

Документация
Тема: Vote for meeting (Doodle)
Разработено от:
Мирослав Маджаров
Калин Лалов
Дамян Хинков

Използвани технологии:

Express – за поддържане на сървър

MySql2 – за работа с MySQL бд

crypto/dotenv – за хеширане и криптиране на данни

server.js: Създава Express сървър, който се свързва с база за данни базирана на MySql2 и предоставя крайни точки за връзка между клиентската и сървърната част на проекта.

```

require('dotenv').config();
const express = require('express');
const path = require('path');
const mysql = require('mysql2');
const bodyParser = require('body-parser');
const crypto = require('crypto');

const app = express();
const port = process.env.PORT || 5000;

const SECRET_SALT = process.env.SECRET_SALT;

app.use(bodyParser.json());
app.use(express.static(path.join(__dirname, '../client/public')));

const db = mysql.createConnection({
  host: process.env.DB_HOST,
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_NAME
});

```

Във .env файла се съхраняват данните за достъп до базата данни за по-добра сигурност чрез dotenv. Там е и солта, използвана за генериране на ИД-тата на събитията.

Предоставят се крайни точки към HTML файлове: /about, /profile, /created-events, /event и т.н.

База данни:

Таблица потребители съдържа информацията за профилите на организаторите.

В таблицата за събития се пазят данните, въвеждани при създаване на ново събитие от отделните организатори.

В таблицата за времеви интервали се съдържа информацията за часови диапазони на едно събитие и съответно колко на брой глас има за всеки часови диапазон.

В таблицата за гласувания се пази името на потребителя, който е гласувал. Създаден е тригър за увеличаване на броя на гласувания.

```
CREATE TABLE Users (  
    user_id INT AUTO_INCREMENT PRIMARY KEY,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    password VARCHAR(100) NOT NULL  
);
```

```
CREATE TABLE Meetings (  
    meeting_id INT AUTO_INCREMENT PRIMARY KEY,  
    title VARCHAR(50) NOT NULL UNIQUE,  
    description TEXT,  
    organizer_id INT NOT NULL,  
    start_date DATE,  
    end_date DATE,  
    start_range INT NOT NULL,  
    end_range INT NOT NULL,  
    FOREIGN KEY (organizer_id) REFERENCES Users(user_id)  
);
```

```
CREATE TABLE TimeSlots (  
    timeslot_id INT AUTO_INCREMENT PRIMARY KEY,  
    meeting_id INT NOT NULL,  
    date DATE NOT NULL,  
    start_time TIME NOT NULL,  
    votes_count INT DEFAULT 0,  
    FOREIGN KEY (meeting_id) REFERENCES Meetings(meeting_id)  
);
```

```
CREATE TABLE Votes (  
    votes_id INT AUTO_INCREMENT PRIMARY KEY,  
    timeslot_id INT NOT NULL,  
    voter_name VARCHAR(30) NOT NULL,  
    FOREIGN KEY (timeslot_id) REFERENCES TimeSlots(timeslot_id)  
);
```

```

CREATE TRIGGER increment_votes_count
AFTER INSERT ON Votes
FOR EACH ROW
BEGIN
    UPDATE TimeSlots
    SET votes_count = votes_count + 1
    WHERE timeslot_id = NEW.timeslot_id;
END//

```

