

2. PROJEKT SYSTEMU

2.1. Wstęp

W ramach projektu systemu przedstawiono schemat bazy danych ogrodu botanicznego wraz z przechowywanymi informacjami, a także projekt interfejsu graficznego. Prawidłowe wykonanie projektu znacząco ułatwia implementację rozwiązania oraz pozwala na uniknięcie pomyłek w trakcie realizacji projektu stanowiąc podstawę do tworzenia systemu internetowego i jego interfejsu.

2.2. Decyzje projektowe z zakresu wykorzystania technologii

Realizacja systemu internetowego wymaga integracji z serwerem oraz posiadania bazy danych, dlatego system zostanie wykonany z użyciem środowiska o nazwie Meteor.js. Framework ten umożliwia utworzenie nowoczesnego systemu webowego w stylu **one page site**, w którego skład wejdą: aplikacja kliencka w postaci strony web, strona serwerowa oraz baza danych co zostało opisane w analizie projektowej.

Niektóre funkcjonalności systemu zrealizowane zostaną przy pomocy paczek/bibliotek, których listę przedstawiono poniżej:

- accounts-password (<https://atmospherejs.com/meteor/accounts-password>) – bezpieczne logowanie przy pomocy hasła,
- session (<https://atmospherejs.com/meteor/session>) - reaktywne zmienne sesji,
- reactive-dict (<https://atmospherejs.com/meteor/reactive-dict>) - reaktywne słowniki zmiennych,
- check (<https://atmospherejs.com/meteor/check>) - sprawdzanie czy wartość spełnia wzorec,
- http (<https://atmospherejs.com/meteor/http>) – zapytania HTTP to innych serwerów,
- meteortools:allthings (<https://atmospherejs.com/meteortools/allthings>) – zestaw narzędzi deweloperskich,
- natestrausser:publish-performant-counts (<https://atmospherejs.com/natestrausser/publish-performant-counts>) – efektywne zliczanie liczebności dokumentów w kolekcji,
- meteorhacks:ssr (<https://atmospherejs.com/meteorhacks/ssr>) – renderowanie po stronie serwera,
- alanning:roles (<https://atmospherejs.com/alanning/roles>) – realizacja uprawnień użytkowników,
- perak:joins (<https://atmospherejs.com/perak/joins>) – możliwość dołączania do dokumentu, dokumentów z innych kolekcji,
- kadira:blaze-layout (<https://atmospherejs.com/kadira/blaze-layout>) – menadżer layout'ów,
- kadira:flow-router (<https://atmospherejs.com/kadira/flow-router>) – routing po stronie klienta,
- nimble:restivus (<https://atmospherejs.com/nimble/restivus>) – tworzenie REST API,
- aldeed:geocoder (<https://atmospherejs.com/aldeed/geocoder>) – geokodowanie,
- momentjs:moment (<https://atmospherejs.com/momentjs/moment>) – manipulacja datami,
- rzymek:moment-locale-pl (<https://atmospherejs.com/rzymek/moment-locales>) – polska lokalizacja paczki moment.js,
- fourseven:scss (<https://atmospherejs.com/fourseven/scss>) – wsparcie dla Sass i SCSS,

- less (<https://atmospherejs.com/meteor/less>) – wsparcie dla LESS,
- semantic:ui (<https://atmospherejs.com/semantic/ui>) – frontend framework,
- natestrauser/animate-css (<https://atmospherejs.com/natestrauser/animate-css>) – animacje,
- dropzone.js (<https://www.npmjs.com/package/dropzone>) – panel umożliwiający przeciąganie i upuszczanie plików i ich wysyłanie,
- clipboard.js (<https://www.npmjs.com/package/clipboard>) - kopiowanie tekstu do schowka,
- phantom.js (<https://www.npmjs.com/package/phantom>) – m.in. dostęp do stron internetowych bez przeglądarki,
- spin.js (<https://www.npmjs.com/package/spin>) - animowany indykator ładowania strony,
- webshot (<https://www.npmjs.com/package/webshot>) - tworzenie zrzutów stron internetowych bez przeglądarki,
- winston (<https://www.npmjs.com/package/winston>) – asynchroniczne przesyłanie logów.

W celu wykonania funkcjonalnej i responsywnej strony internetowej zostanie wykorzystany framework o nazwie Semantic UI w wersji 2.2.7, dzięki czemu system będzie poprawnie wyświetlany zarówno na komputerach, jak i urządzeniach mobilnych oraz będzie wykorzystywał gotowe elementy interfejsu graficznego w celu szybszego procesu implementacji.

Cały kod źródłowy będzie pisany z wykorzystaniem języka JavaScript, który jest używany przez framework Meteor.js, a jako system bazodanowy posłuży MongoDB.

2.3. Projekt systemu informatycznego

2.3.1. Schemat bazy danych

Na rysunku 1.3.1 przedstawiono schemat bazy danych realizowanego systemu. Składa się on z 9 kolekcji:

- order (rząd),
- family (rodzina),
- gardens (ogrody botaniczne),
- seeds (nasiona),
- indexSeminum (lista index seminum),
- orders (zamówienia nasion),
- plants (rośliny),
- news,
- users (użytkownicy).

Kolekcja **order** przechowuje informacje o kategorii systematycznej rząd roślin znajdujących się w posiadaniu przez ogród botaniczny. Kolekcja **family** przechowuje informacje o kategorii rodzina. W kolekcji **gardens** przechowywane są dane o ogrodach botanicznych, z których otrzymano rośliny (nasiona, sadzonki). Kolekcja **seeds** służy do przechowywania informacji o zebranych nasionach. Na podstawie tej kolekcji możliwe będzie utworzenie kolekcji **indexSeminum**. Kolekcja **orders**

przechowuje informacje o zamówieniach na nasiona, które umieszczone są na liście *index seminum*. Kolekcja **plants**, przechowuje informacje o roślinach znajdujących się w ogrodzie botanicznym. Na podstawie kolekcji **plants** możliwe będzie utworzenie listy *index plantarum*. Kolekcja **news** zawiera informacje (wiadomości) umieszczone przez pracowników ogrodu na stronie głównej serwisu. Kolekcja **users** zawiera dokumenty użytkowników zarejestrowanych w systemie.

order	
_id:	string
name:	string
createdAt:	date
editedAt:	date

family	
_id:	string
name:	string
createdAt:	date
editedAt:	date

plants	
_id:	string
catalogNumber:	string
generalInformations: {	
name: {	
genus:	string
species:	string
variety:	string
subspecies:	string
cultivar:	string
authorship:	string
hybrid:	boolean
}	
namePL:	string
order: {	
id:	string
name:	string
}	
family: {	
id:	string
name:	string
}	
division:	string
clade:	string
type:	string
region_of_living: {	
continent: [
continent:	string
]	
description:	string
}	
habitat:	string
typeOfProtection:	string
endangered:	boolean
rawMaterial: [
rawMaterial:	string
]	
abilityToSow:	int
abilityToPropagate:	int
}	
gardenInformations: {	
place: [
place:	string
]	
quantity:	int
importType:	string
importYear:	int
importGarden:	string
researchAtPharmacy:	boolean
herbarium:	boolean
confirmed:	boolean
}	
characteristic: {	
pharmacopoeial:	boolean
medicinal:	boolean
poisonous:	boolean
edible:	boolean
spice:	boolean
appropriable:	boolean
ornamental:	boolean
bulb:	boolean
evergreen:	boolean
}	
note:	string
state: [{	
quantity:	int
date:	date
}]	
createdAt:	date
editedAt:	date

orders	
_id:	string
ordering_party: {	
name:	string
subname:	string
address: {	
street:	string
number:	string
postalCode:	string
city:	string
country:	string
}	
contact: {	
phone:	string
fax:	string
email:	string
website:	string
}	
representative:	string
}	
order: [
id:	string
]	
orderStatus:	string
verified:	boolean
createdAt:	date
editedAt:	date

seeds	
_id:	string
plant: {	
id:	string
name:	string
}	
quantity:	int
unit:	string
usage:	string
date:	date
note:	string
createdAt:	date
editedAt:	date

gardens	
_id:	string
name:	string
subname:	string
address: {	
street:	string
number:	string
postalCode:	string
city:	string
country:	string
}	
contact: {	
phone:	string
fax:	string
email:	string
website:	string
}	
representative:	string
createdAt:	date
editedAt:	date

indexSeminum	
_id:	string
plant: {	
id:	string
name: {	
genus:	string
species:	string
variety:	string
subspecies:	string
cultivar:	string
authorship:	string
hybrid:	boolean
}	
division:	string
clade:	string
family:	string
}	
order: [{	
year:	int
quantity:	int
}]	
createdAt:	date
editedAt:	date

users	
_id:	string
name:	string
username:	string
emails: [{	
address:	string
verified:	boolean
}]	
roles: [
role:	string
]	
createdAt:	date
editedAt:	date

news	
_id:	string
title:	string
content:	string
author: {	
id:	string
name:	string
}	
createdAt:	date
editedAt:	date

Rys. 2.1. Schemat bazy danych

W poniższych tabelach przedstawiono przykładowe wartości pól dla każdej kolekcji.

Tabela 2.1. Przykładowa zawartość pól w kolekcji order

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
name	Nazwa łacińska kategorii <u>rząd</u> zaczynająca się z wielkiej litery, a kończąca się przyrostkiem –ales. Pisana kursywą	string

Tabela 2.2. Przykładowa zawartość pól w kolekcji family

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
name	Nazwa łacińska kategorii <u>rodzina</u> zaczynająca się z wielkiej litery, a kończąca się przyrostkiem –ae (zwykle –aceae). Pisana kursywą	string

Tabela 2.3. Przykładowa zawartość pól w kolekcji gardens

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
name	Nazwa główna, np. jednostka naukowa	string
subname	Podtytuł, np. nazwa ogrodu botanicznego	string
street	Ulica, na której znajduje się ogród botaniczny	string
number	Numer budynku, do którego przypisany jest ogród botaniczny	string
postalCode	Kod pocztowy ogrodu botanicznego	string
city	Miasto, w którym znajduje się ogród botaniczny	string
country	Kraj, w którym znajduje się ogród botaniczny	string
phone	Numer telefonu ogrodu botanicznego	string
fax	Numer faxu ogrodu botanicznego	string
email	Adres e-mail ogrodu botanicznego	string
website	Strona WWW ogrodu botanicznego	string
representative	Przedstawiciel ogrodu botanicznego	string

Tabela 2.4. Przykładowa zawartość pól w kolekcji seeds

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
plant→id	ID dokumentu z kolekcji plants	string
plant→name	Sparsowana nazwa rośliny z kolekcji plants	string
quantity	Ilość zebranych nasion	int
unit	Jednostka, np. sztuk, gramów	string
usage	Przeznaczenie, np. seminum, laboratorium	string
date	Data zebrania nasion	date
note	Notatka, adnotacja	string

Tabela 2.5. Przykładowa zawartość pól w kolekcji orders

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
name	Nazwa zamawiającego	string
subname	Podtytuł zamawiającego	string
street	Ulica zamawiającego	string
number	Numer budynku zamawiającego	string
postalCode	Kod pocztowy zamawiającego	string
city	Miasto zamawiającego	string
country	Kraj zamawiającego	string
phone	Numer telefonu zamawiającego	string
fax	Numer faxu zamawiającego	string
email	Adres e-mail zamawiającego	string
website	Strona WWW zamawiającego	string
representative	Zamawiający (przedstawiciel ogrodu)	string
order→id	ID dokumentu z kolekcji indexSeminum	string

orderStatus	Status zamówienia nasion: - przetwarzanie, - przygotowywanie do wysyłki, - wysłano	string
verified	Status weryfikacji zamawiającego	boolean

Tabela 2.6. Przykładowa zawartość pól w kolekcji plants

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
catalogNumber	Wewnętrzny numer w postaci alfanumerycznej	string
name→genus	Nazwa łacińska kategorii <i>rodzaj</i> , zaczynająca się z wielkiej litery	string
name→species	Nazwa łacińska kategorii <i>gatunek</i> . Pisana kursywą	string
name→variety	Nazwa łacińska kategorii <i>odmiana</i> . Pisana kursywą	string
name→authorship	Nazwa będąca skrótem od nazwiska autora, który sklasyfikował/odkrył gatunek rośliny	string
namePL	Nazwa polska gatunku (i ewentualnej odmiany) rośliny	string
order→id	ID dokumentu z kolekcji order	string
order→name	Nazwa dokumentu z kolekcji order (rząd rośliny), zaczynająca się z wielkiej litery	string
family→id	ID dokumentu z kolekcji rodziny	string
family→name	Nazwa dokumentu z kolekcji rodziny (rodzina rośliny), zaczynająca się z wielkiej litery	string
division	Nazwa kategorii gromada: - angiosperm, - gymnosperm	string
clade	Nazwa kładu (grupy): - monocotyledon, - dicotyledon	string
type	Charakter rośliny (typ): - tree (drzewo), - shrub (krzew), - littleShrub (krzewinka), - perennialVine (pnącze wieloletnie), - perennial (bylina), - biennial (dwuletnia), - annual (jednoroczna), - annualVine (pnącze jednoroczne)	string
regionOfLiving→continent	Kontynent, na którym występuje roślina: - europe (Europa), - asia (Azja), - africa (Afryka), - northAmerica (Ameryka Północna), - southAmerica (Ameryka Południowa), - australia (Australia) - atlanticOcean (Ocean Atlantycki), - indianOcean (Ocean Indyjski), - pacificOcean (Ocean Spokojny). Możliwe sprecyzowanie (środkowa, zachodnia, wschodnia, północna, południowa)	string
regionOfLiving→description	Dokładniejszy opis miejsca występowania rośliny	string
habitat	Opis naturalnego środowiska życia rośliny	string
typeOfProtection	Zakres ochrony: - fullProtection (ochrona ścisła), - partialProtection (ochrona częściowa), - noProtection (brak ochrony)	string
endangered	Czy zagrożona wyginięciem: - Tak, - Nie	boolean

rawMaterial	Opis części rośliny służącej za surowiec	string
abilityToSow	Zdolność nasiewania: - 0, - 1, - 2, - 3, - 4, - 5, - 6	int
abilityToPropagate	Zdolność rozmnażania za pomocą, np. cebul: Od 0 do 6, tak jak powyżej	int
place	Nazwa miejsca w jakim znajduje się roślina w ogrodzie botanicznym: - decorativeDepartment (Dział Roślin Ozdobnych), - arboretum (Arboretum), - moor (Wrzosowisko), - dune (Wydma), - pharmacognosy (Farmakognozja) - systematics (Dział Systematyki Roślin), - medicinalPlantsSector (Dział Roślin Leczniczych), - greenhouse (Szlarnia)	string
quantity	Liczba oznaczająca ilość roślin danego gatunku w ogrodzie	int
importType	Nazwa formy sprowadzenia rośliny: - seeds (nasiona), - seedling (sadzonka)	string
importYear	Liczba oznaczająca rok sprowadzenia rośliny do ogrodu	int
importGarden	ID dokumentu z kolekcji gardens	string
researchAtPharmacy	Informacja czy przeprowadzono badania na farmacji: - Tak, - Nie	boolean
herbarium	Informacja czy roślina znajduje się w zielniku: - Tak, - Nie	boolean
confirmed	Informacja czy roślina została potwierdzona przez botanika: - Tak, - Nie	boolean
pharmacopoeial	Informacja czy roślina jest farmakopealna: - Tak, - Nie	boolean
medicinal	Informacja czy roślina jest lecznicza: - Tak, - Nie	boolean
poisonous	Informacja czy roślina jest trująca: - Tak, - Nie	boolean
edible	Informacja czy roślina jest jadalna: - Tak, - Nie	boolean
spice	Informacja czy roślina jest przyprawowa: - Tak, - Nie	boolean
appropriable	Informacja czy roślina jest użytkowa: - Tak, - Nie	boolean
ornamental	Informacja czy roślina jest ozdobna: - Tak, - Nie	boolean
bulb	Informacja czy roślina jest cebulowa:	boolean

	- Tak, - Nie	
evergreen	Informacja czy roślina jest zimozielona: - Tak, - Nie	boolean
note	Notatka, adnotacja	string
state→quantity	Liczba informująca o stanie liczby roślin w danym dniu	int
state→date	Data odpowiadająca dniu zmiany liczby roślin danego gatunku w ogrodzie botanicznym	date

Tabela 2.7. Przykładowa zawartość pól w kolekcji indexSeminum

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
plant→id	ID dokumentu z kolekcji plants	string
name→genus	Rodzaj rośliny, tak jak w kolekcji plants	string
name→species	Gatunek rośliny, tak jak w kolekcji plants	string
name→variety	Odmiana rośliny, tak jak w kolekcji plants	string
name→subspecies	Podgatunek rośliny, tak jak w kolekcji plants	string
name→cultivar	Kultywar rośliny, tak jak w kolekcji plants	string
name→authorship	Autor rośliny, tak jak w kolekcji plants	string
name→hybrid	Czy roślina jest hybrydą, tak jak w kolekcji plants	boolean
plant→division	Gromada rośliny, tak jak w kolekcji plants	string
plant→clase	Kład rośliny, tak jak w kolekcji plants	string
plant→family	Rodzina rośliny, tak jak w kolekcji plants	string
years→year	Rok zebrania nasion	int
years→quantity	Ilość zebranych nasion	int

Tabela 2.8. Przykładowa zawartość pól w kolekcji news

Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
title	Tytuł news'a	string
content	Treść news'a	string
author→id	ID dokumentu z kolekcji users	string
author→name	Nazwa autora (użytkownika) z kolekcji users	string

Tabela 2.9. Przykładowa zawartość pól w kolekcji users

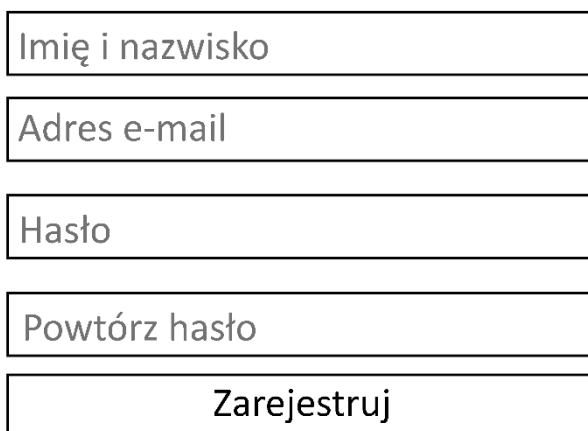
Pole	Wartość	Typ
_id	Generowana automatycznie przez system	string
name	Imię i nazwisko użytkownika	string
username	Login użytkownika	string
emails→address	Adres e-mail użytkownika	string
emails→verified	Informacja czy użytkownik jest zweryfikowany: - Tak, - Nie	boolean
roles	Nazwa uprawnień użytkownika: - admin (uprawnienia administratora), - orders (uprawnienia do zamówień), - plants (uprawnienia do roślin), - seeds (uprawnienia do nasion), - gardens (uprawnienia do ogrodów), - news (uprawnienia do news'ów)	string

2.3.2. Projekt interfejsu graficznego

Projekt GUI (Graphical User Interface) jest wstępną i graficzną reprezentacją elementów systemu, umożliwiającą użytkownikowi wykonywanie czynności w systemie. To istotny etap w projektowaniu, ponieważ interfejs powinien być przemyślany, intuicyjny i funkcjonalny. Poniżej zostały

przedstawione wstępne koncepcje interfejsów graficznych najważniejszych części systemu internetowego.

Zarejestruj się



A registration form with five input fields and one button. The fields are labeled 'Imię i nazwisko', 'Adres e-mail', 'Hasło', and 'Powtórz hasło'. The button is labeled 'Zarejestruj'.

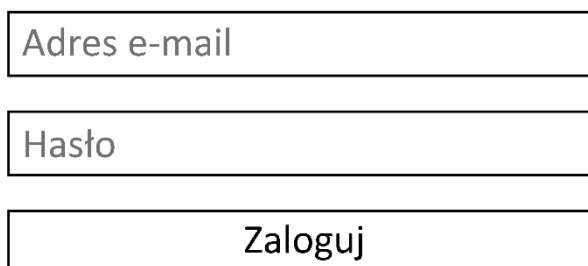
Imię i nazwisko
Adres e-mail
Hasło
Powtórz hasło
Zarejestruj

Masz już konto? *Zaloguj się*

Rys. 2.2. Projekt formularza rejestracji

Rysunek 1.2. prezentuje projekt formularza rejestracji. W celu utworzenia nowego konta wystarczy podać podstawowe dane osobowe (imię i nazwisko), adres e-mail oraz hasło. Wybranie przycisku *Zarejestruj* spowoduje założenie nowego oraz wysłanie adresu z linkiem aktywacyjnym. Ponadto istnieje możliwość powrotu do ekranu logowania po kliknięciu odnośnika znajdującego się pod przyciskiem.

Zaloguj się



A login form with two input fields and one button. The fields are labeled 'Adres e-mail' and 'Hasło'. The button is labeled 'Zaloguj'.

Adres e-mail
Hasło
Zaloguj

Nie masz konta? *Zarejestruj się*

Zapomniałeś hasła? *Kliknij tutaj*

Rys. 2.3. Projekt formularza logowania

Na rysunku 1.3. ukazano projekt formularza logowania. Jest on dość prosty i składa się z pól, do których wpisuje się adres e-mail oraz hasło. Dodatkowymi funkcjonalnościami są przypomnienie hasła z wykorzystaniem adresu mailowego oraz możliwość rejestracji w przypadku braku konta.

generuj raport

dodaj roślinę

generuj listę

Rys. 2.4. Projekt strony widocznej po zalogowaniu się do systemu

Po poprawnej rejestracji użytkownik będzie mógł zalogować się do systemu. Jeśli proces logowania przebiegnie pomyślnie, w oknie przeglądarki użytkownika pojawi się panel widoczny powyżej (Rys. 1.4.). Na górnym pasku widoczne będą opcje, które umożliwią podejrzanie m.in. dostępnych roślin czy zaznajomionych ogrodów botanicznych. Ponadto możliwe będzie przejrzanie listy dostępnych użytkowników (zakładka *Użytkownicy*), zarządzanie kontem (*Konto*) oraz wylogowanie z systemu (*Wyloguj*).

Rośliny

Szukaj

OK

+ Dodaj

Eksportuj

Usuń wszystkie dane

ID	param1	param2	param3	param4	param5	param6
----	--------	--------	--------	--------	--------	--------

1						
---	--	--	--	--	--	--

2						
---	--	--	--	--	--	--

Rys. 2.5. Projekt strony widocznej po wybraniu zakładki *Rośliny*

Na rysunku 1.5. przedstawiono projekt strony widocznej po wybraniu zakładki *Rośliny*. Daje ona sporo możliwości, jeśli chodzi o zarządzanie roślinami w ogrodzie botanicznym. Zakładka ta pozwoli na wyszukiwanie interesujących nas roślin, eksportowanie danych do pliku, czy też usunięcie wszystkich danych za pomocą jednego przycisku. Dane zawarte w bazie wyświetlane są w formie tabeli.

Start Rośliny Szklarnia Ogrody botaniczne Rodzina Rząd Konto Wyloguj

Konto

Zmiana danych

Zmiana hasła

Usuń konto

Zmiana hasła

Stare hasło

Stare hasło

Nowe hasło

Nowe hasło

Powtórz hasło

Powtórz nowe hasło

Anuluj

Zapisz

Rys. 2.6. Projekt strony umożliwiającej zarządzanie kontem

Kolejną funkcjonalnością będzie zarządzanie kontem osoby korzystającej z systemu internetowego. W zakładce tej użytkownik ma możliwość zmiany swoich danych osobowych, hasła, bądź usunięcia konta. Na rysunku 1.6. została przedstawiona sytuacja, w której osoba korzystająca z systemu pragnie dokonać zmiany hasła. W takim przypadku system internetowy prosi użytkownika o podanie swojego dotychczasowego hasła, a następnie nowego. Po uzupełnieniu wszystkich danych należy kliknąć przycisk *Zapisz*, aby zatwierdzić wprowadzone zmiany.

WYKAZ RYSUNKÓW

Rys. 2.1. Schemat bazy danych.....	4
Rys. 2.2. Projekt formularza rejestracji.....	9
Rys. 2.3. Projekt formularza logowania.....	9
Rys. 2.4. Projekt strony widocznej po zalogowaniu się do systemu.....	10
Rys. 2.5. Projekt strony widocznej po wybraniu zakładki <i>Rośliny</i>	10
Rys. 2.6. Projekt strony umożliwiającej zarządzanie kontem	11

WYKAZ TABEL

Tabela 2.1. Przykładowa zawartość pól w kolekcji order	5
Tabela 2.2. Przykładowa zawartość pól w kolekcji family	5
Tabela 2.3. Przykładowa zawartość pól w kolekcji gardens.....	5
Tabela 2.4. Przykładowa zawartość pól w kolekcji seeds	5
Tabela 2.5. Przykładowa zawartość pól w kolekcji orders.....	5
Tabela 2.6. Przykładowa zawartość pól w kolekcji plants	6
Tabela 2.7. Przykładowa zawartość pól w kolekcji indexSeminum.....	8
Tabela 2.8. Przykładowa zawartość pól w kolekcji news.....	8
Tabela 2.9. Przykładowa zawartość pól w kolekcji users	8