

```
//Importing required modules and libraries
const express = require('express');
const mysql = require('mysql');
const bodyParser = require('body-parser');
const app = express();

//Configure the database connection
const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'Roope123',
  database: 'opintorekisteri'
});

//Connecting to the database
connection.connect((err) => {
  if (err) throw err;
  console.log('Connected to MySQL Database');
});

//Middlewares
app.use(bodyParser.urlencoded({extended:
false}));
app.use(bodyParser.json());

//GET request to fetch all opiskeliija
app.get('/opiskeliija', (req, res) => {
  const sql = 'SELECT * FROM opiskeliija';
  connection.query(sql, (err, results) => {
    if (err) throw err;
    res.send(results);
  });
});

//GET request to fetch a specific student by id
app.get('/opiskeliija/:id', (req, res) => {
  const id = req.params.id;
  const sql = `SELECT * FROM opiskeliija WHERE`
```

## Tietokannat - Tehtävä 17

```
id = ${id}`;
    connection.query(sql, (err, result) => {
        if (err) throw err;
        res.send(result);
    });
});

//POST request to add a new student
app.post('/opiskelija', (req, res) => {
    const student = req.body;

    if (!student.name || !student.author) {
        console.log(student);
        console.log("Required info missing.");

        return;
    }

    const sql = 'INSERT INTO opiskelija (nimi,
tekija) VALUES ("' + student.name + '", ' +
student.author + ')';
    connection.query(sql, student, (err,
result) => {
        if (err) throw err;
        res.send('student added successfully');
    });
});

//PUT request to update a student
app.put('/opiskelija/:id', (req, res) => {
    const id = req.params.id;
    const student = req.body;

    if (!id || !student.name ||
!student.author) {
        console.log(student);
        console.log("Required info missing.");
    }
}
```

```
        return;
    }

    const sql = `UPDATE opiskeli SET nimi =
"${student.name}", tekija = ${student.author}
WHERE id = ${id}`;
    connection.query(sql, student, (err,
result) => {
        if (err) throw err;
        res.send('student updated
successfully');
    });
});

//DELETE request to delete a student
app.delete('/opiskeli/:id', (req, res) => {
    const id = req.params.id;
    const sql = `DELETE FROM opiskeli WHERE
id = ${id}`;
    connection.query(sql, (err, result) => {
        if (err) throw err;
        res.send('student deleted
successfully');
    });
});

//Starting the server
app.listen(3000, () => {
    console.log('Server started on port 3000');
});
```