

# Assignment 1

---

- Group: 12
  - Kasper J. Hermansen, au557919, 201607110
  - Sameer A. Habibi, au543939, 201606697
- Class: ITWEB
- Date: 28-09-2019

## Table of contents

- [Assignment 1](#)
  - [Table of contents](#)
  - [Production url](#)
  - [Technological requirements](#)
  - [Purpose](#)
    - [The principles for using a MVC framework on a webserver](#)
    - [Design and implement a web site that include persistence of data in a database](#)
    - [Hosting the web application in cloud based hosting environments](#)
  - [Install & Run](#)
  - [Contribute](#)

## Production url

<https://itweb-g12-a1.herokuapp.com>

## Technological requirements

Node is used as a webserver, with express as an opinionated MVC framework. Pug is used as the templating engine and mongodb (using mongoose) as the database.

Heroku is used as the Cloud hosting provider and Mongodb Atlas as the distributed database provider.

## Purpose

### The principles for using a MVC framework on a webserver

The purpose of using MVC on a webserver is to separate logic, such as domain, dataaccess and presentation from each other.

The Model layer will contain the domain knowledge, and is the layer that interacts with the data persisted, or not.

The View layer handles the presentation logic, it can be html, json or any other format. The purpose of the format is just that it is separated from the business logic in the model and isn't tied to the actual handling of the view itself, such as decoding messages from clients or sending data to clients. The view layer should be stateless and only contain logic to enhance the presentation of the contents.

The Controller layer is the layer that binds it together, it is the client facing layer, and acts as an orchestrator in the fact that it acts as a delegating mechanic, which handles incoming requests, calls the model and returns the view, as an example. The controller is sometimes tied to the framework, and can be hard to test, as such it is often kept as thin as possible, to not pollute the Model layer with framework specific code. The Controller should in the best case only serve as an orchestrator, and have as little business logic as possible.

## Design and implement a web site that include persistence of data in a database

To handle the connection to mongodb (Mongo Atlas) in our case, mongoose is used, to provide ease of use, and give a higher level api. Locally a mongodb is spun up with docker to provide a local development environment. Remotely or on in the cloud (heroku) Mongodb Atlas is used, with their free tier sandbox model, which is more than enough for these exercises.

To spin up the local docker image simply run

```
./start-mongo.sh # linux / bash
```

on windows

```
docker run --rm --name mongo -p 27017:27017 mongo
```

optionally a local instance of mongodb can be spun up without docker, as long as it is exposed on localhost on port 27017

## Hosting the web application in cloud based hosting environments

Using heroku as a cloud provider, it is simple to use the service. Heroku provides an easy to use pipeline, that automatilly compiles a production build, and starts it, even without the use of docker.

Heroku does have some requirements. The builds scripts in `package.json` has to follow best practices, such as *build*, *start* etc.

Heroku also requires a specific variable in code to set the exposed port of the web server.

```
let port = process.env.PORT
if (port == null || port == '') {
  port = '8080'
}
```

Heroku will set the environment variable PORT at runtime, so the server should be able to change dynamically. Heroku will restart the server if neccessary.

## Install & Run

To use the this repo have npm and/or yarn installed and either docker, or mongodb started.

We use yarn, but npm should work just fine.

```
./start-mongo.sh # starts mongo container in the current terminal, open  
another for the rest of the walkthrough  
yarn # will get packages  
yarn dev # for development
```

Open a browser and checkout [localhost:8080](http://localhost:8080)

Should be a success, otherwise open an issue or whisper me at @kjuulh

## Contribute

To contribute simply ask @kjuulh for permission, or fork the repo and open a PR.

The website will automatically be updated when the master branch is updated.