Programowanie dynamicznych serwisów internetowych ASP .NET, Grupa N22-32,

Jakub Świercz – 277609

System Rezerwacji Sal

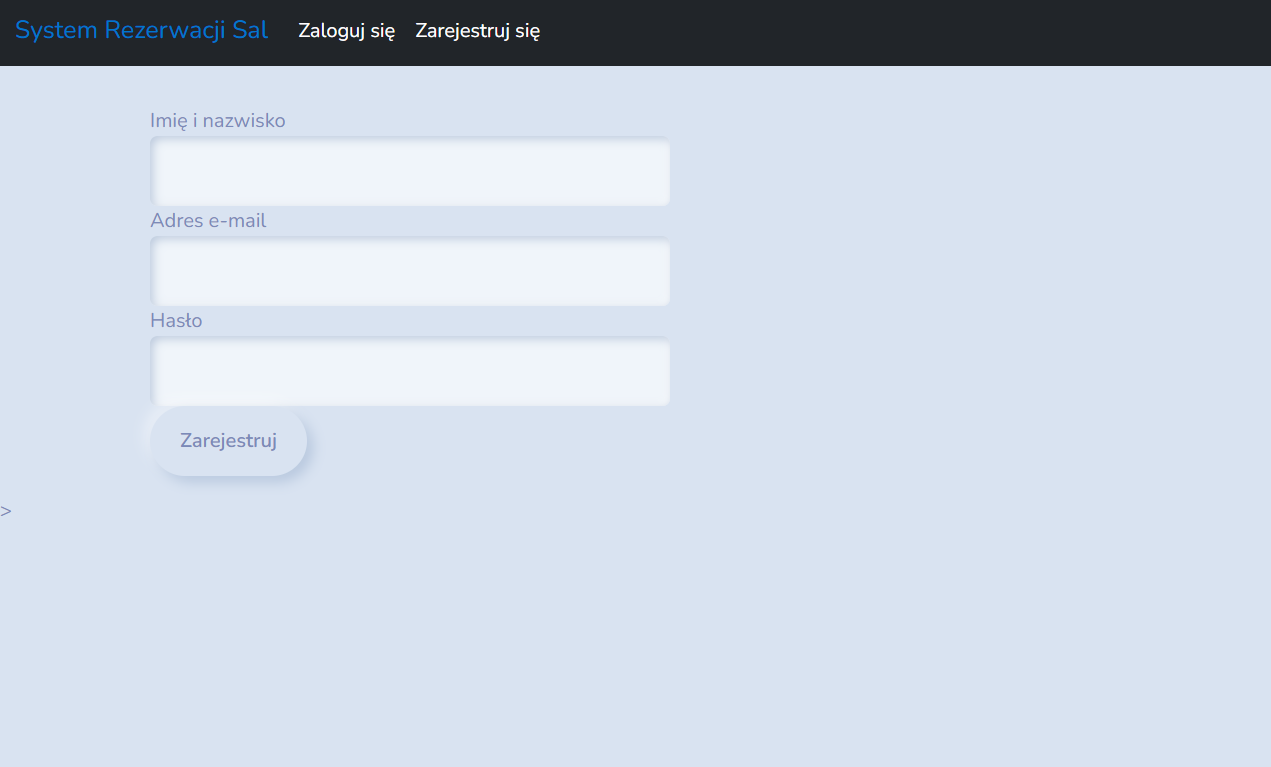
Cel:

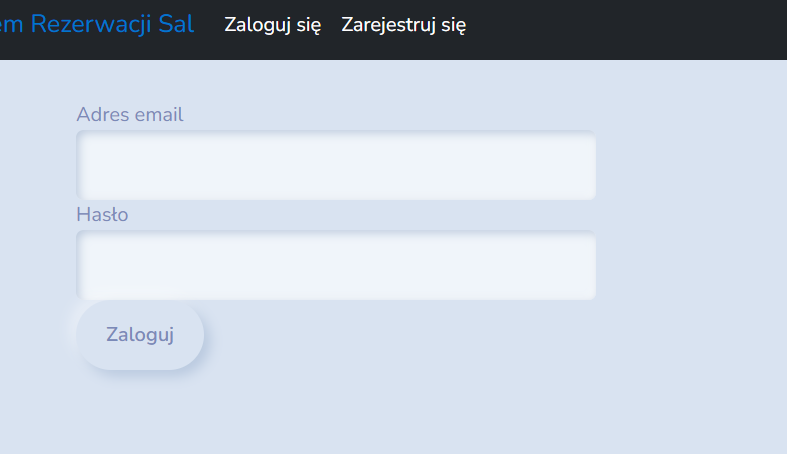
Celem projektu jest stworzenie serwisu internetowego, który umożliwi użytkownikom w przystępny sposób zarezerwować salę, która spełnia wymagane przez użytkownika kryteria. Interfejs serwisu powinien być intuicyjny i prosty w obsłudze.

Założenia projektu:

Projekt został wykonany wykorzystując ASP.NET 6.0 MVC – platformę aplikacyjną do budowy aplikacji internetowych opartych na wzorcu Model-View-Controller. Głównym założeniem projektu było wykorzystanie możliwości platformy jak i wzorca MVC do zbudowania sprawnie działającego serwisu, który może służyć jako system do rezerwacji sal.

Funkcjonalność:

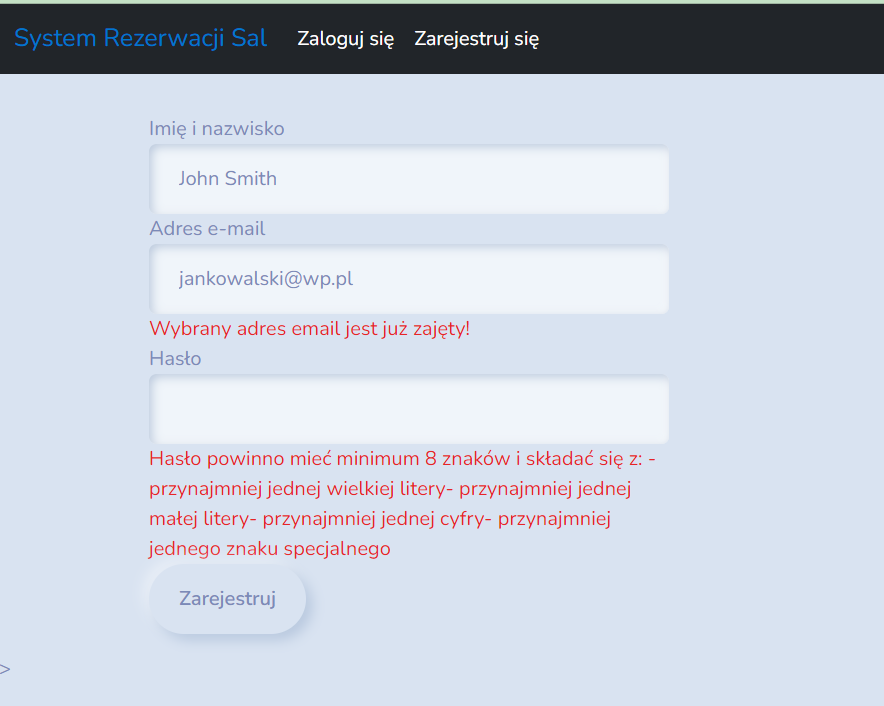
Serwis posiada oczywiście możliwość zakładania konta. Poniżej podstawowe widoki logowania i rejestracji:



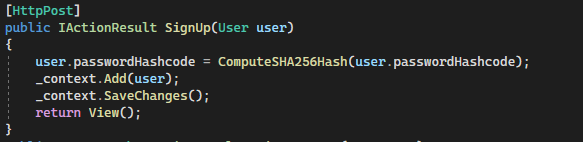
Zaimplementowano podstawową obsługę błędów:

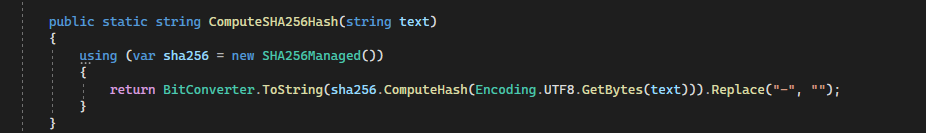
Obraz zawierający tekst

Opis wygenerowany automatycznie

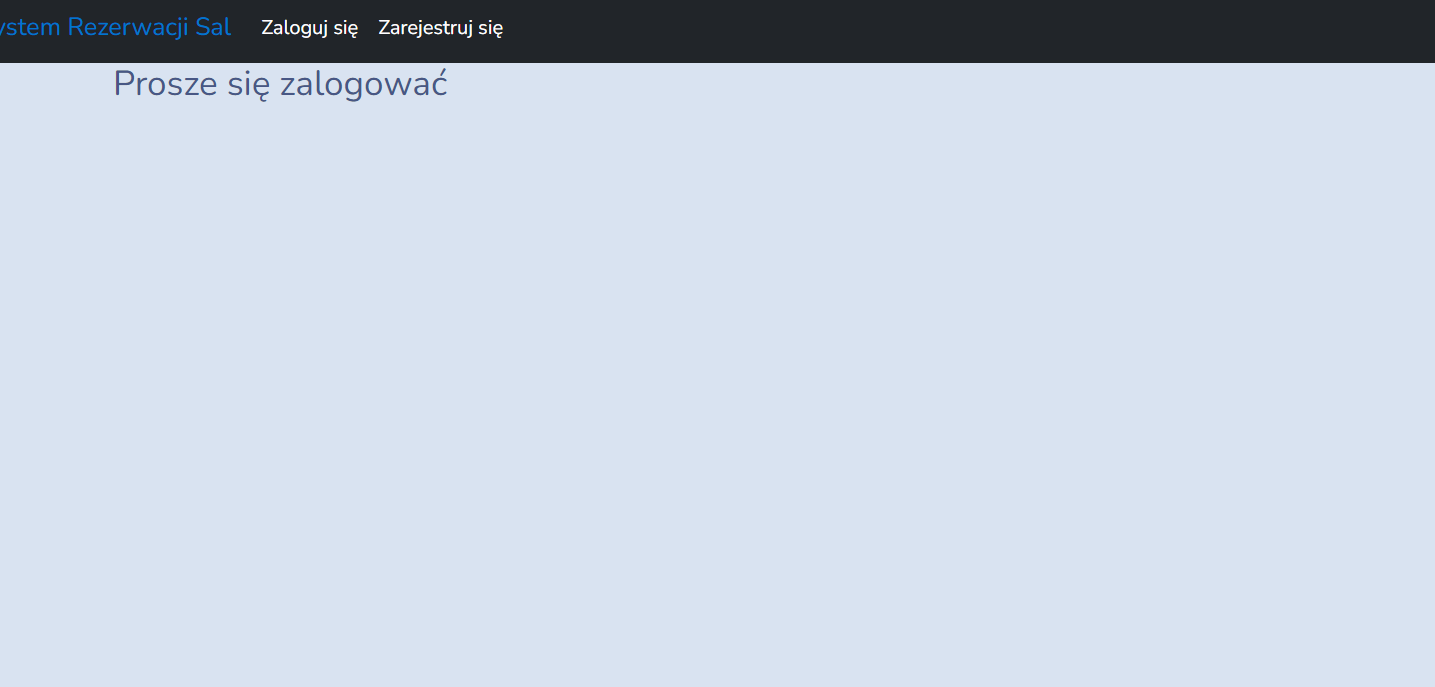


Serwis nie trzyma haseł użytkowników w bazie. Hasła są hashowane, i ta wartość jest zapisywana do bazy danych. Poniżej działanie akcji w kontrolerze odpowiedzialnej za zapisanie zarejestrowanego użytkownika w bazie:



Oraz wykorzystana metoda:

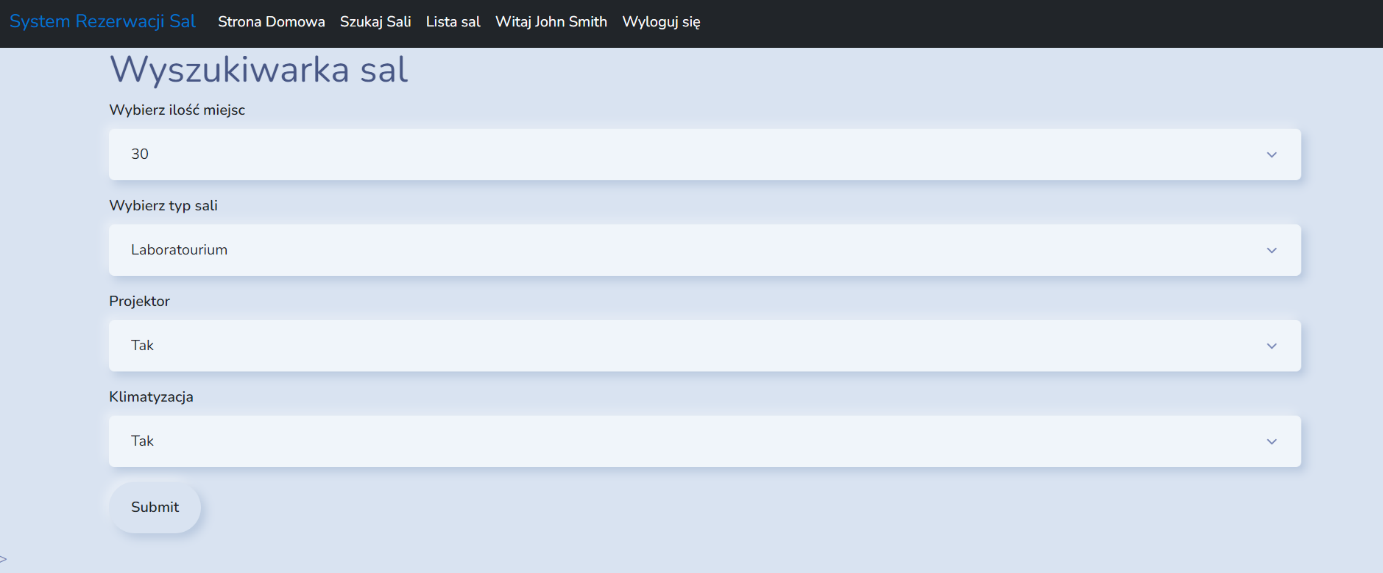
Strona domowa serwisu wyświetla odpowiednie informacje na podstawie tego czy użytkownik jest zalogowany czy nie:



Jeśli użytkownik się zalogował, na stronie głównej zobaczy wszystkie rezerwacje, których dokonał. Na pasku również zostaną udostępnione wszystkie funkcjonalności serwisu.



Strona posiada prostą wyszukiwarkę sal, gdzie użytkownik może wybrać odpowiednie kryteria i wyszukać salę według wybranych kryteriów:



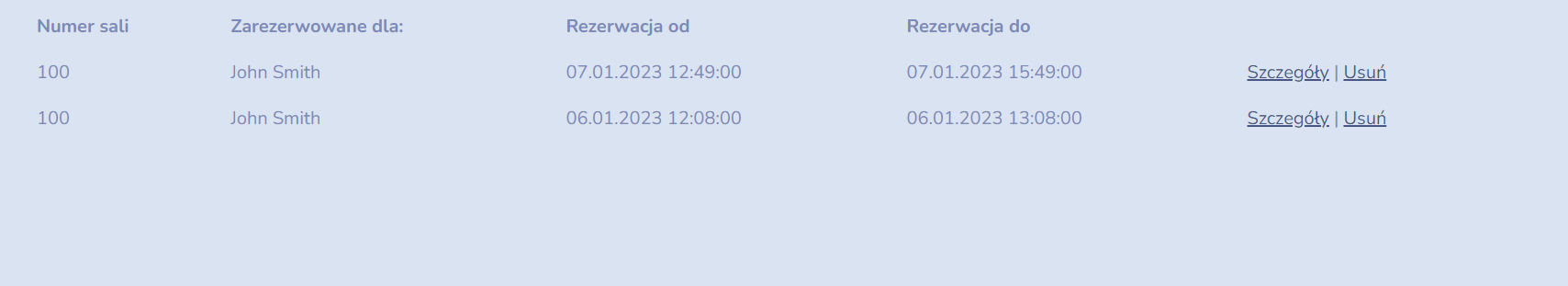
Użytkownik może również otworzyć listę wszystkich dostępnych sal i z tego poziomu dokonać rezerwacji lub sprawdzić rezerwacje na daną salę:

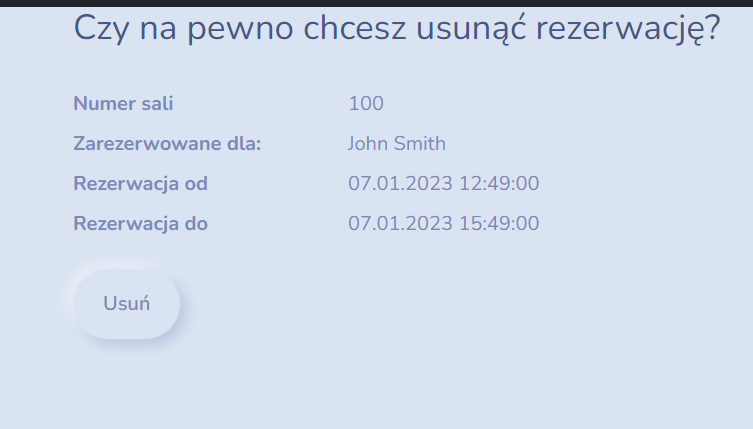
Obraz zawierający stół

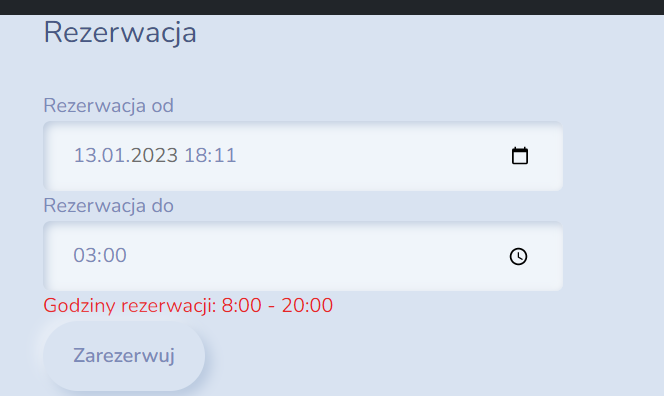
Opis wygenerowany automatycznie

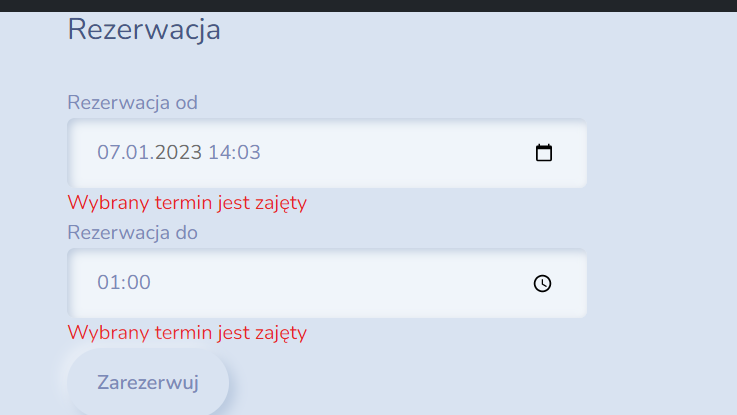


Użytkownik może usunąć rezerwację:



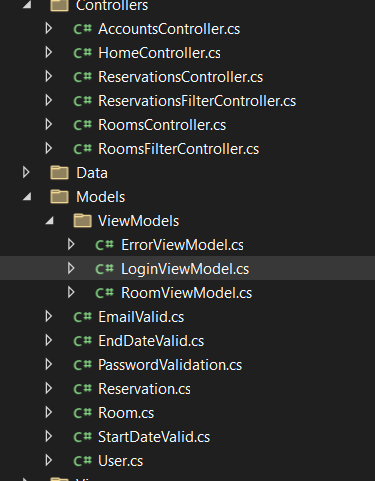


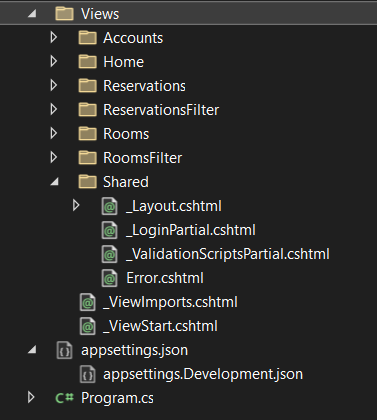
Tutaj również zaimplementowano podstawową obsługę błędów:



Projekt został przygotowany w oparciu o dokumentacje i wykorzystanie wzorca Model-View-Controller.

Każda rezerwacja, użytkownik, sala itd. posiada odpowiednie widoki, modele oraz kontrolery.

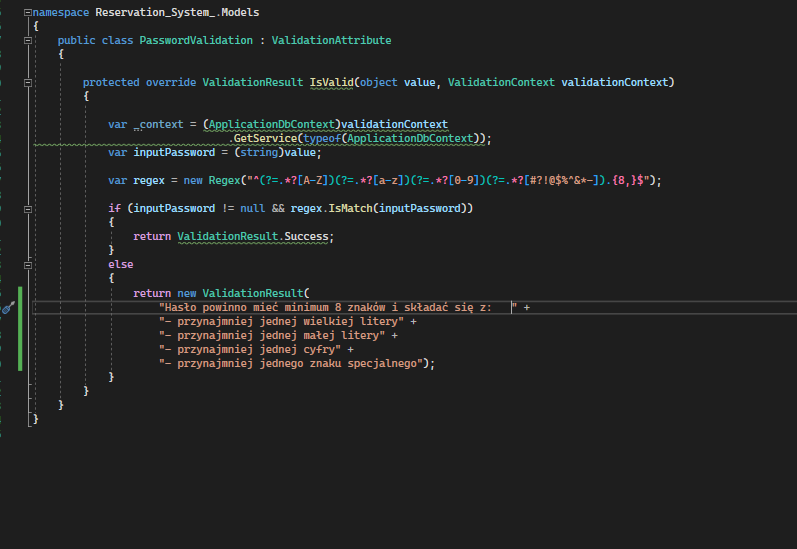
Poniżej została przedstawiona struktura projektu:



W projekcie najpierw zostały napisane odpowiednie modele i później kontrolery. Następnie nawiązano połączenie z bazą danych i utworzono podstawowe widoki.

Wraz z rozwojem projektu implementowano różne zmiany, np. edycja istniejących widoków, kontrolerów, kreacja nowych akcji i kontrolerów dla nowych modeli. Na przykład w trakcie rozwoju dopisano własne walidacje dla niektórych typów danych z modeli.

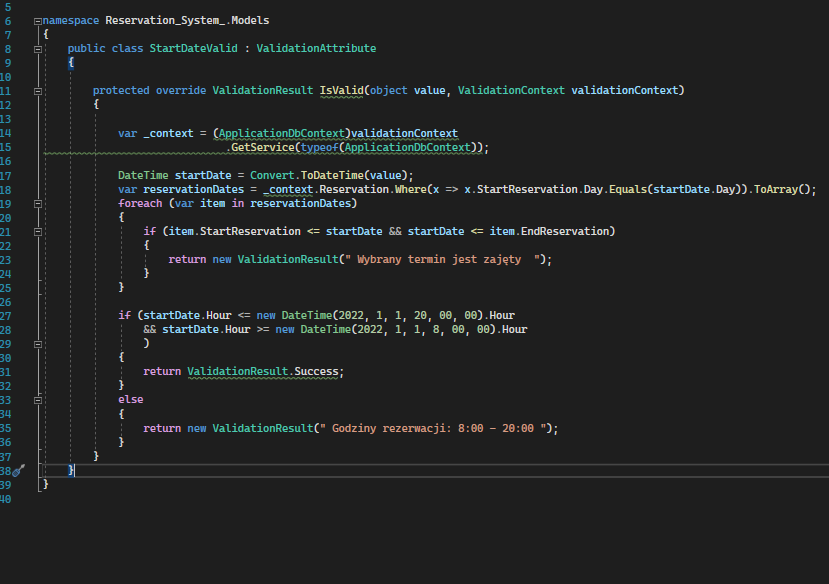
Przykład:



Powyższa walidacja wykorzystuje odpowiednie wyrażenie regularne, które sprawdza czy hasło spełnia podstawowe wymagania bezpieczeństwa, jeśli nie to zwracana jest odpowiednia wiadomość dla użytkownika.

Poniższa walidacja jest stworzona dla wprowadzanej daty i godziny startu rezerwacji.

Zaimplementowano odpowiednie metody, które sprawdzają czy termin nie jest już zajęty i czy podana data mieści się w dozwolonych godzinach rezerwacji. Jeśli wprowadzona data nie spełnia wymagań jest zwracana odpowiednia wiadomość. Odpowiednia walidacja jest również zastosowana do daty końca rezerwacji.



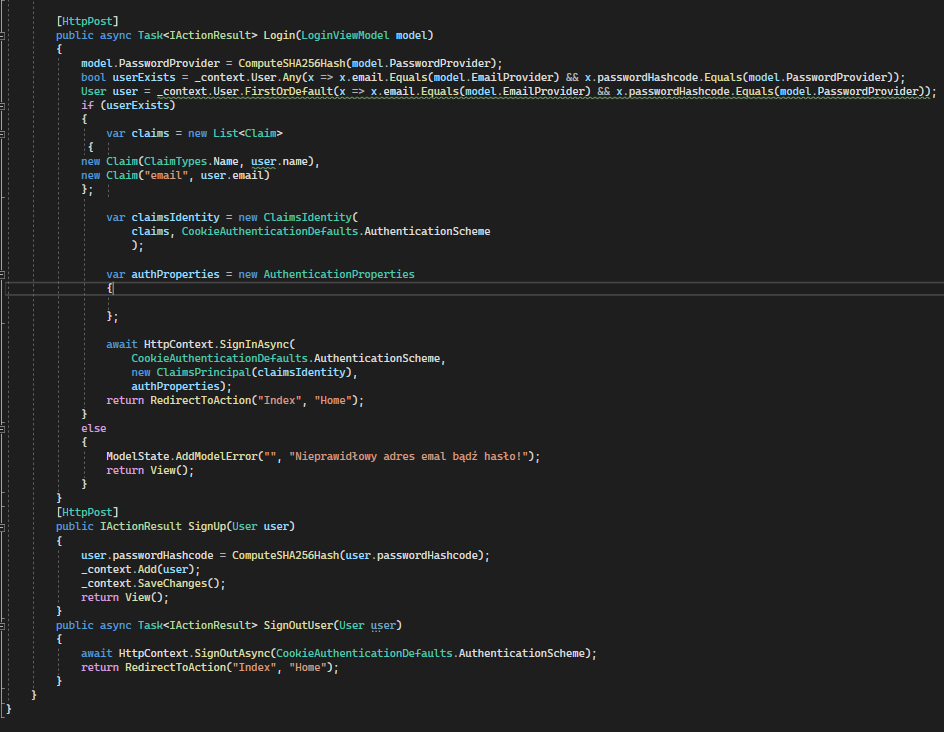
Poniżej znajdują się implementacje ważniejszych komponentów projektu:

Poniżej znajduje się fragment kontrolera odpowiedzialnego za logowanie i rejestracje.

W akcji *Login* pobierane są dane dotyczące wprowadzonego hasła, metoda sprawdza czy użytkownik istnieje w bazie i jeśli tak to jest logowany do systemu. Jeśli nie to zwracany jest odpowiedni błąd używając:

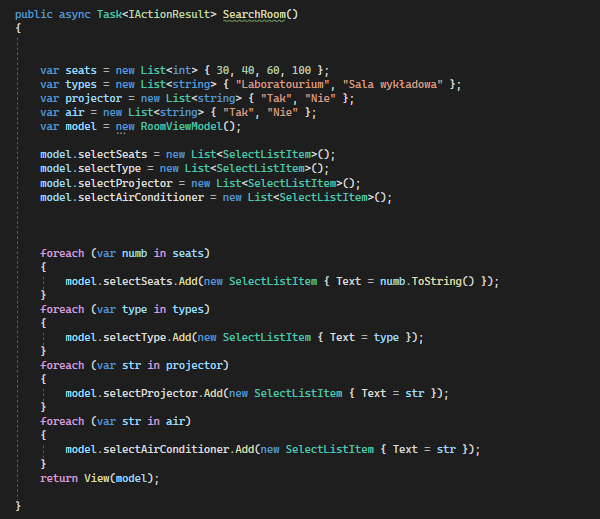
ModelState.AddModelError("", "Nieprawidłowy adres emal bądź hasło!");

return View();



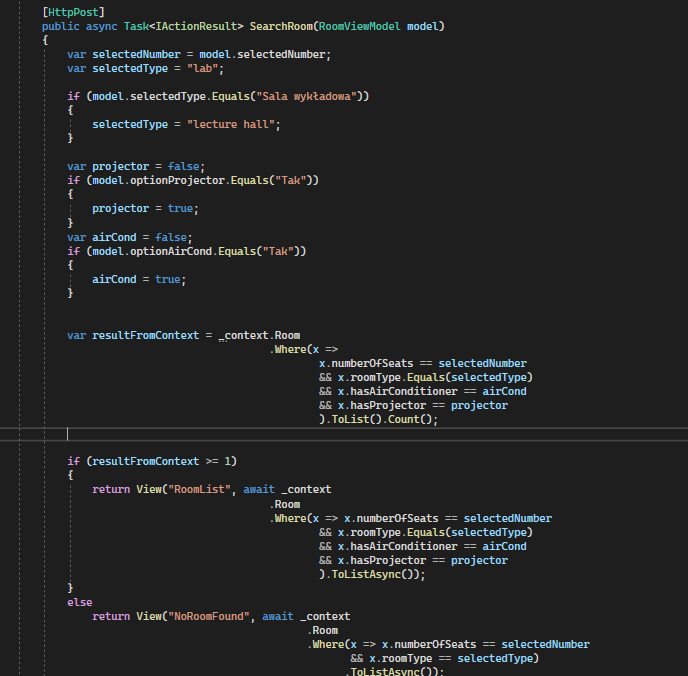
W akcji SignUp można zaobserwować, że hasło jest hashowane i potem zapisywane do bazy. Dla każdej akcji zwracany jest odpowiedni widok.

Poniżej przedstawiono fragment kontrolera odpowiedzialnego za sale:



Fragment przedstawia akcję, która jest wykorzystywana w implementacji wyszukiwarki sal.

Powyższa akcja „ładuje” dane dla odpowiedniego modelu, który jest wczytywany do widoku, który został zaprezentowany wcześniej.



Powyższa akcja zbiera wybrane przez użytkownika kryteria zawarte w modelu RoomViewModel i zwraca przefiltrowane wyniki w odpowiednim widoku.