



Zachodniopomorski
Uniwersytet
Technologiczny
w Szczecinie



Wydział
Informatyki

Virtual Python School - Kurs do nauki programowania w języku Python

Virtual Python School - Learning course for Python language

Autorzy

Paweł Kałdus

Kacper Dogiel

Sebastian Byczyk

Mariusz Lubowicki

Zespołowy projekt inżynierski
tworzony pod kierunkiem
dr hab. inż. Krzysztofa Małeckiego

Spis treści

1	Cel i założenia projektu	2
2	Projekt aplikacji	3
2.1	Podział ról	3
2.2	Zastosowane technologie	4
2.3	Diagram klas	5
2.4	Wykaz czynności	7
2.5	Aktualny stan projektu	9

1 Cel i założenia projektu

Projekt Virtual Python School to interaktywny kurs języka Python wspomagający naukę zarówno dla początkujących jak i zaawansowanych osób. Kurs podzielony jest na lekcje ze stopniowo rosnącym poziomem trudności zagadnień. Dodatkowo posiada interpreter języka Pythona, który pozwala na przetestowanie kodu. Aplikacja ma na celu zachęcenie do nauki języka Python. W tym celu projekt został zaprojektowany w zgodzie z trendami przy użyciu nowoczesnych technologii i przyjaznego dla oka interfejsu w wersji jasnej i ciemnej. Strona wspiera język polski i angielski z łatwą możliwością rozbudowy o kolejne języki. Do uwierzytelnienia użytkowników wykorzystano uczelniany system logowania poprzez mechanizm IPA wykorzystujący dane logowania studentów. Istnieje jednak możliwość implementacji dodatkowych systemów uwierzytelnienia dla użytkowników spoza uczelni.

2 Projekt aplikacji

2.1 Podział ról

Osoba	Role
Paweł Kałdus	Programista Front-End
Mariusz Lubowicki	Programista Back-End, Scrum master
Sebastian Bczyk	Programista Back-End
Kacper Dogiel	Programista Back-End

2.2 Zastosowane technologie

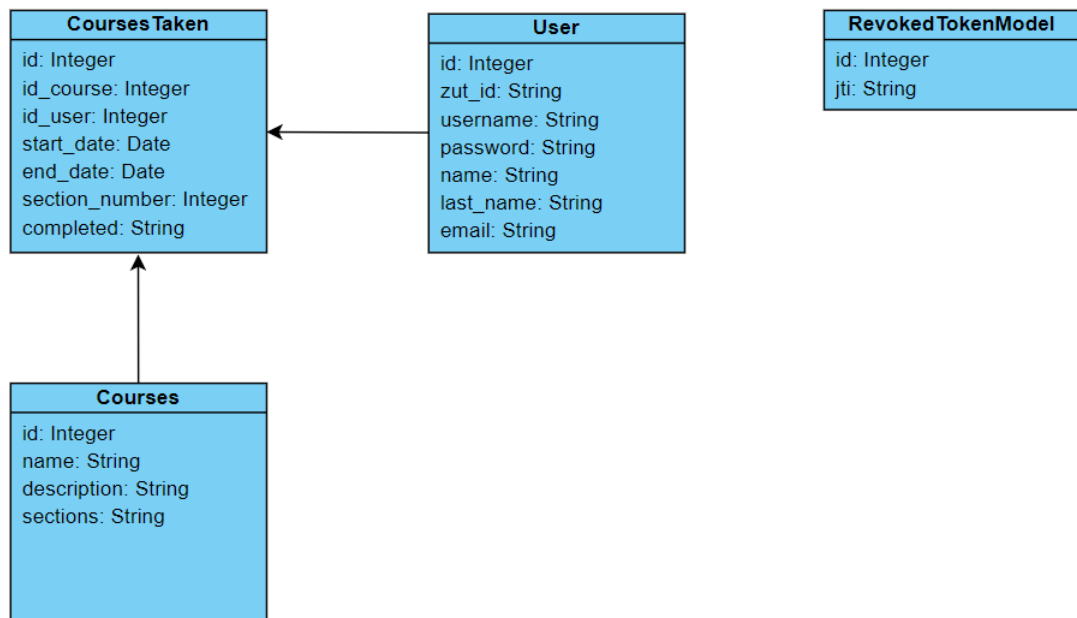
Front-end	
Nazwa technologii	Zastosowanie
React.js	Biblioteka do tworzenia interfejsów webowych
Next.js	Framework opierający się na React.js wspierający budowę rozbudowanych aplikacji webowych
Tailwind CSS	Modułowy framework CSS
Redux	Zarządzanie stanem aplikacji React.js

Back-end	
Nazwa technologii	Zastosowanie
Flask	Biblioteka do tworzenia aplikacji webowych
SQLAlchemy	Biblioteka służąca do pracy z bazami danych typu ORM (Object-relational mapping)
Flask-JWT	Biblioteka służąca tworzeniu tokenu autoryzacji JWT (JSON Web Token)
ipahhttp	Framework do obsługi połączenia z ipa w celu uwierzytelnienia z wewnętrznymi serwerami ZUT
pytest	Biblioteka do tworzenia testów jednostkowych

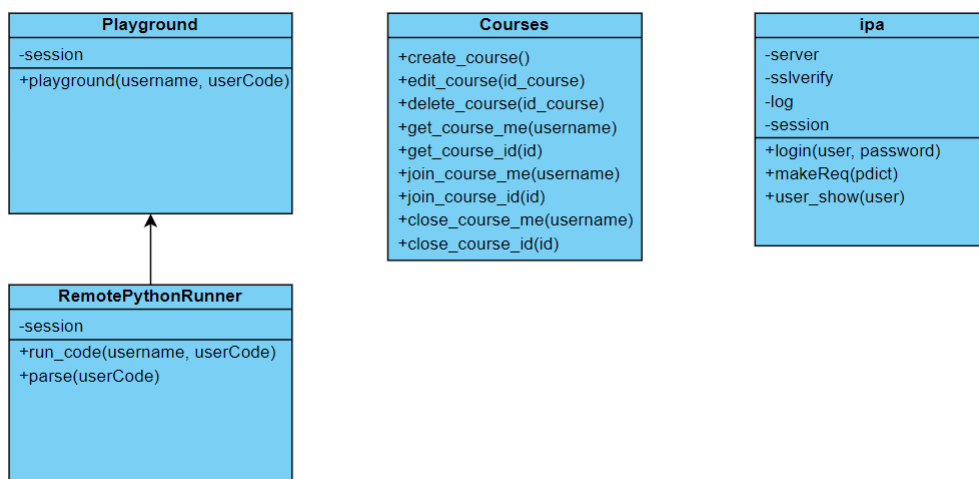
Inne	
Nazwa technologii	Zastosowanie
MySQL	Baza danych
LXC	Lekkie kontenery Linuksowe
Nginx	Serwer HTTP, który może pracować również jako reverse proxy (czyli odwrotny serwer pośredniczący) dla innego serwera HTTP, load balancer, proxy

2.3 Diagram klas

- **ipa** - Klasa odpowiedzialna za połączenie z uczelnianą bazą danych użytkowników
- **Playground** - Serwis odpowiedzialny za możliwość wpisywania kodu oraz wyświetlanie przetworzonego kodu.
- **Courses** - Serwis odpowiedzialny za wyświetlanie wszystkich dostępnych kursów, możliwość ich usuwania, dodawania oraz dołączania do nich.
- **RemotePythonRunner** - Klasa odpowiedzialna za przyjmowanie wprowadzonego na stronie kodu, parsowanie go, uruchamianie na kontenerze danego użytkownika oraz zwracanie danych wyjściowych.



Rysunek 1: Diagram mapowanych klas na strukturę bazy danych (ORM)



Rysunek 2: Diagram klas

2.4 Wykaz czynności

Title	Assigned To	State
Design UI/UX of a website	Paweł Kałdus	Done
Choose frameworks and programming languages	All	Done
Set-up Git repo	Mariusz Lubowicki	Done
Prepare and configure basic website structure	Paweł Kałdus	Done
Setup server with flask, mysql, python, node.js	All	Done
Set-up i18n(internationalization and localization) on frontend	Paweł Kałdus	Done
Database UML Diagram/Structure	Mariusz Lubowicki	Done
Setup database on server	All	Done
Landing page	Paweł Kałdus	Done
Test database sql dump	Mariusz Lubowicki	Done
Dark theme	Paweł Kałdus	Done
Login page	Paweł Kałdus	Done
Register page	Paweł Kałdus	Done
Form validation	Paweł Kałdus	Done
Configure server	All	Done
Beginner course	Mariusz Lubowicki	Done
Authorization		Done
Parse JSON from IPA	Sebastian Bczyk	Done
Deliver JSON to Front-End	Kacper Dogiel	Done
Show JSON data in Front-End	Paweł Kałdus	Done
Function to create users in database by IPA login	Sebastian Bczyk	Done
Prepare api client on frontend	Paweł Kałdus	Done
Set-up state management with Redux on frontend	Paweł Kałdus	Done
Code playground	Mariusz Lubowicki	Done
Website for code playground	Paweł Kałdus	Done
Choose container software	Mariusz Lubowicki	Done
Run users code in containers	Mariusz Lubowicki	Done
Catch output/errors from container	Mariusz Lubowicki	Done
Show output/Handle errors on site	Paweł Kałdus	Done
Connect with api/back-end	Mariusz Lubowicki	Done
Set-up reverse proxy with nginx for production	Paweł Kałdus	Done
Store JWT token in a cookie	Paweł Kałdus	Done
Logout user	Paweł Kałdus	Done
Refresh user token	Sebastian Bczyk	Done
Set-up unit testing and storybook	Paweł Kałdus	Done
Add login route at backend	Sebastian Bczyk	Done
Prepare connection to ipa	Kacper Dogiel	Done
Add user routes	Kacper Dogiel	Done

Add jwt routes to backend	Sebastian Byczyk	Done
Prepare course management page	Paweł Kałdus	To Do
Prepare backend for courses		To Do
Adding courses functionality	Kacper Dogiel	Done
Editing courses functionality	Sebastian Byczyk	Done
Deleting courses functionality	Kacper Dogiel	Done
Set up routes	Kacper Dogiel	Done
Returning user's courses functionality	Sebastian Byczyk	Done
Return course by id functionality	Sebastian Byczyk	Doing
Adding users to courses functionality	Kacper Dogiel	Done
Add user to a course by id functionality (for admin)	Kacper Dogiel	Doing
Set course as done (when user ends course)	Sebastian Byczyk	Done
Set course as done for user by course id (for admin)	Sebastian Byczyk	Done
Set-up routes	Mariusz Lubowicki	Done
Parse code sent from Playground	Mariusz Lubowicki	To Do
Handle infinite loops	Mariusz Lubowicki	Doing
Handle imports	Mariusz Lubowicki	Done
Handle Infinite recurrency functions	Mariusz Lubowicki	To Do
Timeout executing after some time	Mariusz Lubowicki	To Do
Container creation/Error handling	Mariusz Lubowicki	To Do
Timeout container if user inactive	Mariusz Lubowicki	To Do
Re-run container for existing user	Mariusz Lubowicki	To Do
Create container for new user	Mariusz Lubowicki	To Do
Admin page	Paweł Kałdus	Doing
Add course-rating sub-site	Paweł Kałdus	To Do
Finish course selection site	Paweł Kałdus	To Do
Set up server for containers	Mariusz Lubowicki	Done

2.5 Aktualny stan projektu

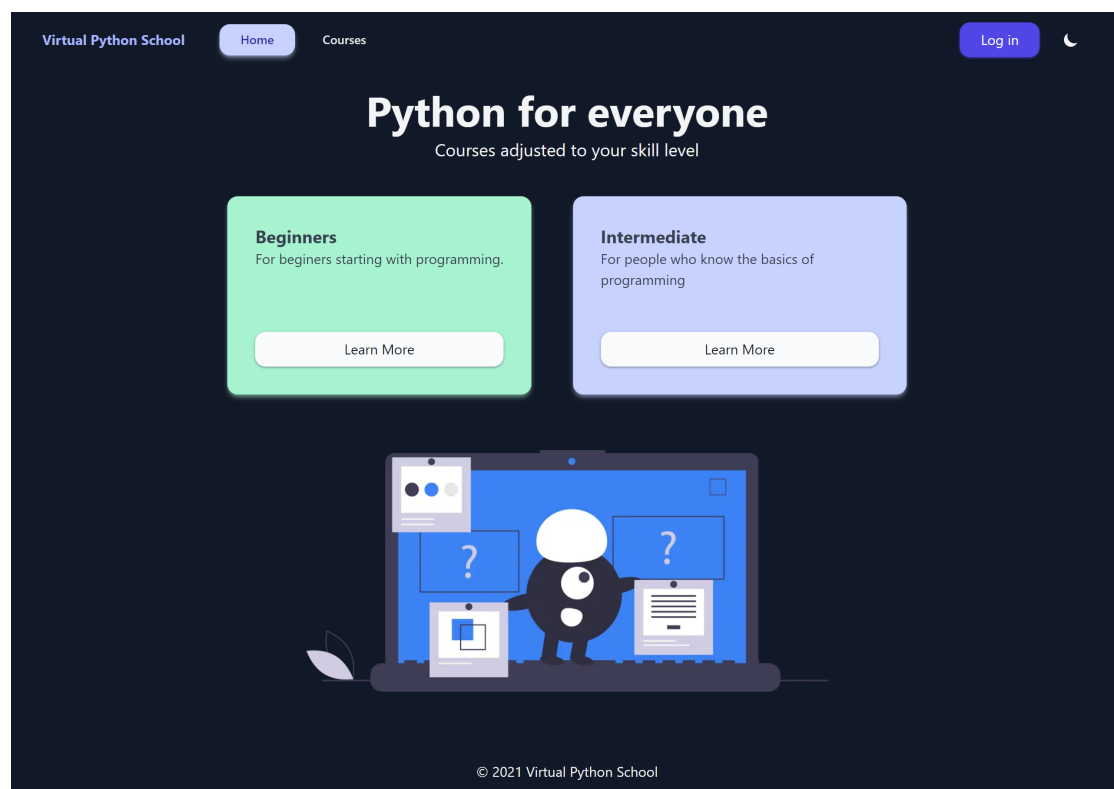
Link do strony: <http://virtualschool.wi.zut.edu.pl> (wymagany dostęp poprzez uczelniany VPN).



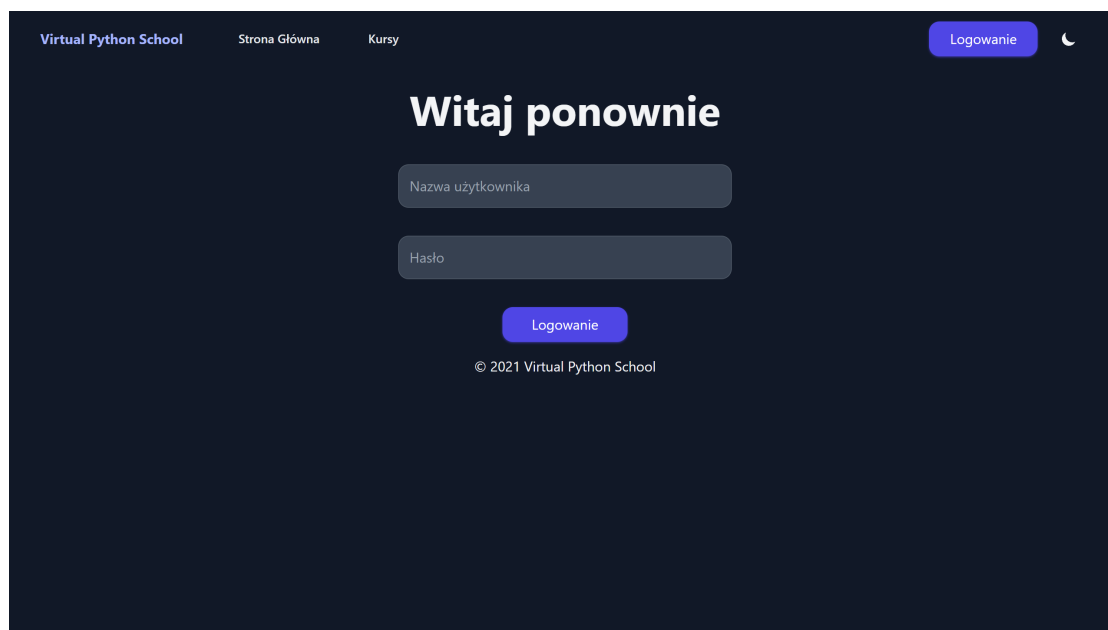
Rysunek 3: Strona główna w jasnym motywie



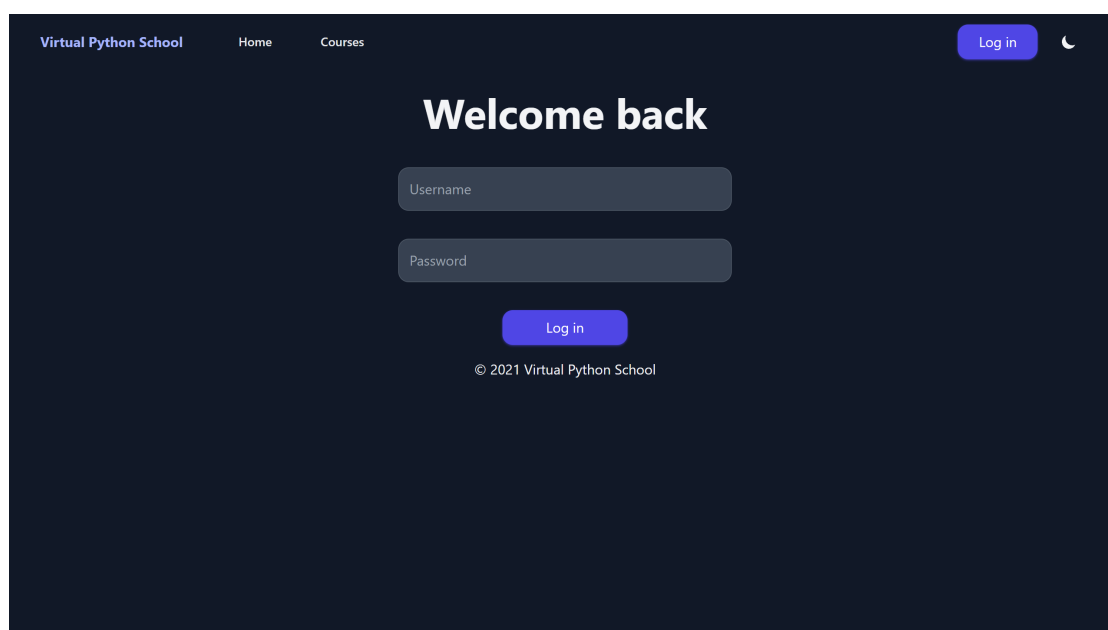
Rysunek 4: Strona główna bez zalogowania



Rysunek 5: Strona główna w języku angielskim



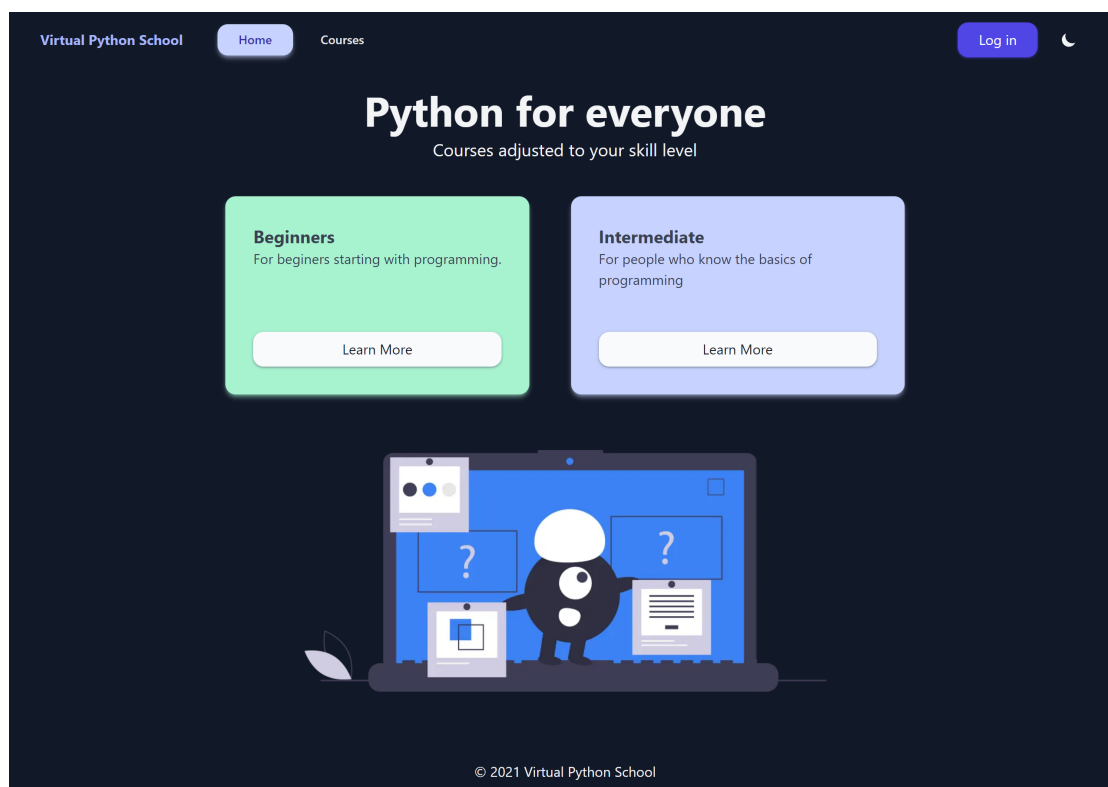
Rysunek 6: Strona logowania



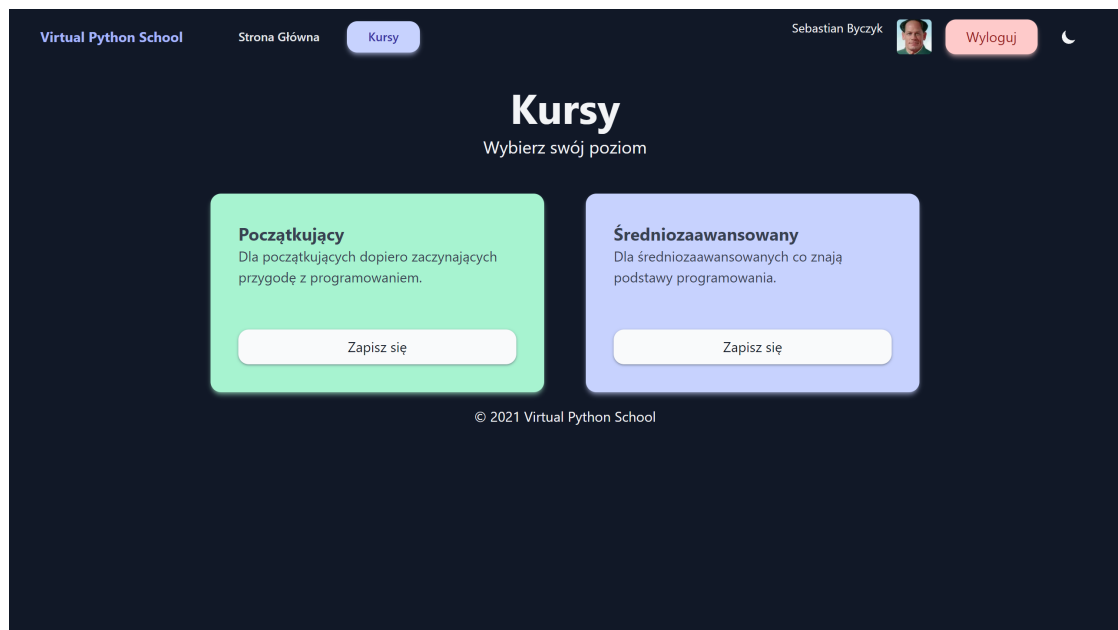
Rysunek 7: Strona logowania w języku angielskim



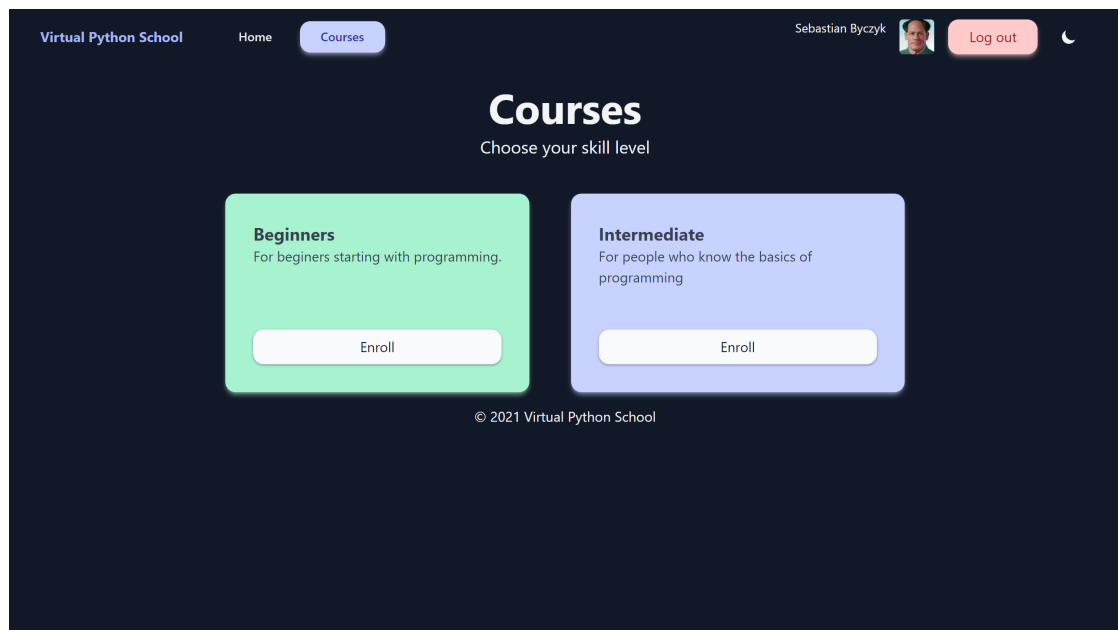
Rysunek 8: Strona główna po zalogowaniu



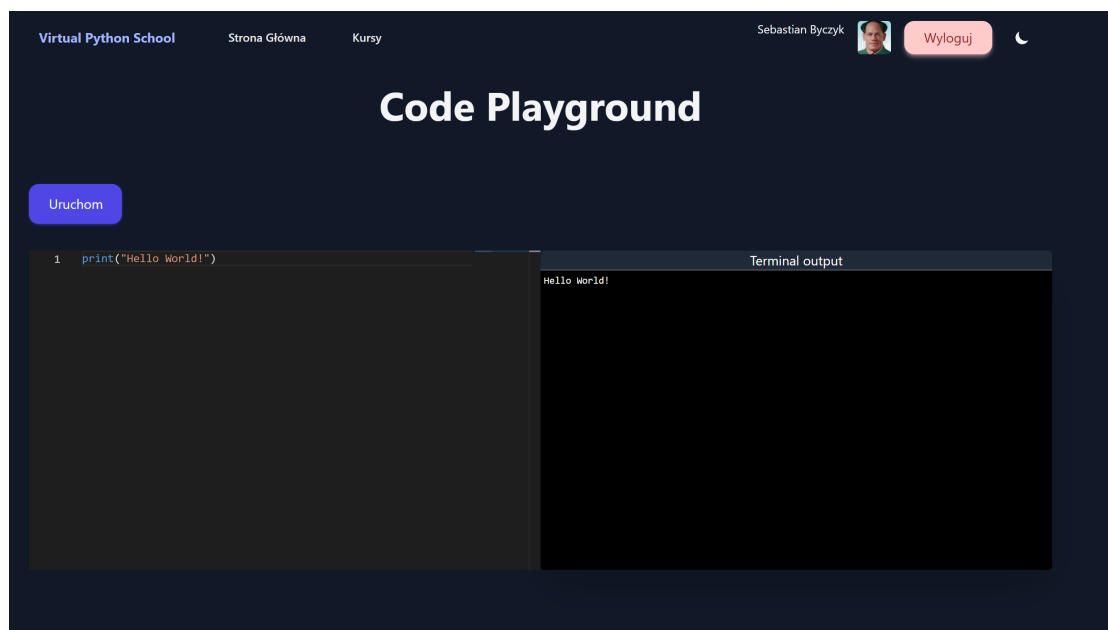
Strona główna po zalogowania w języku angielskim



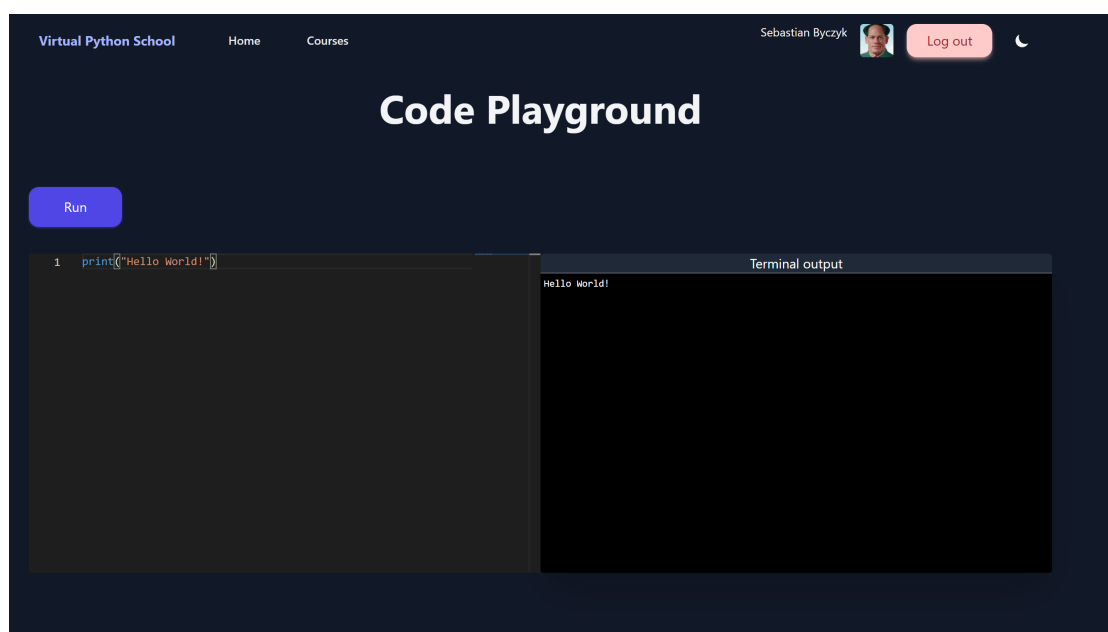
Rysunek 9: Wybór kursu



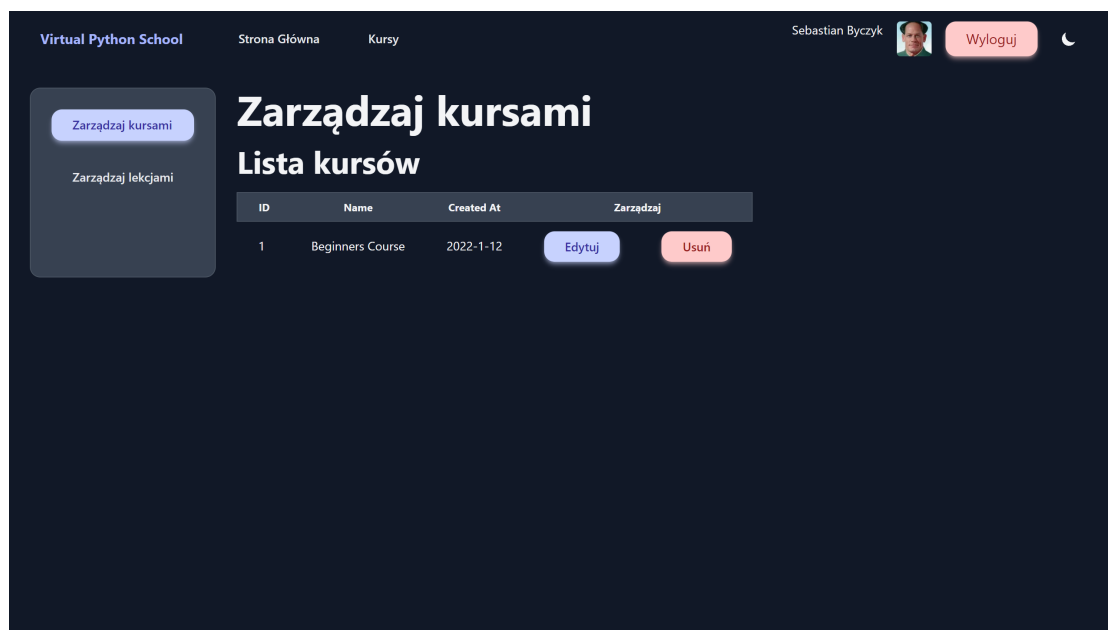
Rysunek 10: Wybór kursu w języku angielskim



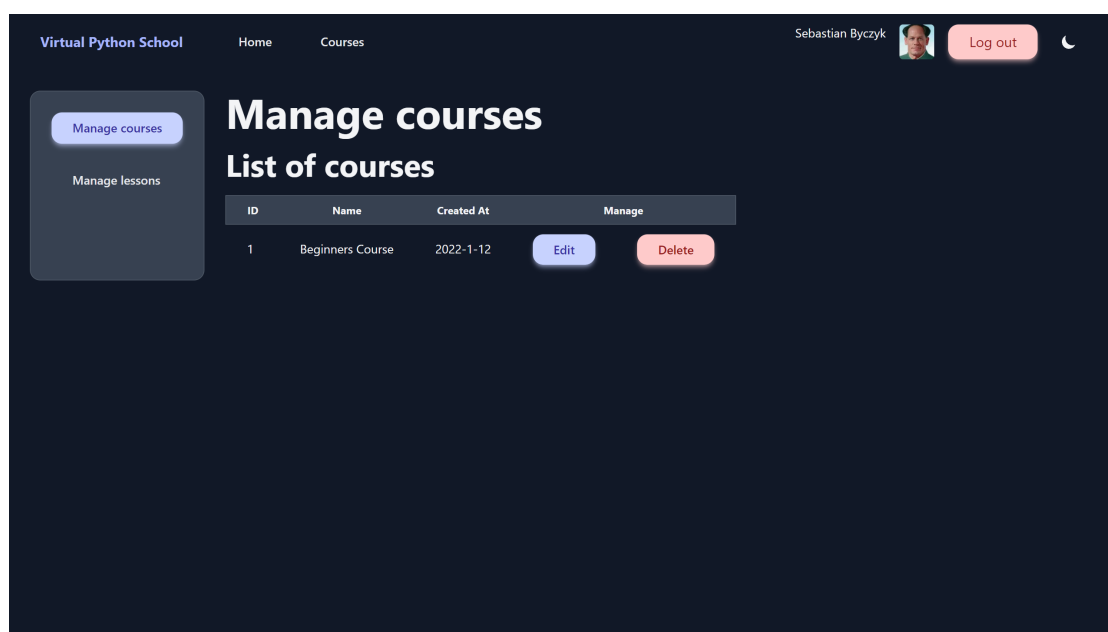
Rysunek 11: Code playground



Rysunek 12: Code playground w języku angielskim



Rysunek 13: Panel zarządzania kursami



Rysunek 14: Panel zarządzania kursami w języku angielskim