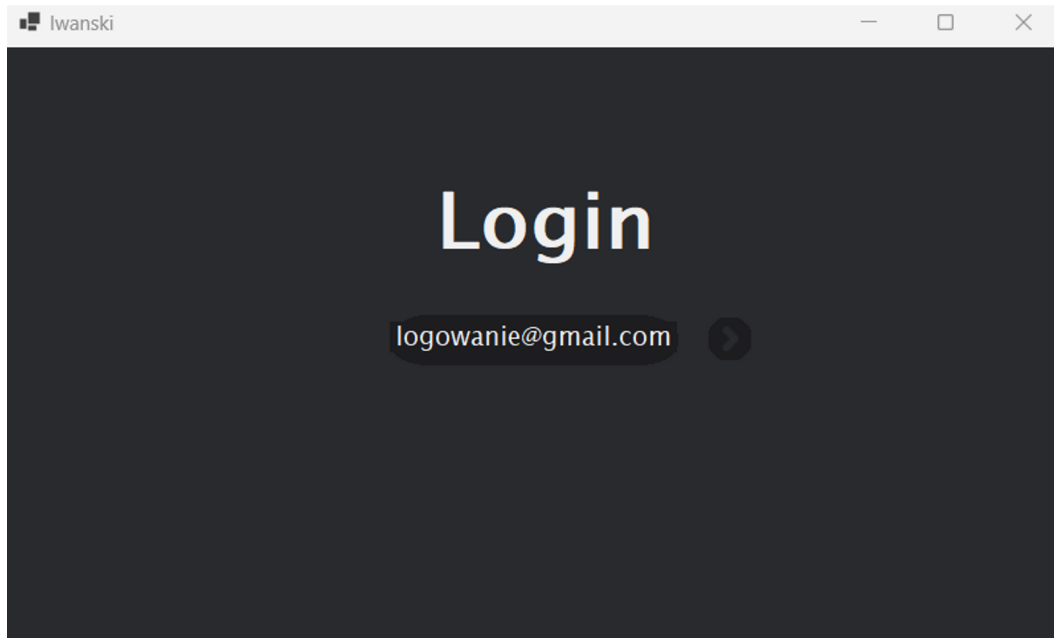


Przebieg autoryzacji/tworzenie użytkownika

Wygląd formularza logowania:



Pobranie e-maila z pola tekstowego i sprawdzenie czy jest on prawidłowy:

```
// Pobieranie emaila z pole tekstowego var email =  
txtEmail.Text; var emailValidator = new EmailAddressAttribute();  
// Sprawdzanie emaila if (!emailValidator.IsValid(email)) {  
lblError.Text = "Nieprawidłowy adres email!"; return; }
```

funkcja sprawdzająca poprawność maila:

```
public override bool IsValid(object? value) { if (value == null)  
{ return true; } if (!(value is string valueAsString)) { return  
false; } if (!EnableFullDomainLiterals &&  
(valueAsString.Contains('\r') || valueAsString.Contains('\n'))  
{ return false; } // only return true if there is only 1 '@'  
character // and it is neither the first nor the last character  
int index = valueAsString.IndexOf('@'); return index > 0 &&
```

```
index != valueAsString.Length - 1 && index ==  
valueAsString.LastIndexOf('@'); }
```

Następnie następuje sprawdzenie czy podany adres e-mail istnieje już w bazie danych. Jeśli nie to jest tworzony nowy wpis do bazy. Sprawdzone jest również to czy nie jest to przypadkiem adres admina:

```
// Pobieranie użytkownika po emailu z bazy danych var user =  
await _userService.GetUserByEmailAsync(email); // Jeśli  
użytkownik jest pusty to utwórz nowego z jakimiś defaultowymi  
polami if (user == null) { // IsAdmin jest bardzo ważny bez  
niego nie można wejść do panelu admina user = new User { Email =  
email, IsAdmin = false }; await  
_userService.CreateUserAsync(user); }
```

Po sprawdzeniu generowany jest token, który zostaje przypisany do maila w bazie danych i wysłany na adres użytkownika. Po wpisaniu tokenu w pole tekstowe sprawdzana jest jego zgodność z tym w bazie danych, co przy potwierdzeniu zgodności kończy proces autoryzacji

```
var enteredToken = txtToken.Text; // Sprawdź czy token jest  
prawidłowy if (_user.Token == enteredToken) { // Utwórz  
MainForm var mainForm = new MainForm(_user, _eventService);  
mainForm.Show(); this.Close(); } else { lblError.Text = "Błędny  
Token!"; }
```

