

# Projekt Programowanie Aplikacji w Chmurze Obliczeniowej

Temat: System analizowania popularności języków  
programowania



Autorzy: Michał Wrona, Damian Wojtal

Grupa: IO 6.10

<b>Opis</b>	<b>3</b>
<b>Pliki Docker</b>	<b>4</b>
Drzewo projektu	4
Zawartość pliku docker-compose.yaml	5
Plik Dockerfile usługi ReactJS	6
Plik Dockerfile usługi Express	6
Plik Dockerfile usługi MongoDB	7
Pliki .dockerignore dla usług Express oraz ReactJS	7
<b>Widoki projektu</b>	<b>8</b>
Strona logowania	8
Strona rejestracji	8
Widok strony głównej	9
Widok formularza	10
<b>Sortowanie zawartości tabel</b>	<b>11</b>
Najpopularniejsze języki programowania w zależności od regionów świata	11
Sposób sortowania tabeli	12
Najpopularniejsze języki programowania w zależności od wieku	12
Sposób sortowania tabeli:	13
Najpopularniejsze języki programowania w zależności od wykształcenia	14
Sposób sortowania tabeli:	14
Najpopularniejsze języki programowania w zależności od płci	15
Sposób sortowania tabeli:	15
Najpopularniejsze języki programowania w zależności od wieku i regionu świata	16
Sposób sortowania tabeli:	16
<b>Wnioski</b>	<b>16</b>
Sposób uruchomienia projektu:	17
Sposób dockeryzowania projektu:	17

# Opis

System analizuje dane na temat popularności różnych języków programowania używanych przez użytkowników w różnym wieku, rejonach świata, płci oraz wykształcenia. Frontend aplikacji został wykonany w ReactJS. Do stworzenia serwera aplikacji wykorzystaliśmy szkielet Express. Baza danych została wykonana w systemie MongoDB. Aplikacja działa w kontenerach Docker.

System będzie posiadał następujące funkcjonalności:

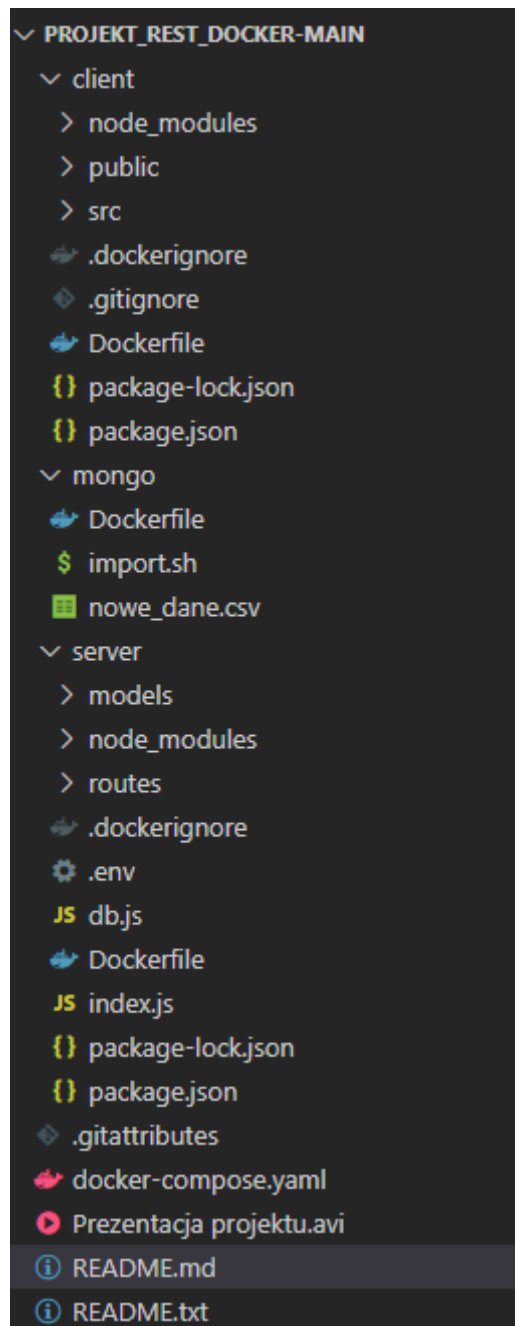
- Mechanizm uwierzytelnienia i autoryzacji
- Formularz dodający dane do bazy danych
- Eksport danych do pliku w formacie JSON
- Strona wyświetlająca wybrane dane przez użytkownika
- Obsługa usługi SOAP dla pobierania danych

Poza wymaganiami funkcjonalnymi aplikacja będzie spełniała następujące wymagania niefunkcjonalne:

- Kontener Docker obejmujący stworzoną aplikację

# Pliki Docker

## Drzewo projektu



## Zawartość pliku docker-compose.yaml

```
version: "3"
services:
  server:
    build: ./server
    ports:
      - "8080:8080"
    volumes:
      - logs:/app/logs
      - ./server:/app
      - /app/node_modules
    depends_on:
      - mongodb
    networks:
      - mern-app
    restart: always
  client:
    build: ./client
    ports:
      - "3000:3000"
    volumes:
      - ./client/src:/app/src
    stdin_open: true
    networks:
      - mern-app
  mongodb:
    build: ./mongo
    volumes:
      - data:/data/db
    ports:
      - "27017:27017"
    networks:
      - mern-app
networks:
  mern-app:
    driver: bridge
volumes:
  data:
  logs:
```

## Plik Dockerfile usługi ReactJS

```
FROM node

WORKDIR /app

COPY package.json .

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]
```

## Plik Dockerfile usługi Express

```
FROM node

WORKDIR /app

COPY package.json .

RUN npm install

COPY . .

EXPOSE 8080

CMD ["npm", "start"]
```

## Plik Dockerfile usługi MongoDB

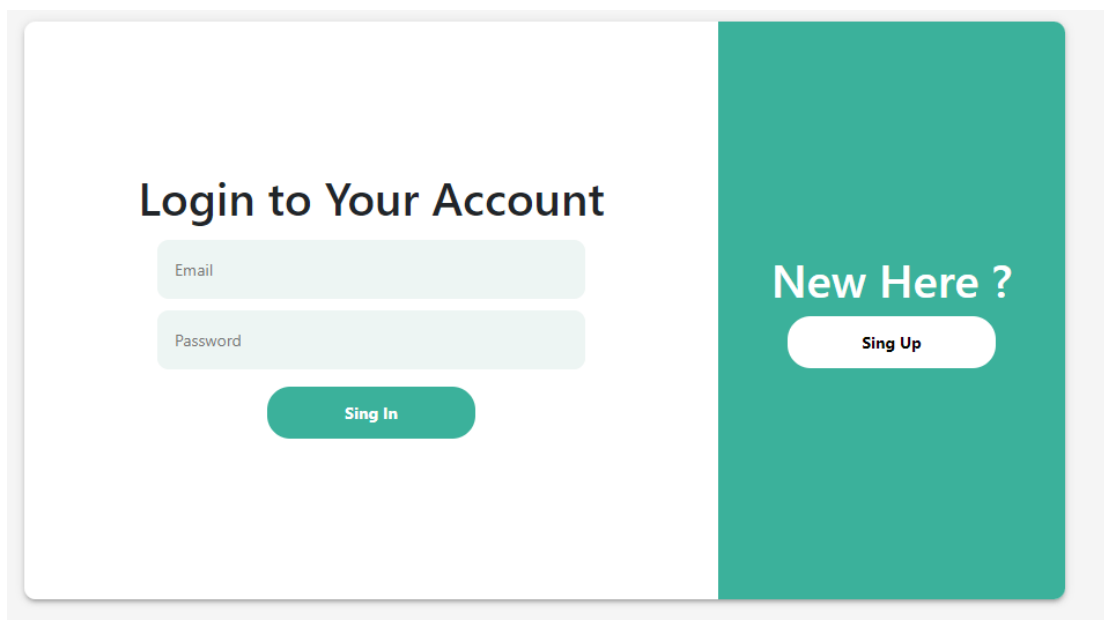
```
FROM mongo:4
COPY nowe_dane.csv /nowe_dane.csv
COPY import.sh /import.sh
RUN ["chmod", "777", "/import.sh"]
```

## Pliki .dockerignore dla usług Express oraz ReactJS

```
node_modules
Dockerfile
.git;
```

## Widoki projektu

### Strona logowania



The login page features a white background with a teal sidebar on the right. The main content area has the heading "Login to Your Account". Below it are two light green input fields labeled "Email" and "Password". A teal "Sing In" button is positioned below the password field. The teal sidebar contains the text "New Here ?" and a white "Sing Up" button.

**Login to Your Account**

Email

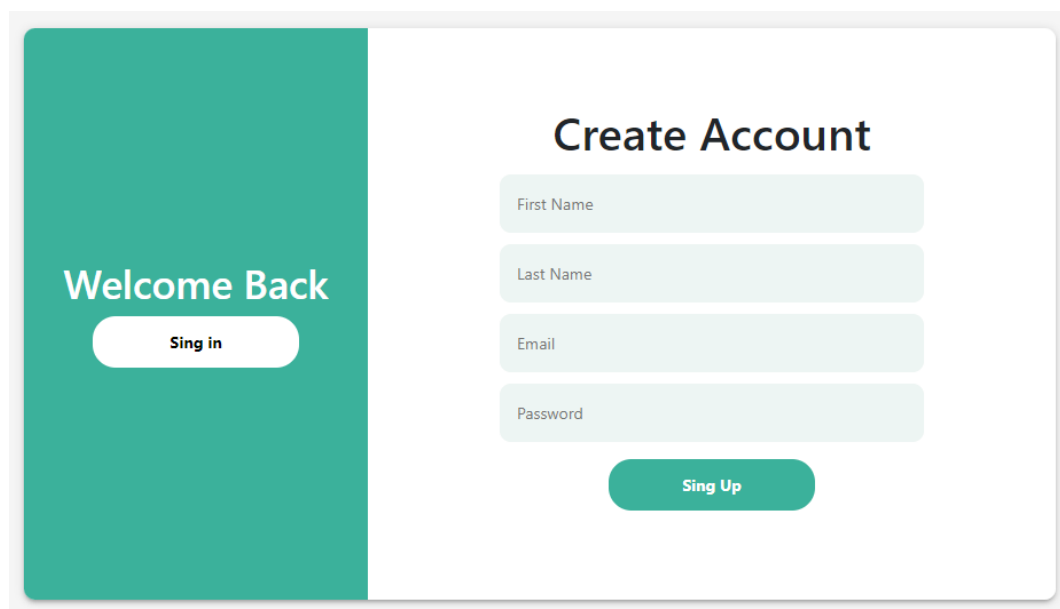
Password

**Sing In**

**New Here ?**

**Sing Up**

### Strona rejestracji



The registration page features a teal sidebar on the left and a white main content area. The sidebar contains the text "Welcome Back" and a white "Sing in" button. The main content area has the heading "Create Account". Below it are four light green input fields labeled "First Name", "Last Name", "Email", and "Password". A teal "Sing Up" button is positioned below the password field.

**Welcome Back**

**Sing in**

**Create Account**

First Name

Last Name

Email

Password

**Sing Up**



## Widok strony głównej

Projekt			Formularz	Logout
Most popular languages based on region of the world. ▾			Export current	Export all
Region	Language	Votes		
Argentina	Python	39		
Australia	Python	99		
Austria	Python	14		
Bangladesh	Python	35		
Belarus	Python	29		
Belgium	Python	34		
Brazil	Python	276		
Canada	Python	201		
Chile	Python	24		
China	Python	581		
Colombia	Python	40		
Czech Republic	Python	24		
Denmark	Python	45		
Egypt	Python	35		
Finland	Python	29		
France	Python	223		
Germany	Python	306		
Greece	Python	46		
Hong Kong (S.A.R.)	Python	24		
Hungary	Python	33		
I do not wish to disclose my location	Python	98		
India	Python	1550		
Indonesia	Python	39		
Iran Islamic Republic of...	Python	36		
Ireland	Python	28		

## Widok formularza

### Complete form

What is your gender?

Male



What is your age?

18-21



In which country do you currently reside?

Country

What is the highest level of formal education that you have attained or plan to attain within the next 2 years?

Education

What is your undergraduate major?

Major

What is your current role at your work?

Current role

In what industry is your current employer/contract?

Industry

What specific programming language do you use most often?

Programming language

Send

Reset

## Sortowanie zawartości tabel

Najpopularniejsze języki programowania w zależności od regionów świata

Most popular languages based on region of the world. ▼

Export currentExport all

Region	Language	Votes
Argentina	Python	39
Australia	Python	99
Austria	Python	14
Bangladesh	Python	35
Belarus	Python	29
Belgium	Python	34
Brazil	Python	276
Canada	Python	201
Chile	Python	24
China	Python	581
Colombia	Python	40
Czech Republic	Python	24
Denmark	Python	45
Egypt	Python	35
Finland	Python	29
France	Python	223
Germany	Python	306
Greece	Python	46
Hong Kong (S.A.R.)	Python	24
Hungary	Python	33
I do not wish to disclose my location	Python	98
India	Python	1550
Indonesia	Python	39
Iran Islamic Republic of...	Python	36

## Sposób sortowania tabeli

```
router.route('/fetchData/language').get((req, res) => {
  const aggr = [
    { $group: { _id: { Q3: '$Q3', Q8: '$Q8' }, count: { $sum: 1 } } },
    { $sort: { count: -1 } },
    {
      $group: {
        _id: { Q3: '$_id.Q3' }, Q8: { $first: '$_id.Q8' },
        count: { $first: '$count' }
      }
    },
    { $sort: { _id: 1 } }
  ]

  Form.Forms.aggregate(aggr).exec((error, data) => {
    if (error) {
      return next(error)
    } else {
      res.json(data)
    }
  })
})
```

## Najpopularniejsze języki programowania w zależności od wieku

Most popular languages based on the age. <span>▼</span>			<a href="#">Export current</a>	<a href="#">Export all</a>
Age	Language	Votes		
18-21	Python	1043		
22-24	Python	1825		
25-29	Python	2239		
30-34	Python	1290		
35-39	Python	695		
40-44	Python	434		
45-49	Python	266		
50-54	Python	177		
55-59	Python	114		
60-69	Python	84		
70-79	R	11		
80+	Python	4		

Sposób sortowania tabeli:

```
router.route('/fetchData/language_age').get((req, res) => {
  const aggr = [
    { $group: { _id: { Q2: '$Q2', Q8: '$Q8' }, count: { $sum: 1 } } },
    { $sort: { count: -1 } },
    {
      $group: {
        _id: { Q2: '$_id.Q2' }, Q8: { $first: '$_id.Q8' },
        count: { $first: '$count' }
      }
    },
    { $sort: { _id: 1 } }
  ]

  Form.Forms.aggregate(aggr).exec((error, data) => {
    if (error) {
      return next(error)
    } else {
      res.json(data)
    }
  })
})
```

## Najpopularniejsze języki programowania w zależności od wykształcenia

Most popular languages based on degree. ▼ Export current Export all

Degree	Language	Votes
Bachelor's degree	Python	2366
Doctoral degree	Python	1344
I prefer not to answer	Python	75
Master's degree	Python	3885
No formal education past high school	Python	34
Professional degree	Python	174
Some college/university study without earning a bachelor's degree	Python	302

Sposób sortowania tabeli:

```
router.route('/fetchData/language_degree').get((req, res) => {
  const aggr = [
    { $group: { _id: { Q4: '$Q4', Q8: '$Q8' }, count: { $sum: 1 } } },
    { $sort: { count: -1 } },
    {
      $group: {
        _id: { Q4: '$_id.Q4' }, Q8: { $first: '$_id.Q8' },
        count: { $first: '$count' }
      }
    },
    { $sort: { _id: 1 } }
  ]

  Form.Forms.aggregate(aggr).exec((error, data) => {
    if (error) {
      return next(error)
    } else {
      res.json(data)
    }
  })
})
```

## Najpopularniejsze języki programowania w zależności od płci

Most popular languages based on sex. ▼ Export current Export all

Sex	Language	Votes
Female	Python	1130
Male	Python	6935
Prefer not to say	Python	86
Prefer to self-describe	Python	29

Sposób sortowania tabeli:

```
router.route('/fetchData/language_sex').get((req, res) => {
  const aggr = [
    { $group: { _id: { Q1: '$Q1', Q8: '$Q8' }, count: { $sum: 1 } } },
    { $sort: { count: -1 } },
    {
      $group: {
        _id: { Q1: '$_id.Q1' }, Q8: { $first: '$_id.Q8' },
        count: { $first: '$count' }
      }
    },
    { $sort: { _id: 1 } }
  ]

  Form.Forms.aggregate(aggr).exec((error, data) => {
    if (error) {
      return next(error)
    } else {
      res.json(data)
    }
  })
})
```

## Najpopularniejsze języki programowania w zależności od wieku i regionu świata

Most popular languages based on age in regions of the world

Export currentExport all

Region	Age	Language	Votes
Argentina	18-21	Python	1
Argentina	22-24	Python	7
Argentina	25-29	Python	10
Argentina	30-34	Python	6
Argentina	35-39	Java	4
Argentina	40-44	Python	6
Argentina	45-49	Python	4
Argentina	50-54	Python	2
Australia	18-21	Python	6
Australia	22-24	Python	11
Australia	25-29	Python	21
Australia	30-34	Python	27
Australia	35-39	Python	11
Australia	40-44	Python	8
Australia	45-49	C#/.NET	3
Australia	50-54	Python	7
Australia	55-59	Python	4
Australia	60-69	Visual Basic/VBA	2
Austria	18-21	Python	1
Austria	22-24	Python	2
Austria	25-29	R	4
Austria	30-34	Python	1
Austria	35-39	Python	2

Sposób sortowania tabeli:

```
router.route('/fetchData/language_regions-age').get((req, res) => {
  const aggr = [
    {$group: {_id: {Q3: '$Q3', Q2: '$Q2', Q8: '$Q8'}, count: {$sum: 1}}},
    {$sort: {count: -1}},
    {$group: {_id: {Q3: '$_id.Q3', Q2: '$_id.Q2'}, Q8: {$first: '$_id.Q8'},
    count: {$first: '$count'}}},
    {$sort: {_id: 1}}
  ]

  Form.Forms.aggregate(aggr).exec((error, data) => {
    if (error) {
      return next(error)
    } else {
      res.json(data)
    }
  })
})
```



# Wnioski

Zadaniem strony jest przedstawienie najpopularniejszych języków programowania w zależności od wieku, regionu świata, płci oraz wykształcenia. Dane zostały pobrane z darmowej strony kaggle.com z bazy o treści: **2018 Kaggle Machine Learning & Data Science Survey**. Zawartość bazy danych to odpowiedzi pobrane z formularza wypełnionego przez osoby z całego świata. Stworzony serwis internetowy pozwala na dodawanie swoich odpowiedzi w zakładce Formularz znajdujący się w prawym górnym rogu ekranu. Dane z wypełnionego formularza są przekazywane do serwera, który zapisuje dane do bazy danych MongoDB. Na stronie głównej prezentowane są dane pobrane z bazy danych. Wyświetlanymi danymi można manipulować za pomocą przycisku Select, który umożliwia wybranie rodzaju sortowania tabeli. Do strony dostęp mają autoryzowani użytkownicy. W celu przeglądania danych należy najpierw zarejestrować konto użytkownika, a następnie zalogować się na utworzone konto.

## Sposób uruchomienia projektu:

1. Uruchomienie kontenerów poleceniem docker-compose up  
# Z POWODU NIEWYJAŚNIONYCH PRZYCZYN I PROBLEMÓW PRZY PIERWSZYM URUCHOMIENIU PROJEKTU NALEŻY URUCHOMIĆ SKRYPT ZNAJDUJĄCY SIĘ NA KONTENERZE MONGO O NAZWIE IMPORT.SH:
2. Otworzyć nowy terminal
  - 2.1. Poleceniem docker ps -a sprawdzić nazwę kontenera MongoDB
  - 2.2. Uruchomić skrypt importujący dane poleceniem: *docker exec nazwa\_kontenera /import.sh*
3. Utworzyć nowe konto w celu używania serwisu.
4. Zalogować się do serwisu.

## Sposób dockeryzowania projektu:

1. Stworzenie plików Dockerfile dla usług MongoDB, Express, ReactJS
2. Stworzenie plików .dockerignore dla usług Express, ReactJS
3. Stworzenie pliku docker-compose.yaml