

## Harjoitus 4 – MAX 5p

Jatketaan edellistä harjoitusta. Toteuta sovellus siten, että students tietorakennetta ei ole koodissa, vaan kaikki tieto käsitellään tietokannassa

- Tiedon näyttäminen, opiskelijanumeron mukaisessa järjestyksessä 1p
- Tiedon lisääminen, 1p
- Tiedon päivittäminen, 1p
- Tiedon poistaminen, 1p
- Lisäksi toteuta ominaisuus pääsivulle, jossa on käyttäjän on mahdollisuus valita (select -elementti) näytetäänkö "kaikki", "it" vai "art" opiskelijat. Filteröi hakua kyseisen valinnan mukaisesti. (1p)

```
<select id="degreeSelect" onchange="degreeSelected()">
  <option value="all">All</option>
  <option value="it">IT</option>
```

```
<script>
  function degreeSelected() {
    let select = document.getElementById("degreeSelect")
    let value = select.options[select.selectedIndex].value
    window.location.href = `/students?degree=${value}`;
```

```
@app.route("/students")
```

```
def getAll():
    degree="all"
    params = request.args
    # If you are using the database instead of using allstudents you should use
    # Student.objects (Student name depends how you named your model class)
    result = allstudents should
    if params:
        if params["degree"] and params["degree"] != "all":
            degree = params["degree"]
            result = list(filter(lambda s: s["degree"] == degree, result))

    return render_template("index.html", students=result, degree=degree)
```

(1p)