

# Lab assignment 2

TI1506 2017/18; [ti1506-students.slack.com](https://ti1506-students.slack.com)

---

## Introduction

In the first part of this assignment, you will add client-side JavaScript code to your application to make it *interactive*. In the second part of this assignment you will practice with a relational database instance, using both a command line environment and a user friendly interface.

**Deadline:** when this assignment is due depends on the cluster your team is in.

**In order to pass** the assessment, you have to make a valid attempt at working through the lab assignment.

Students who are already well-versed in Web technology can also attempt to make their application available offline (using our AppCache approach).

If you are looking for an IDE we suggest [Visual Studio Code](#), a free & open-source dev. environment available for all major operating systems. It was originally designed to support node.js programmers (a server-side JavaScript framework we use in this class), but now has extensions for all major platforms. Best of all, Visual Studio Code itself is written in JavaScript.

---

# 1. OVERVIEW OF ACTIONS PLAN

Recall the “Step by step” guide of the JavaScript lecture:

## Step-by-step: making a responsive UI control

1. Pick a control (e.g. a button)

2. Pick an event (e.g. a click on a button)

3. Write a JavaScript function: what should happen when the event occurs? (e.g. a popup appears)

4. Attach the function to the event ON the control

Create such a step by step plan for *each* UI element of your Habit tracker application that will become interactive. Note: *this is a plan for what you will implement later, no actual code needs to be written here!* You must include the following interactive elements:

- Add a habit with a desired frequency (e.g. daily, every weekday, every hour)
- Delete a habit
- Update the status of a habit at a particular time
- Change an existing habit’s content (i.e. habit text and status)
- Sort the habits according to how well they are meeting the desired goals

Add one more action of your choice depending on your initial design (e.g. allow a user to tag a habit as “work”-related or “sports”-related).

It should be possible to use the habit tracker application with a keyboard alone or with a combination of keyboard and mouse (i.e. you need to implement actions for both types of

events). Ensure that the habit tracker app behaves as we would expect based on our own experience: for instance, we expect that once a habit item has been added to the list of habits, it should no longer be visible in the input element of the application.

**Deliverable:** a plan of actions

---

## 2. OBJECT-ORIENTED PROGRAMMING

Think about the design of your JavaScript code – which aspects of your action plan can you translate into the OO paradigm? For example, you might want to model each habit as an object. Similarly, you can also think about modeling the list of all habits as an object and so on. Decide on the use of at least one of the introduced JavaScript design patterns.

**Deliverable:** a description of how you will aim for your JavaScript code to be object-oriented.

---

## 3. WRITING CODE

Now that you have made your plan and decided on the use of OO principles, start coding! Implement your plan of actions one action at a time. Reduce the redundancy in the code as much as possible. Create as few global variables as possible.

Strive to achieve a **complete separation between content and interaction**: the JavaScript-based interaction functionality should not be embedded in your HTML files. Likewise, in the next assignment the styling information should not be embedded in your HTML files. You should separate JavaScript, CSS and HTML as much as possible. While this causes more work initially, it pays off in the long run: your code can be more easily debugged, it is more readable and more easily extendable.



Figure 1: Keeping things apart. Separate interaction, presentation and content.

**Deliverable:** the project files (HTML and JavaScript).

Note: do not have to incorporate style elements yet (CSS), we will cover the style in a later lecture. If you choose to incorporate CSS, be aware that the TAs will ignore the CSS during the assessment.

---

## 4. START MY SQL SERVER

Now it is time for you to have hands-on experience with a real-world relational database environment. The description of the assignment assumes you successfully installed a MySQL instance on your machine, using the instructions available at the following link:

[https://docs.google.com/document/d/1orxxhAZOMWhbUah8K83-LvFfmR4ihiwy\\_6i\\_gFXg2Y8](https://docs.google.com/document/d/1orxxhAZOMWhbUah8K83-LvFfmR4ihiwy_6i_gFXg2Y8)

With different versions of MySQL you might experience differences in the available commands.

In this exercise, you will get familiar with the command line instructions required to start/stop/restart an instance of the MySQL server.

**Deliverable:** this exercise has no deliverable, but its successful execution will allow you to perform the following exercises

---

## 5. CONNECT TO SQL SERVER VIA COMMAND LINE

This exercise and the following require you to use the command-line interface (CLI) of your operative system. The goal is to let you familiarize with the MySQL command line client, and explore a running instance of the MySQL server.

Import the database dump available here

[https://drive.google.com/file/d/1yG-zdXMy1Bk\\_br1I3ZOGKdTE5xJYAa4L](https://drive.google.com/file/d/1yG-zdXMy1Bk_br1I3ZOGKdTE5xJYAa4L)

using the same procedure used to import the IMDB database (see Brightspace, Lecture 3).

Use the MySQL client to connect to the `Habits` application database. Start your "conversation" with the database with

```
=====
```

```
$ mysql -u $USERNAME$ -p
```

```
=====
```

Use the following MySQL proprietary commands to check for the version and list of databases currently included in the MySQL server instance.

- `SELECT VERSION();`
- `SELECT now();`
- `HELP`
- `HELP Contents`
- `HELP Data Manipulation`
- `HELP SHOW DATABASES`
- `SHOW DATABASES;`

**Deliverable:** For each executed command, copy and paste in a textual file the returned response.

---

## 6. EXPLORE THE STRUCTURE OF THE Habits DATABASE

In this exercise you will start interacting with the `Habits` database, using the MySQL proprietary commands.

Use the following commands to explore the schema of the `Habits` database.

- `USE Habits`
- `SELECT DATABASE();`
- `SHOW TABLES;`

For each table in the database, use the `DESC` command (as exemplified below) to show the structure of each table, and write down the returned response.

```
DESC $TABLENAME$;
```

**Deliverable:** Given the response obtained by executing the previous commands, provide a *visual representation* (i.e. a drawing) of the schema of the `Habits` database. Adopt the visual notation used in the *Relational Model* section of the lecture slides to denote:

- Relations
- Attributes
- Primary Keys
- Foreign Keys and Integrity Constraints

You should be able to comment and motivate each element in the database schema.