

Sprawozdanie nr 10

Zadanie 1

Połączenie z bazą danych utworzenie tabeli Samochody i wprowadzenie kilku przykładów

```
const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/carDB', {
  useNewUrlParser: true,
  useUnifiedTopology: true
});

const samSchema = new mongoose.Schema({
  model: String,
  marka: String,
  moc: Number,
  kolor: String,
  rokProdukcji: Number
});

const Samochod = mongoose.model('Samochod', samSchema);

const samochody = [
  { model: 'Model S', marka: 'Tesla', moc: 1020, kolor: 'Czarny',
rokProdukcji: 2021 },
  { model: 'Civic', marka: 'Honda', moc: 158, kolor: 'Niebieski',
rokProdukcji: 2020 },
  { model: '104', marka: 'Volvo', moc: 450, kolor: 'Granatowy',
rokProdukcji: 2001 },
  { model: 'A4', marka: 'Audi', moc: 261, kolor: 'Biały',
rokProdukcji: 2022 },
  { model: 'Multipla', marka: 'Daewo', moc: 90, kolor: 'Czerwony',
rokProdukcji: 1999 }
];

Samochod.insertMany(samochody)
  .then(() => {
```

```
    console.log('Samochody dodane do bazy danych.');
```

```
    mongoose.connection.close();
```

```
  })
```

```
  .catch(error => {
```

```
    console.error('Błąd podczas dodawania samochodów:', error);
```

```
  });
```

Zadanie 2

Na cel zadania utworzono dwa widoki marki.html i samochody.html

Pobieranie marek z bazy danych

```
app.get('/marki', async (req, res) => {
```

```
  try {
```

```
    const marki = await Samochod.distinct('marka');
```

```
    res.json(marki);
```

```
  } catch (error) {
```

```
    res.status(500).send('Error');
```

```
  }
```

```
});
```

Pobranie samochodów na podstawie marki z bazy danych

```
app.get('/samochody', async (req, res) => {
```

```
  const marka = req.query.marka;
```

```
  try {
```

```
    const cars = await Samochod.find({ marka: marka });
```

```
    res.json(samochody);
```

```
  } catch (error) {
```

```
    res.status(500).send('Error');
```

```
  }
```

```
});
```

Przesłanie rezultatów do plików html

```
app.get('/marki.html', (req, res) => {
```

```
  res.sendFile(path.join(__dirname, 'widoki', 'marki.html'));
```

```
});
```

```
app.get('/samochody.html', (req, res) => {
  res.sendFile(path.join(__dirname, 'widoki', 'samochody.html'));
});
```

Zadanie 3

Każdemu samochodowi na stronie dodano przycisk powodujący jego usunięcie z bazy danych. Dodano również funkcję dla przycisku wysyłającej żądanie DELETE do serwera i odświeżającej stronę po pomyślnym usunięciu.

Usunięcie samochodu z bazy danych

```
app.delete('/samochody/:id', async (req, res) => {
  const samId = req.params.id;
  try {
    await Samochod.findByIdAndDelete(samId);
    res.status(200).send('Pomyślnie usunięto samochód');
  } catch (error) {
    res.status(500).send('Error');
  }
});
```

Zadanie 4

Do widoku dodano formularz pozwalający na wybór samochodu według marki lub koloru.

Usunięcie samochodu na podstawie marki lub koloru

```
app.delete('/samochody', async (req, res) => {
  const { marka, kolor } = req.query;
  try {
    if (marka) {
      await Samochod.deleteMany({ marka });
      res.status(200).send('Samochód usunięto poprawnie');
    } else if (kolor) {
      await Samochod.deleteMany({ kolor });
      res.status(200).send('Samochód usunięto poprawnie');
    } else {
      res.status(400).send('Error zły kolor lub marka');
    }
  }
});
```

```
    }  
  } catch (error) {  
    res.status(500).send('Error');  
  }  
});
```