracji

**Klasa: Menu** ( zawiera obsługę menu )

Metody:

doGetMenu() - wykonuje GET do przekazania adresu strony, przekazuje podstawowe dane dotyczące użytkownika, wycieczek

doPostMenu() - wykonuje POST do przesłania danych, dodaje wybrane wycieczki przez użytkownika do tabeli aktualnego zamówienia

**Klasa: ReservationTour** ( zawiera obsługę początkowo wybranego zamówienia )

Metody:

doGetReservation() - wykonuje GET do przekazania adresu strony, przekazuje dane użytkownika i wybrane wycieczki dla aktualnego zamówienia

doPostReservation() - wykonuje POST do przesłania danych, usuwa wycieczki z których zrezygnował użytkownik

**Klasa: Invoice** ( zawiera obsługę ostatecznego zamówienia )

Metody:

doGetInvoice() - wykonuje GET do przekazania adresu strony, przekazuje wszystkie dane dotyczące użytkownika, aktualnego zamówienia, wybranych wycieczek

**Klasa: Exit** ( zawiera obsługę ostatniej strony )

Metody:

doGetExit() - wykonuje GET do przekazania adresu strony, przekazuje aktualnie zalogowanego użytkownika

**Klasa: SQLite** ( służy do połączenia z bazą danych )

Metody:

DbConnect - odpowiada za połączenie z bazą danych

DbDisconnect - odpowiada za odłączenie się od bazy danych

**Klasa: SQLiteDB** ( zawiera wszystkie funkcjonalności, dotyczące bazy )

Metody:

AddNewUser(User user) - Metoda przyjmuje obiekt klasy User i wykonuje zapytanie do bazy w celu stworzenia nowego użytkownika

CheckIfUserExist(User user) - Metoda przyjmuje obiekt klasy User i sprawdza czy przekazany użytkownik istnieje w bazie

CheckLogin(User user) - Metoda przyjmuje obiekt klasy User i sprawdza czy podane dane przy logowaniu zgadzają się z tymi w bazie danych

GetAllTours() - Metoda zwraca wszystkie wycieczki znajdujące się w bazie

NewOrder() - Metoda dodająca zamówienie do bazy danych dla zalogowanego użytkownika

FindTourByName(String city) - Metoda przyjmuje nazwę miasta i sprawdza czy istnieje w bazie

AddOrderTour(int tourId) - Metoda przyjmuje id wycieczki i wykonuje zapytanie do bazy w celu dodania nowego wiersza Rezerwacji dla podanego id

GetReservationTours() - Metoda zwracająca dane dotyczące aktualnego zamówienia, wykorzystująca do tego zmienną statyczną id zamówienia

AddData(RemovedTours data) - Metoda przyjmuje obiekt klasy RemovedTours i aktualizuje dane dotyczące aktualnie zalogowanego użytkownika, sumuje cenę wszystkich jego zamówień oraz aktualizuje cenę tego zamówienia w tabeli Orders

RemoveTourFromReservation(int TourId) - Metoda przyjmuje id wycieczki i wykonuje zapytanie do bazy usuwając wycieczki z których zrezygnował użytkownik w czasie składania zamówienia

CheckSelectedTours() - Metoda zwracająca wszystkie dane wycieczek dla podanego numeru id\_orders w tabeli Reservations

GetUserNameAndSurname() - Metoda zwracająca imię i nazwisko użytkownika

GetOrdersData() - Metoda zwracająca dane dotyczące aktualnego zamówienia