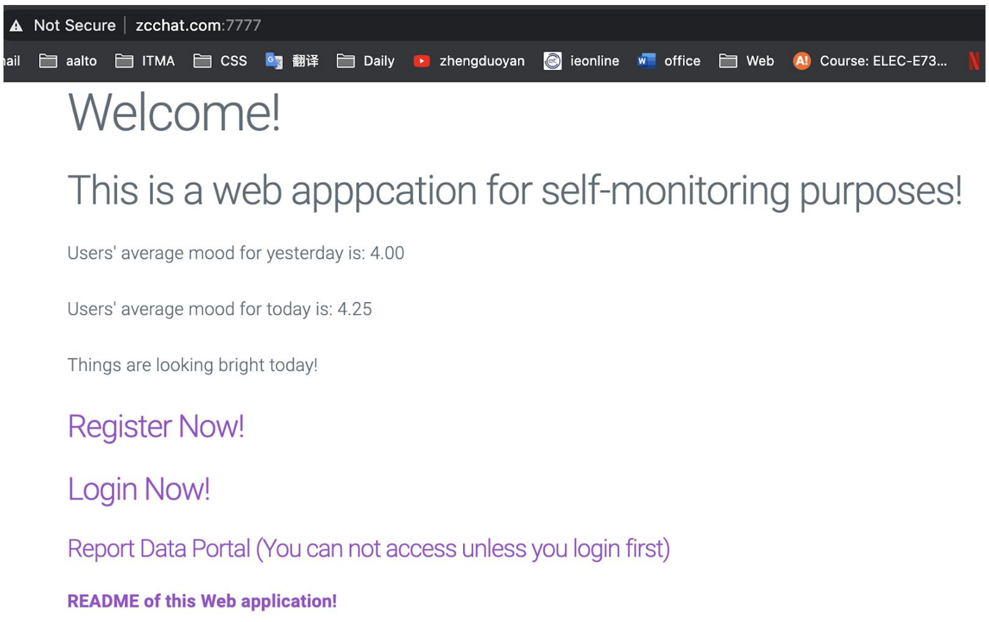
Web Software Development Course Project

0. Public address for this app website

http://www.zcchat.com:7777/
http://www.zcchat.com:7777/auth/registration
http://www.zcchat.com:7777/auth/login
http://www.zcchat.com:7777/api/summary
http://www.zcchat.com:7777/api/summary/2020/12/09 (1 example)
http://www.zcchat.com:7777/behavior/reporting
http://www.zcchat.com:7777/behavior/summary



Welcome!

This is a web apppcation for self-monitoring purposes!

Users' average mood for yesterday is: 4.00

Users' average mood for today is: 4.25

Things are looking bright today!

Register Now!

Login Now!

Report Data Portal (You can not access unless you login first)

README of this Web application!

1. Database Design

1.1 Elephantsql Authentication Info

hostname: "lallah.db.elephantsql.com",

```
  database: "ftkcjojp",
  user: "ftkcjojp",
  password: "tZtmt0lfnxfYul8OHs6qRXDDCt87XGg2",
  port: 5432
```

1.2 Tables Design

1.2.1 Table “users”

```
# Used for storing “Users” info, including “Email and password”
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  email VARCHAR(320) NOT NULL,
  password CHAR(60) NOT NULL
);
CREATE UNIQUE INDEX ON users((lower(email)));
```

1.2.2 Table “morning”

```
# Used for storing “morning” info of self-monitor application, including “email, date,
sleep_duration, sleep_quality and generic_mood”
CREATE TABLE morning (
    id SERIAL PRIMARY KEY,
    email VARCHAR(320) NOT NULL,
    date DATE NOT NULL,
    sleep_duration FLOAT NOT NULL,
    sleep_quality INTEGER NOT NULL,
    generic_mood INTEGER NOT NULL
);

# Example to insert data
INSERT INTO morning (email, date, sleep_duration,sleep_quality,generic_mood)
VALUES ('jing.yan@aalto.fi','2020-12-03',7,5,2);
```

1.2.3 Table “evening”

```
# Used for storing “evening” info of self-monitor application, including “email, date,
time_sports, time_study, eating_quality and generic_mood”
CREATE TABLE evening (
    id SERIAL PRIMARY KEY,
    email VARCHAR(320) NOT NULL,
    date DATE NOT NULL,
    time_sports FLOAT NOT NULL,
    time_study FLOAT NOT NULL,
    eating_quality INTEGER NOT NULL,
    generic_mood INTEGER NOT NULL
);
```

# Example to insert data
INSERT INTO evening (email, date, time_sports,time_study,eating_quality,generic_mood)
VALUES ('jing.yan@aalto.fi', '2020-12-01',1,12,5,1);
INSERT INTO evening (email, date, time_sports,time_study,eating_quality,generic_mood)
VALUES ('jing.yan@aalto.fi', '2020-12-02',2,4,5,5);
INSERT INTO evening (email, date, time_sports,time_study,eating_quality,generic_mood)
VALUES ('jing.yan@aalto.fi', '2020-12-03',3,4,5,5);

2. Function Description

2.1 Deno cmd to start the application

```
yanjing@yanjingdeMacBook-Pro final_yanj3 % deno run --allow-read --allow-net
--allow-env --unstable app.js
Check file:///Users/yanjing/Desktop/Web/final_20201211/app.js
```

2.2 File Tree

```
yanjing@yanjingdeMacBook-Pro final_20201211 % tree -L 10
.
├── app.js
├── config
│   └── config.js
├── database
│   └── database.js
├── deps.js
├── middlewares
│   └── middlewares.js
├── routes
│   ├── apis
│   │   ├── apisummaryApi.js
│   │   ├── authApi.js
│   │   ├── landingApi.js
│   │   ├── reportApi.js
│   │   ├── summaryApi.js
│   │   └── userApi.js
│   ├── controllers
│   │   ├── authController.js
│   │   ├── reportController.js
│   │   └── userController.js
│   └── routes.js
├── services
│   ├── apisummaryService.js
│   ├── authService.js
│   ├── landingService.js
│   ├── reportService.js
│   ├── summaryService.js
│   └── userService.js
└── views
    ├── auth.ejs
    ├── landing.ejs
    ├── partials
    │   ├── footer.ejs
    │   └── header.ejs
    ├── report.ejs
    ├── summary.ejs
    └── user.ejs

9 directories, 28 files
```

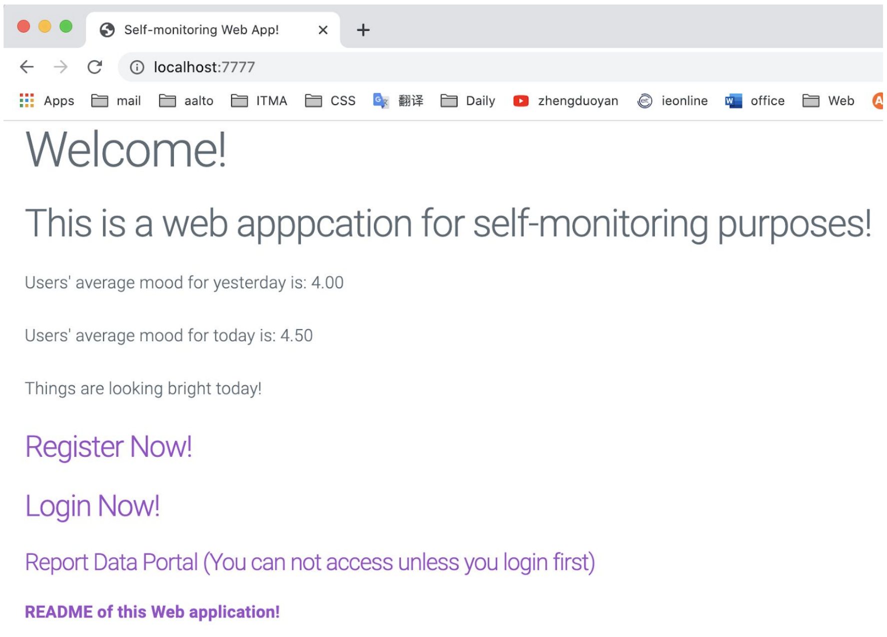
2.3 Landing Page (/)

2.3.0 Mark Done

Landing page (i.e. page at the root path of the application)

- Landing page briefly describes the purpose of the application - Done
- Landing page shows a glimpse at the data and indicates a trend - Done
    - Landing page shows users' average mood for today and and yesterday
    - If the average mood yesterday was better than today, tells that things are looking gloomy today
    - If the average mood yesterday was was worse today, tells that things are looking bright today
- Landing page has links / buttons for login and register functionality - Done
- Landing page has links / buttons for reporting functionality - Done

2.3.1 Browse the landing page



Self-monitoring Web App!

localhost:7777

Welcome!

This is a web apppcation for self-monitoring purposes!

Users' average mood for yesterday is: 4.00

Users' average mood for today is: 4.50

Things are looking bright today!

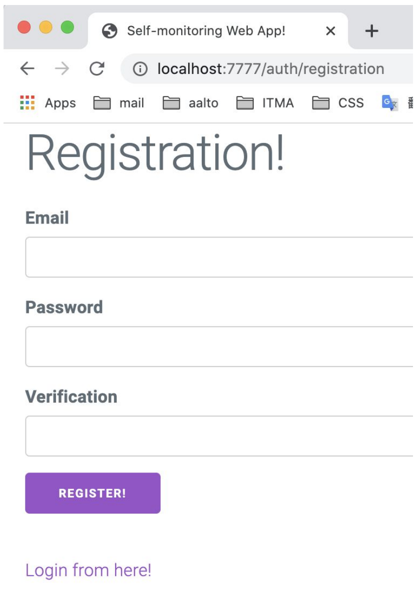
Register Now!

Login Now!

Report Data Portal (You can not access unless you login first)

README of this Web application!

2.3.2 Jump to register functionality by clicking "Register Now!"



Self-monitoring Web App!

localhost:7777/auth/registration

Registration!

Email

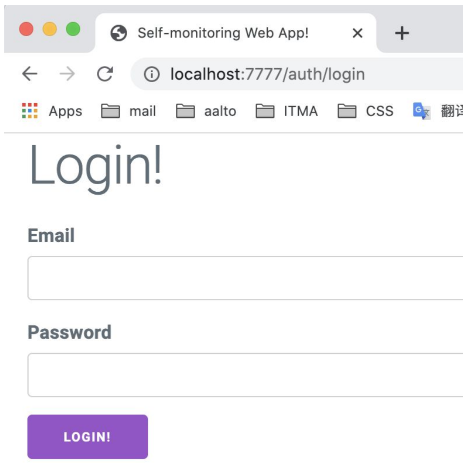
Password

Verification

REGISTER!

Login from here!

2.3.3 Jump to login functionality by clicking “Login Now!”



Login!

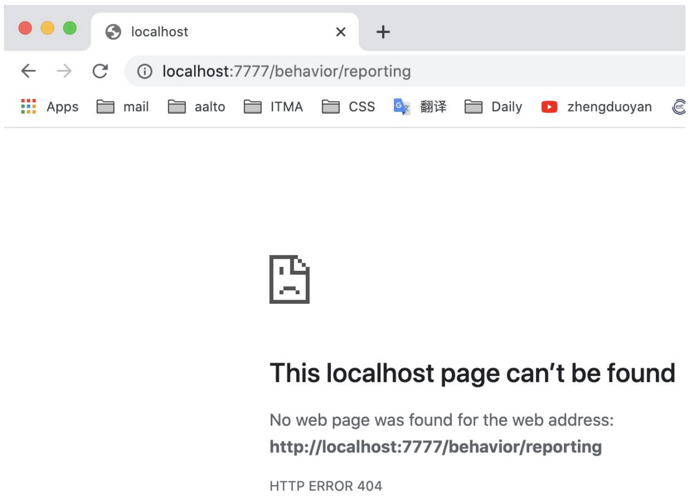
Email

Password

LOGIN!

2.3.4 Jump to reporting functionality by clicking “Report Data Portal (You can not access unless you login first)”

The following err is caused by no registered user logins!

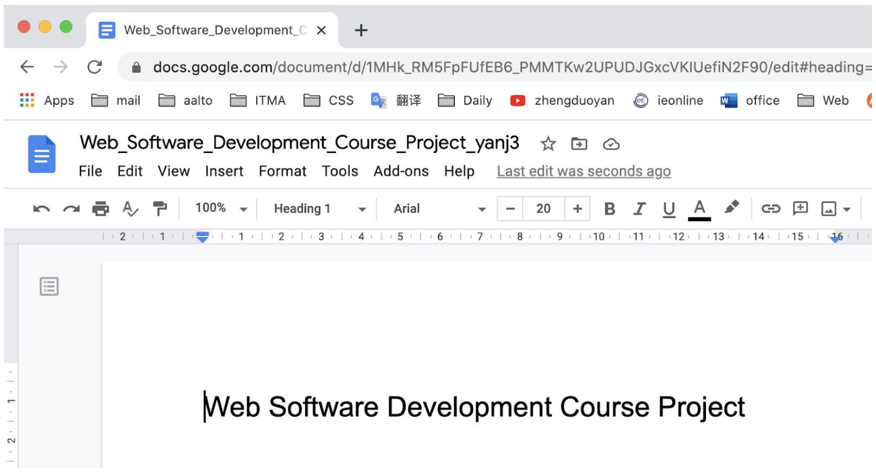


This localhost page can’t be found

No web page was found for the web address: **http://localhost:7777/behavior/reporting**

HTTP ERROR 404

2.3.5 Jump to README doc by clicking “README of this Web application!”


Web_Software_Development_C
docs.google.com/document/d/1MHk_RM5FpFUfEB6_PMMTKw2UPUDJGxcVKIUefiN2F90/edit#heading=
Apps
mail
aalto
ITMA
CSS
翻译
Daily
zhengduoyan
ieonline
office
Web
Web_Software_Development_Course_Project_yanj3
File
Edit
View
Insert
Format
Tools
Add-ons
Help
Last edit was seconds ago
100%
Heading 1
Arial
20

Web Software Development Course Project

2.4 Users’ Registration (/auth/registration)

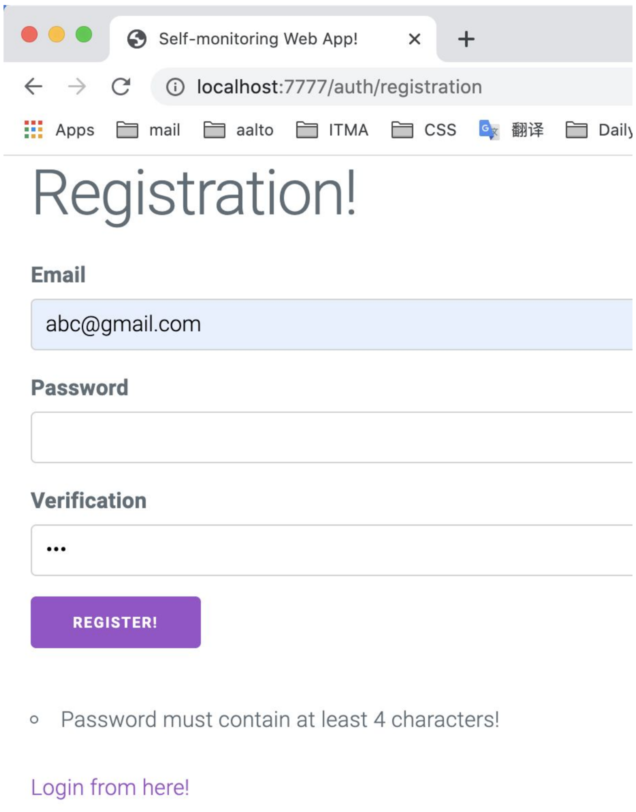
2.4.0 Mark Done

Users

- Email and password stored in the database for each user - Done
    - Password not stored in plaintext format
    - Emails must be unique (same email cannot be stored twice in the database)
- Users can register to the application - Done
- Registration form is accessible at /auth/registration - Done
    - Registration uses labels to clarify the purpose of the input fields
    - Registration form is validated on the server
        - Email must be a valid email (clarified from before, i.e. email must be validated - no need to e.g. send a mail to the address though)
        - Password must contain at least 4 characters
        - Validation errors shown on page
        - In case of validation errors, email field is populated (password is not)
- User-specific functionality is structured into logical parts (e.g. userController.js, userService.js) - Done

2.4.1 Register a user with invalid password (less than 4 digits)

Email: abc@gmail.com, Password: abc, Verification: abc


Self-monitoring Web App!

localhost:7777/auth/registration

Registration!

Email

abc@gmail.com

Password

Verification

•••

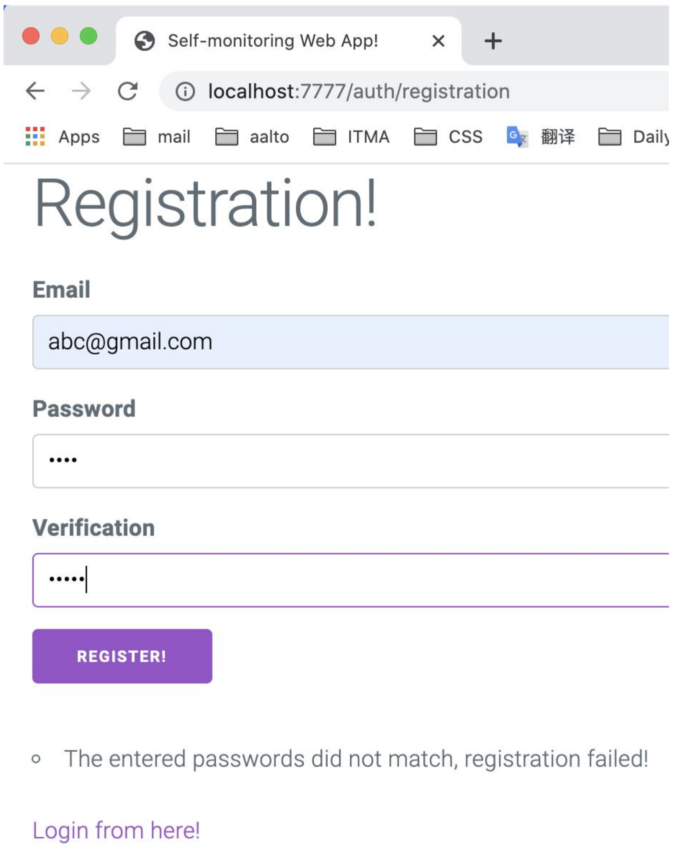
REGISTER!

- Password must contain at least 4 characters!

Login from here!

2.4.2 Register a user when password differs from verification

Email: abc@gmail.com, Password: abcd, Verification: abcd1


Self-monitoring Web App!

localhost:7777/auth/registration

Registration!

Email

abc@gmail.com

Password

••••

Verification

•••••

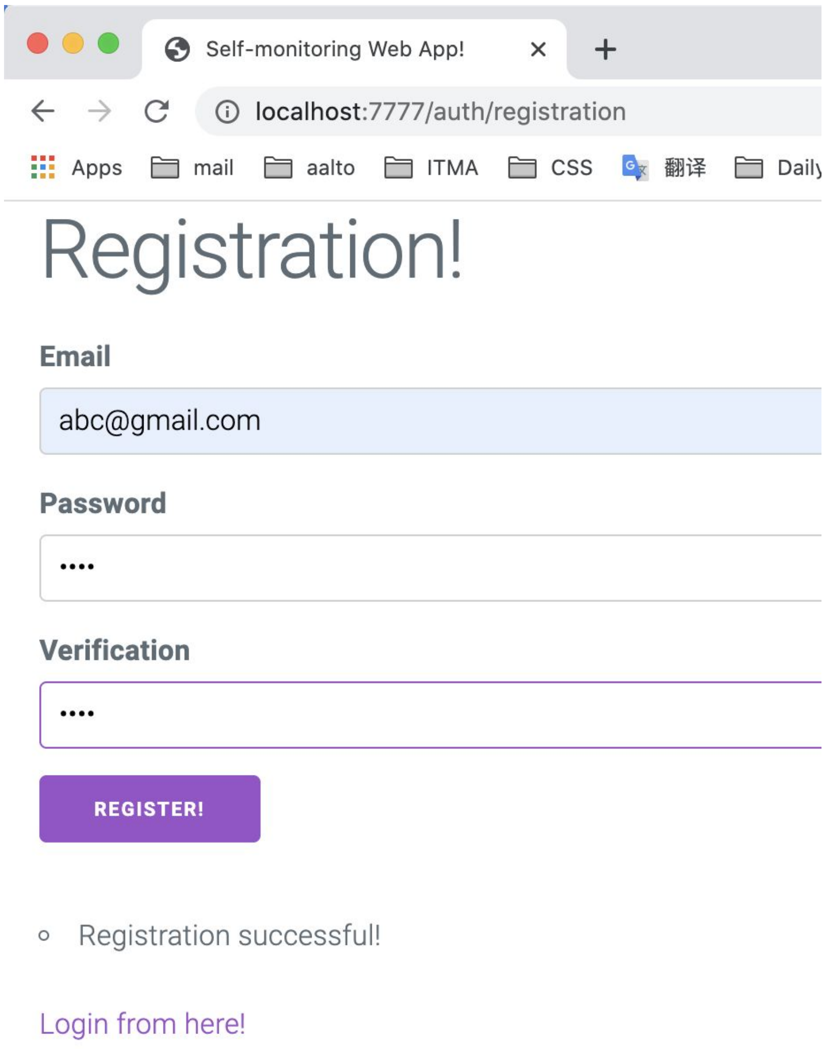
REGISTER!

- The entered passwords did not match, registration failed!

Login from here!

2.4.3 Register a user with valid info

Email: abc@gmail.com, Password: abcd, Verification: abcd



Registration!

Email

abc@gmail.com

Password

••••

Verification

••••

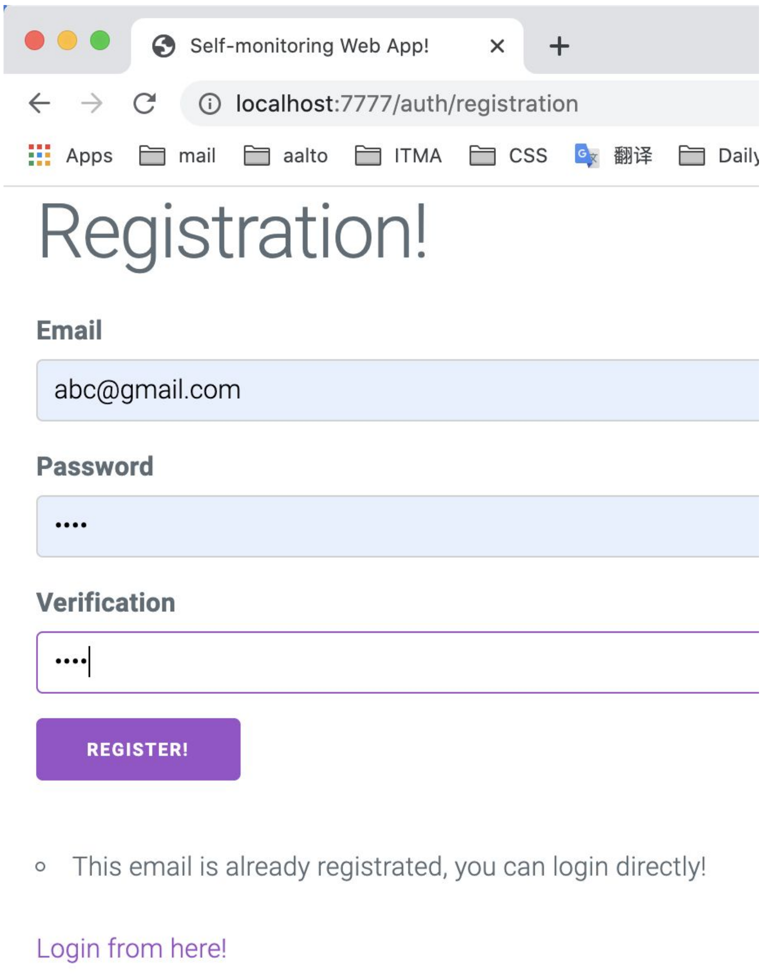
REGISTER!

- Registration successful!

Login from here!

2.4.4 Register a user twice

Email: abc@gmail.com, Password: abcd, Verification: abcd


Self-monitoring Web App!
localhost:7777/auth/registration
Apps mail aalto ITMA CSS 翻译 Daily

Registration!

Email

abc@gmail.com

Password

••••

Verification

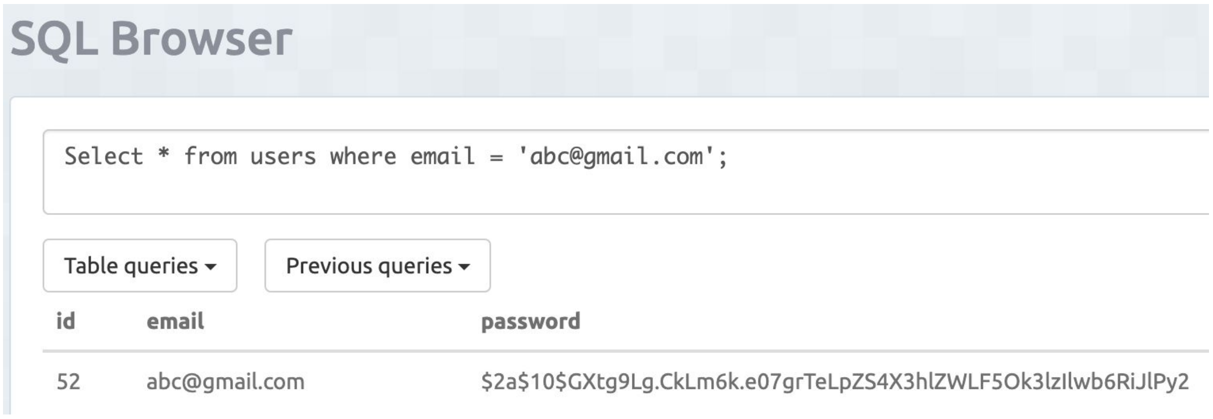
••••

REGISTER!

- This email is already registrated, you can login directly!

Login from here!

2.4.5 Check successfully registered user in database



```
Select * from users where email = 'abc@gmail.com';
```

id	email	password
52	abc@gmail.com	$2a$10$GXtg9Lg.CkLm6k.e07grTeLpZS4X3hlZWLF5Ok3lzIlwb6RiJlPy2

2.5 Users' Authentication (/auth/login, /auth/logout)

2.5.0 Mark Done

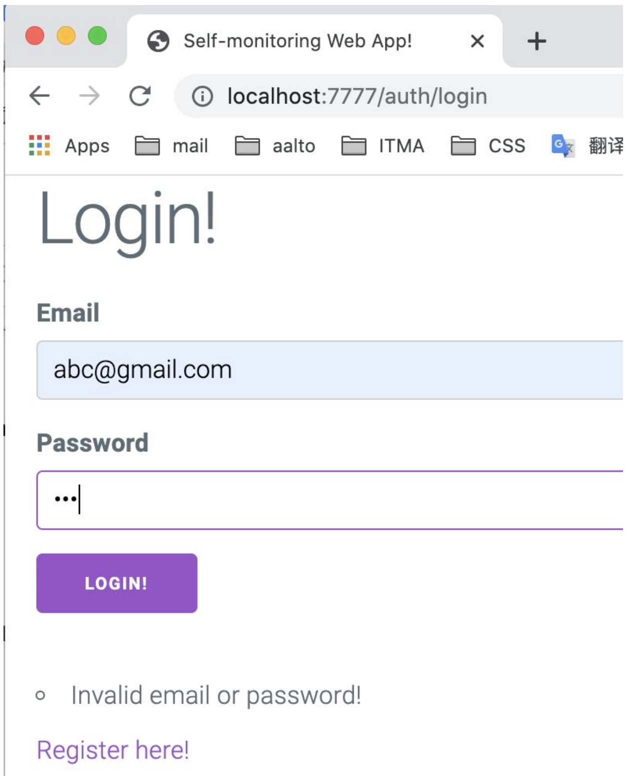
Authentication

- Application uses session-based authentication - Done
- Login form is accessible at /auth/login - Done

        - Login form asks for email and password
        - Login uses labels to clarify the purpose of the input fields
        - Login form has a link to the registration form
        - If the user types in an invalid email or password, a message "Invalid email or password" is shown on the login page.
            - Form fields are not populated
- Authentication functionality is structured into logical parts (e.g. authController.js or part of userController.js, ...). - Done
- Application has a logout button that allows the user to logout (logging out effectively means clearing the session) - Done
    - Logout functionality is at /auth/logout

2.5.1 Login with successfully registered email but with wrong password

Email: abc@gmail.com, Password: abc



Login!

Email

abc@gmail.com

Password

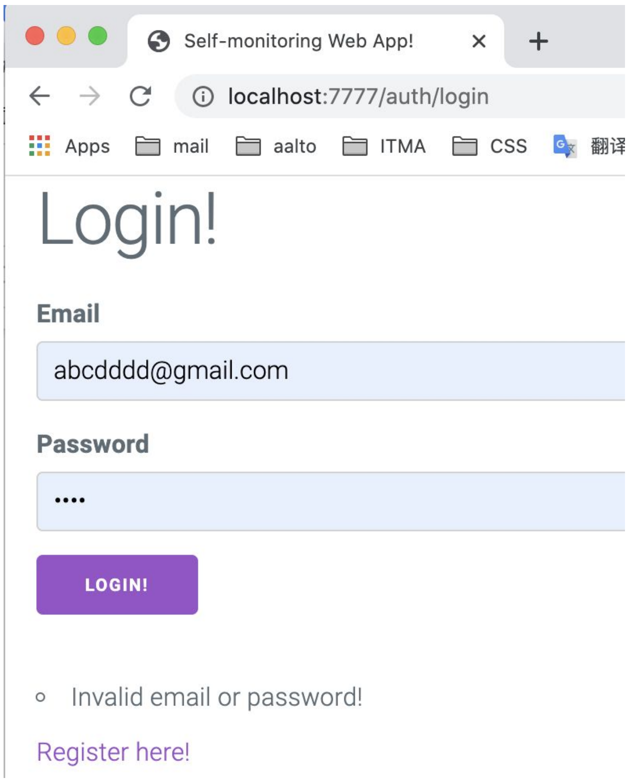
•••

- Invalid email or password!

Register here!

2.5.2 Login with non-registered email

Email: abcdddd@gmail.com, Password: abcd


Self-monitoring Web App!

localhost:7777/auth/login

Apps mail aalto ITMA CSS 翻译

Login!

Email

abcdddd@gmail.com

Password

••••

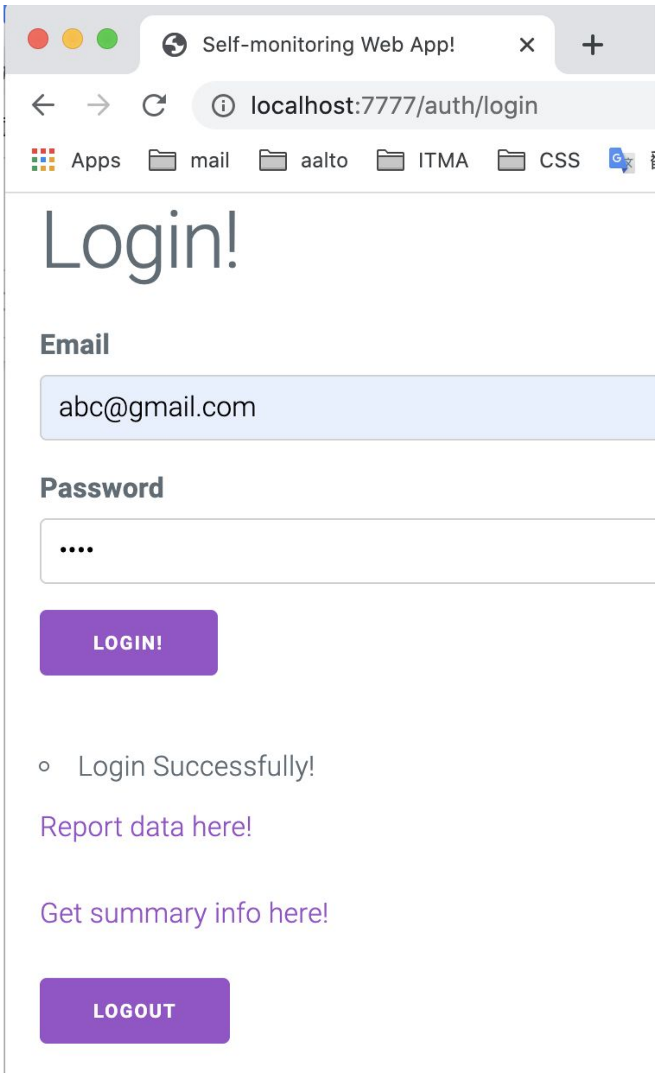
LOGIN!

- Invalid email or password!

Register here!

2.5.3 Login with valid user info

Email: abc@gmail.com, Password: abcd


Self-monitoring Web App!

localhost:7777/auth/login

Apps mail aalto ITMA CSS

Login!

Email

abc@gmail.com

Password

••••

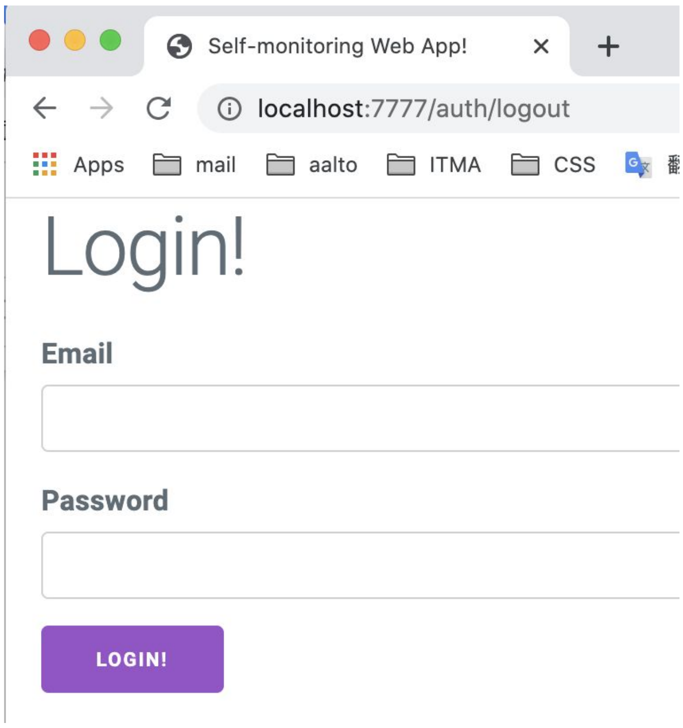
- Login Successfully!

Report data here!

Get summary info here!

2.5.4 Logout function

In “2.5.3 Login with valid user info”, after a user logged successfully, then the webUI will show the link to “/behavior/reporting” and “/behavior/summary” resources, and “logout” function. When clicking the “logout” button, it will jump back to the “/auth/login” website.


Self-monitoring Web App!

localhost:7777/auth/logout

Apps mail aalto ITMA CSS

Login!

Email

Password

LOGIN!

2.6 Report Data (/behavior/reporting)

2.6.0 Mark Done

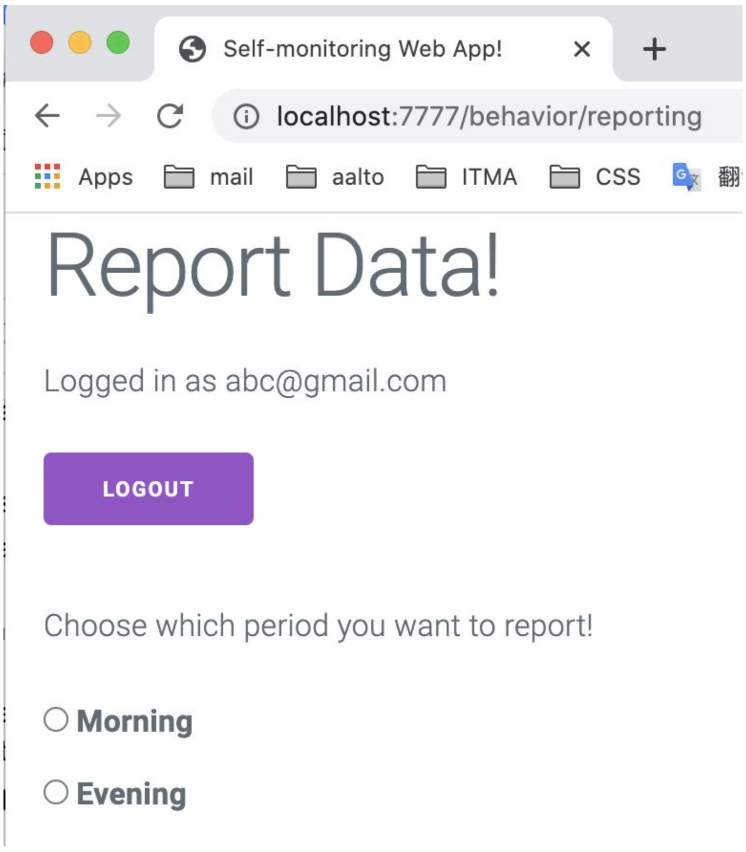
Reporting

- Reporting functionality is available under the path /behavior/reporting - Done
- Reporting cannot be done if the user is not authenticated - Done
- When accessing /behavior/reporting, user can choose whether morning or evening is being reported - Done
    - User reporting form depends on selection
    - Page at /behavior/reporting shows whether morning and/or evening reporting for today has already been done
- Morning reporting form contains fields for date, sleep duration, sleep quality, and generic mood - Done
    - Date is populated by default to today, but can be changed
        - Form has a date field for selecting the date
    - Sleep duration is reported in hours (with decimals)
    - Sleep quality and generic mood are reported using a number from 1 to 5, where 1 corresponds to very poor and 5 corresponds to excellent.
        - Form has a slider (e.g. range) or radio buttons for reporting the value
    - Form contains labels that clarify the purpose of the input fields and the accepted values
    - Form fields are validated

        - Sleep duration must be entered, must be a number (can be decimal), and cannot be negative
        - Sleep quality and generic mood must be reported using numbers between 1 and 5 (integers).
        - In case of validation errors, form fields are populated
- Evening reporting form contains fields for date, time spent on sports and exercise, time spent studying, regularity and quality of eating, and generic mood - Done
    - Date is populated by default to today, but can be changed
        - Form has a date field for selecting the date
    - Time spent on sports and exercise and time spent studying are reported in hours (with decimals)
    - Regularity and quality of eating and generic mood are reported using a number from 1 to 5, where 1 corresponds to very poor and 5 corresponds to excellent.
        - Form has a slider (e.g. range) or radio buttons for reporting the value
    - Form contains labels that clarify the purpose of the input fields and the accepted values
    - Form fields are validated
        - Time spent on sports and exercise and time spent studying are reported in hours must be entered, must be a number (can be decimal), and cannot be negative
        - Regularity and quality of eating and generic mood must be reported using numbers between 1 and 5 (integers).
        - In case of validation errors, form fields are populated
- Reported values are stored into the database - Done
    - The database schema used for reporting works for the task
    - Reporting is user-specific (all reported values are stored under the currently authenticated user)
    - If the same report is already given (e.g. morning report for a specific day), then the older report is removed
        - If the functionality for handling duplicate reports is something else, the functionality is described in documentation
- Reporting functionality structured into logical parts (separate views folder, separate controller for reporting, service(s), ...) - Done

2.6.1 Report Data

In "2.5.3 Login with valid user info", after a user logged successfully, then the webUI will show the link to "/behavior/reporting" and "/behavior/summary" resources, and "logout" function. When clicking "Report data here!", the user can see the following web UI.



Report Data!

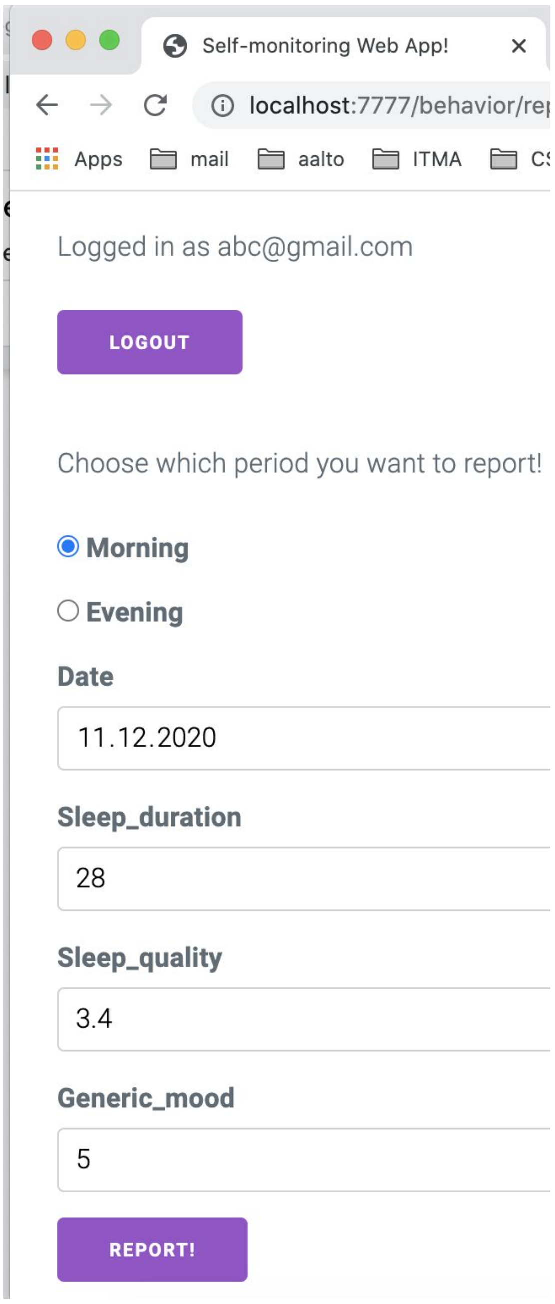
Logged in as abc@gmail.com

Choose which period you want to report!

2.6.1.1 Report Morning Data

In “2.6.1 Report Data”, clicking the “Morning” button to report morning data.

2.6.1.1.1 Report wrong data



Self-monitoring Web App!

localhost:7777/behavior/rep

Apps mail aalto ITMA CS

Logged in as abc@gmail.com

LOGOUT

Choose which period you want to report!

(● Morning) ○ Evening

Date

11.12.2020

Sleep_duration

28

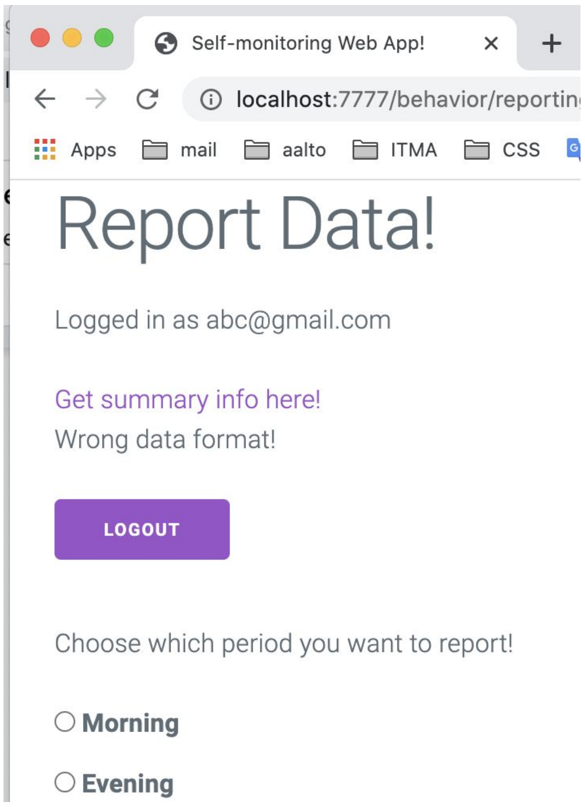
Sleep_quality

3.4

Generic_mood

5

REPORT!



Self-monitoring Web App!

localhost:7777/behavior/reportin

Apps mail aalto ITMA CSS

Report Data!

Logged in as abc@gmail.com

Get summary info here!

Wrong data format!

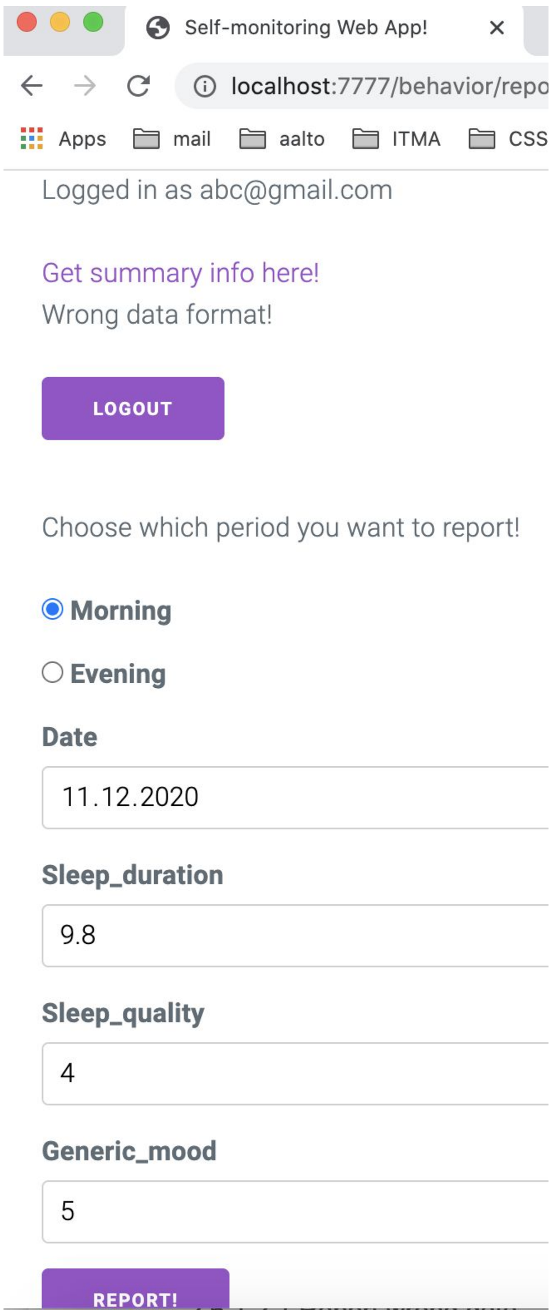
LOGOUT

Choose which period you want to report!

○ Morning

○ Evening

2.6.1.1.2 Report right data and check the record in database



Self-monitoring Web App!

localhost:7777/behavior/repo

Apps mail aalto ITMA CSS

Logged in as abc@gmail.com

Get summary info here!

Wrong data format!

LOGOUT

Choose which period you want to report!

(●) Morning

( ) Evening

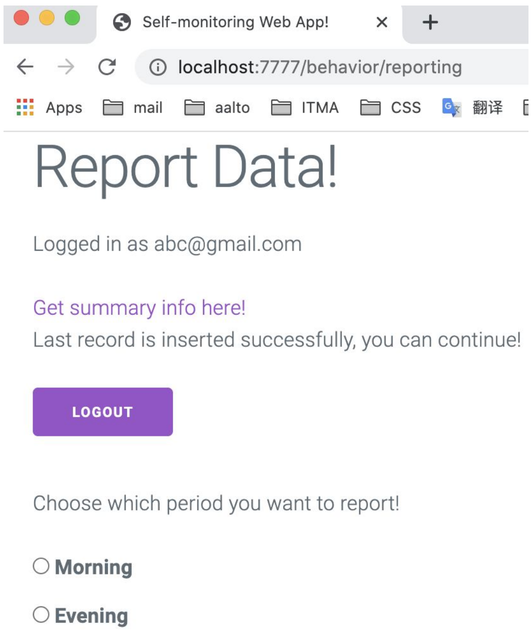
Date: 11.12.2020

Sleep_duration: 9.8

Sleep_quality: 4

Generic_mood: 5

REPORT!



Self-monitoring Web App!

localhost:7777/behavior/reporting

Apps mail aalto ITMA CSS 翻译

Report Data!

Logged in as abc@gmail.com

Get summary info here!

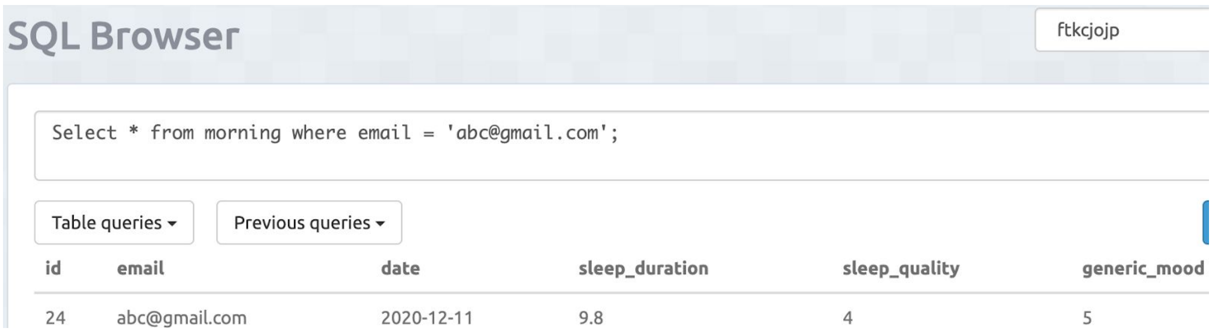
Last record is inserted successfully, you can continue!

LOGOUT

Choose which period you want to report!

( ) Morning

( ) Evening



SQL Browser

ftkcjojp

```
Select * from morning where email = 'abc@gmail.com';
```

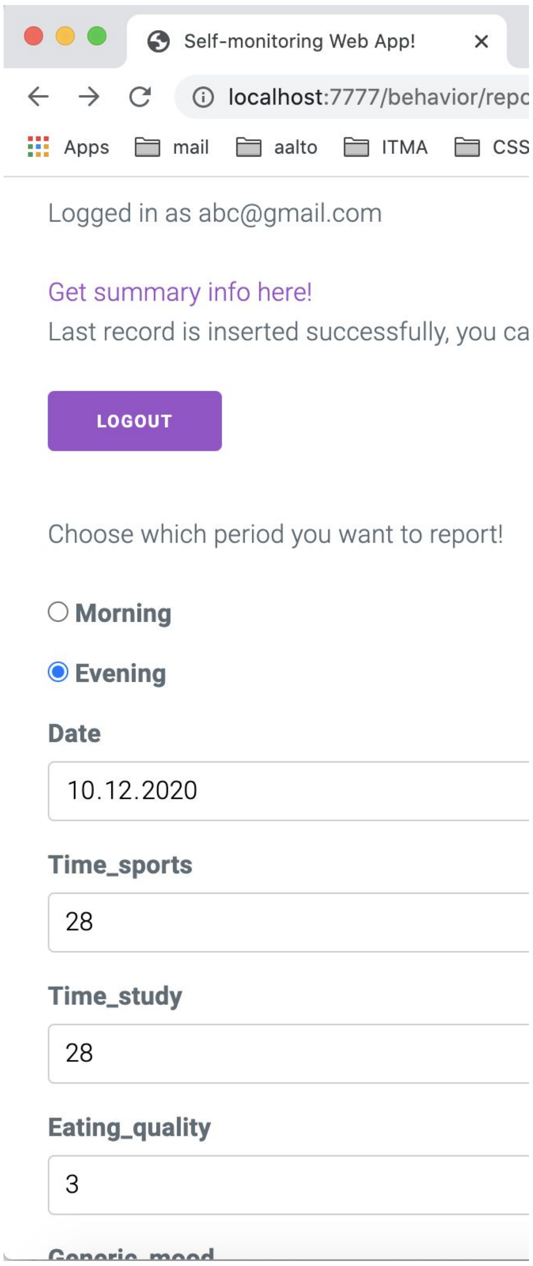
Table queries ▾ Previous queries ▾

id	email	date	sleep_duration	sleep_quality	generic_mood
24	abc@gmail.com	2020-12-11	9.8	4	5

2.6.1.2 Report Evening Data

In “2.6.1 Report Data”, clicking the “Evening” button to report evening data.

2.6.1.2.1 Report wrong data



Self-monitoring Web App!

localhost:7777/behavior/repo

Logged in as abc@gmail.com

Get summary info here!

Last record is inserted successfully, you ca

LOGOUT

Choose which period you want to report!

Morning [✓ Evening]

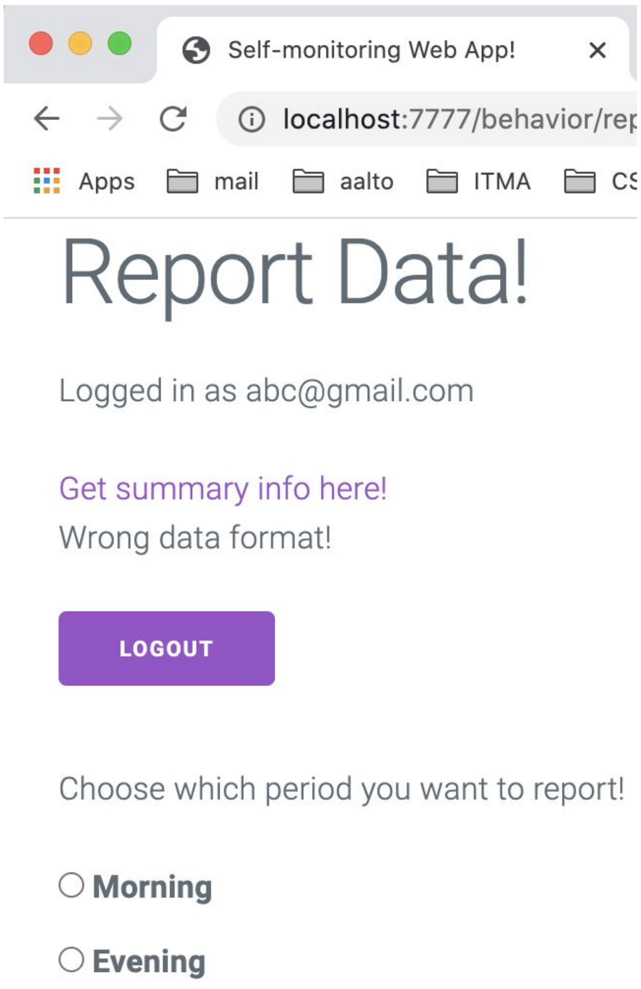
Date: 10.12.2020

Time_sports: 28

Time_study: 28

Eating_quality: 3

Generic_mood



Self-monitoring Web App!

localhost:7777/behavior/rep

Report Data!

Logged in as abc@gmail.com

Get summary info here!

Wrong data format!

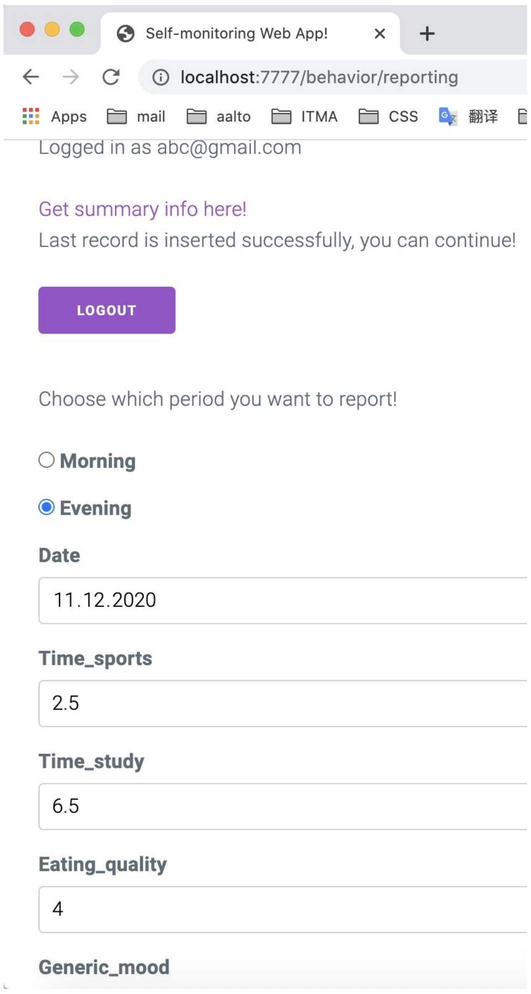
LOGOUT

Choose which period you want to report!

Morning

Evening

2.6.1.2.2 Report right data and check the record in database


Self-monitoring Web App!

localhost:7777/behavior/reporting

Logged in as abc@gmail.com

Get summary info here!

Last record is inserted successfully, you can continue!

LOGOUT

Choose which period you want to report!

○ Morning

● Evening

Date

11.12.2020

Time_sports

2.5

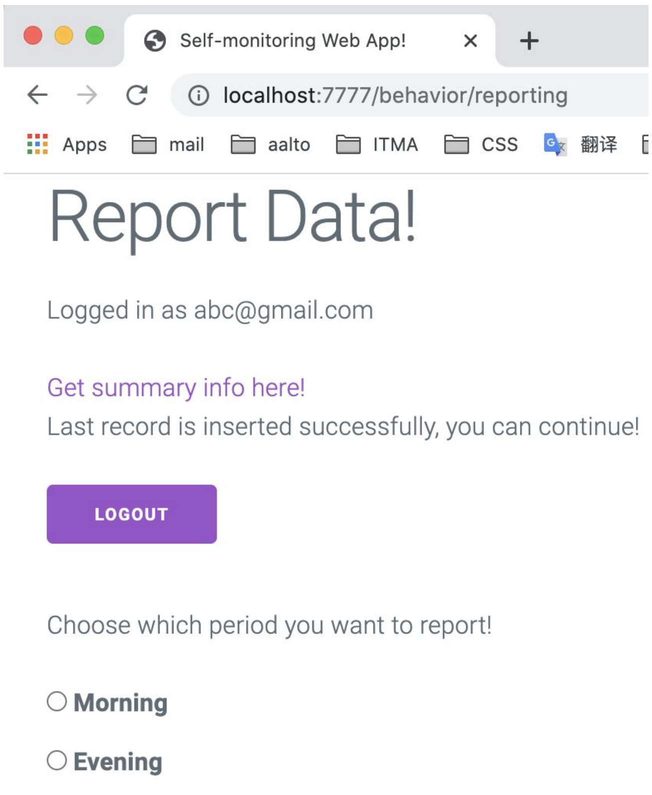
Time_study

6.5

Eating_quality

4

Generic_mood


Self-monitoring Web App!

localhost:7777/behavior/reporting

Report Data!

Logged in as abc@gmail.com

Get summary info here!

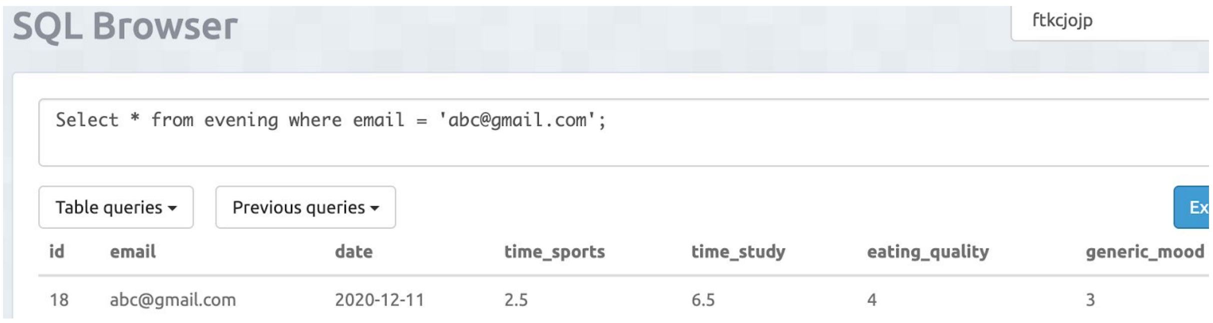
Last record is inserted successfully, you can continue!

LOGOUT

Choose which period you want to report!

○ Morning

○ Evening


SQL Browser

```
Select * from evening where email = 'abc@gmail.com';
```

id	email	date	time_sports	time_study	eating_quality	generic_mood
18	abc@gmail.com	2020-12-11	2.5	6.5	4	3

2.6.2 Logout Function

Which is same with “2.5.4 Logout function”

2.7 Summary (/behavior/summary)

2.7.0 Mark Done

Summarization

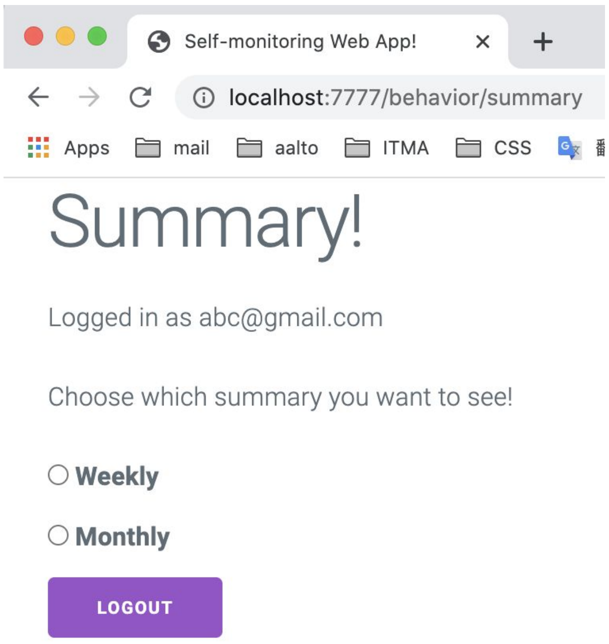
- Summary functionality is available under the path /behavior/summary - Done

- Main summary page contains the following statistics, by default shown for the last week and month - Done
    - Weekly average (by default from last week)
        - Average sleep duration
        - Average time spent on sports and exercise
        - Average time spent studying
        - Average sleep quality
        - Average generic mood
    - Monthly average (by default from last month)
        - Average sleep duration
        - Average time spent on sports and exercise
        - Average time spent studying
        - Average sleep quality
        - Average generic mood
- Summary page has a selector for week and month. Check input type="week" and input type="month". - Done
    - When the week is changed, the weekly average will be shown for the given week.
    - When the month is changed, the monthly average will be shown for the given month.
    - If no data for the given week exists, the weekly summary shows text suggesting that no data for the given week exists.
    - If no data for the given month exists, the monthly summary shows text suggesting that no data for the given month exists.
- Summary data / averages calculated within the database - Done
    - When doing weekly reporting, the weekly averages are calculated in the database
    - When doing monthly reporting, the monthly averages are calculated in the database
- Summarization page contains statistics only for the current user. - Done

2.7.1 Summary

In “2.5.3 Login with valid user info”, after a user logged successfully, then the webUI will show the link to “/behavior/reporting” and “/behavior/summary” resources, and “logout” function. When clicking “Get summary info here!”, the user can see the summary info.

In “2.6.1 Report Data”, after a user logged and checked the report data portal, this web also provides the link for the user to see summary info by clicking “Get summary info here!”.


Summary!

Logged in as abc@gmail.com

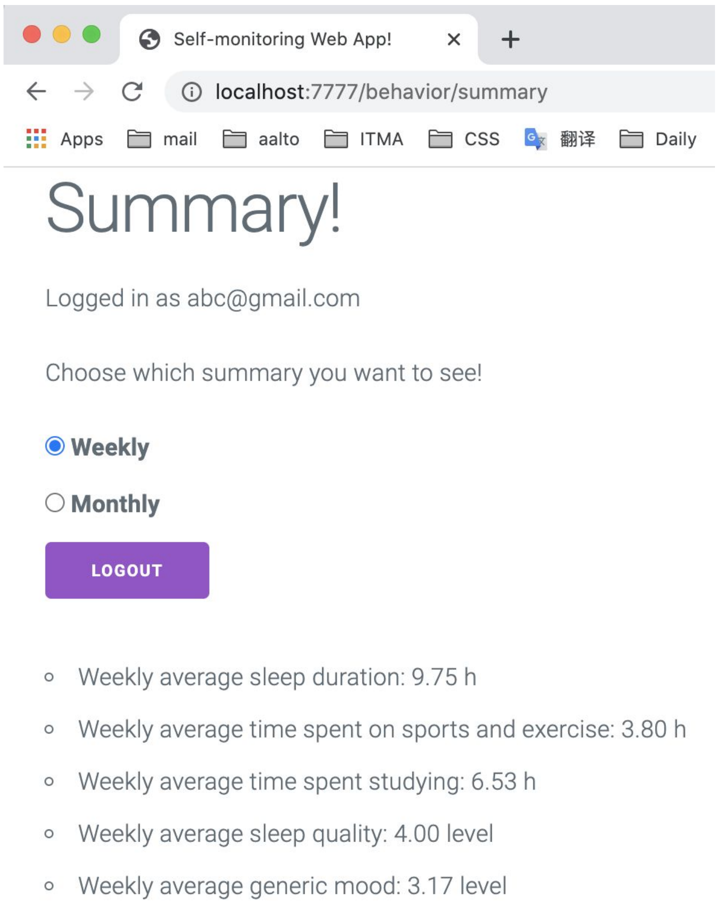
Choose which summary you want to see!

○ Weekly

○ Monthly

2.7.1.1 Weekly average Summary

By clicking “Weekly”, the user can see the following info:


Summary!

Logged in as abc@gmail.com

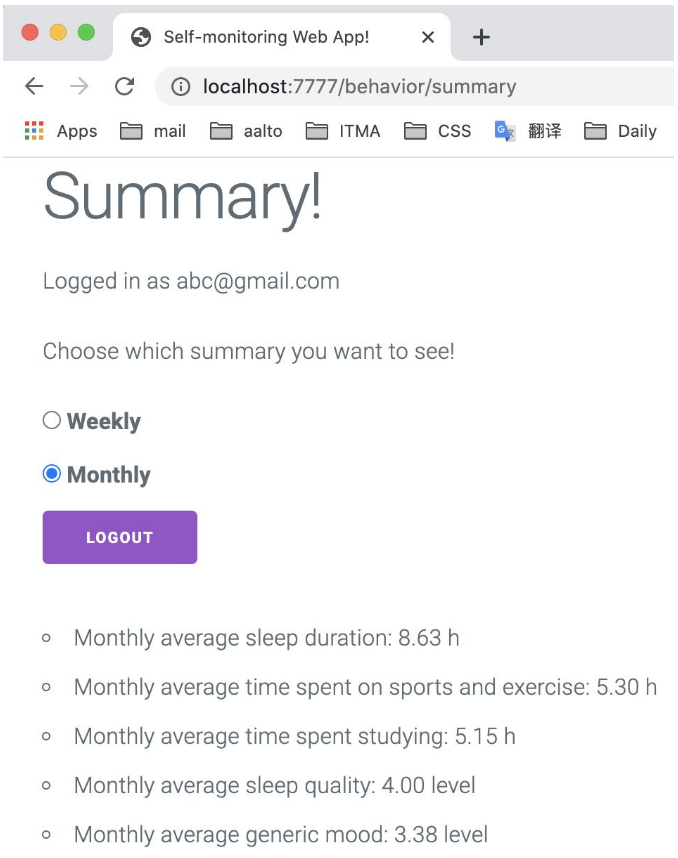
Choose which summary you want to see!

● Weekly

○ Monthly

- Weekly average sleep duration: 9.75 h
- Weekly average time spent on sports and exercise: 3.80 h
- Weekly average time spent studying: 6.53 h
- Weekly average sleep quality: 4.00 level
- Weekly average generic mood: 3.17 level

2.7.1.2 Monthly average Summary



Summary!

Logged in as abc@gmail.com

Choose which summary you want to see!

( ) Weekly

[✓ Monthly]

LOGOUT

- Monthly average sleep duration: 8.63 h
- Monthly average time spent on sports and exercise: 5.30 h
- Monthly average time spent studying: 5.15 h
- Monthly average sleep quality: 4.00 level
- Monthly average generic mood: 3.38 level

2.7.2 Logout Function

Which is same with “2.5.4 Logout function”

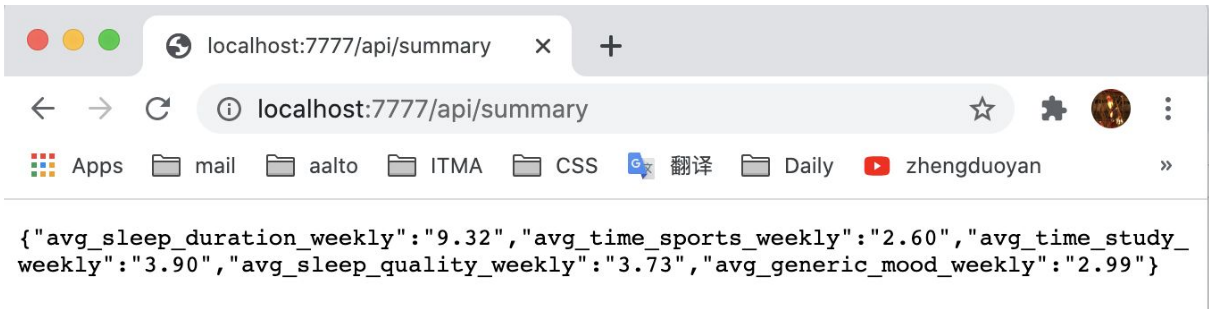
2.8 APIs

2.8.0 Mark Done

APIs

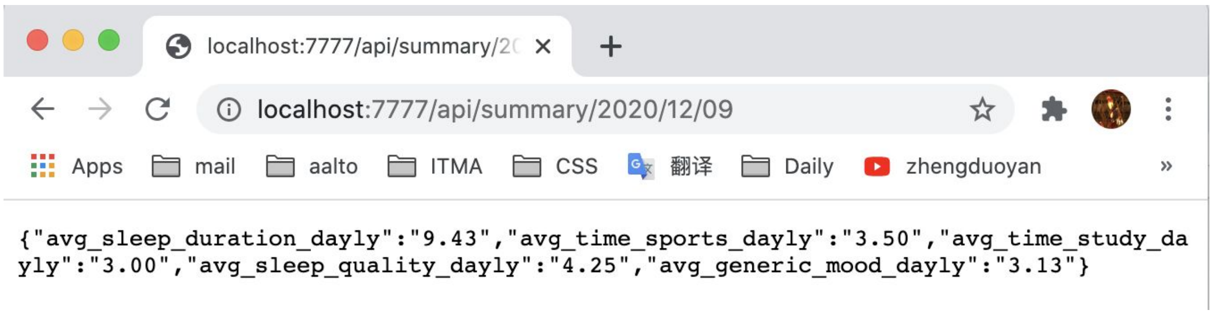
- The application provides an API endpoint for retrieving summary data generated over all users in a JSON format
- The API is accessible by all
- The API allows cross-origin requests
- Endpoint /api/summary provides a JSON document with sleep duration, time spent on sports and exercise, time spent studying, sleep quality, and generic mood averaged over the last 7 days
- Endpoint /api/summary/:year/:month/:day provides a JSON document with averages for sleep duration, time spent on sports and exercise, time spent studying, sleep quality, and generic mood for the given day

2.8.1 API summary over the last 7 da



```
{"avg_sleep_duration_weekly":"9.32","avg_time_sports_weekly":"2.60","avg_time_study_
weekly":"3.90","avg_sleep_quality_weekly":"3.73","avg_generic_mood_weekly":"2.99"}
```

2.8.2 API summary for a given day


localhost:7777/api/summary/2(
localhost:7777/api/summary/2020/12/09
Apps
mail
aalto
ITMA
CSS
翻译
Daily
zhengduoyan

```
{"avg_sleep_duration_dayly":"9.43","avg_time_sports_dayly":"3.50","avg_time_study_da
yly":"3.00","avg_sleep_quality_dayly":"4.25","avg_generic_mood_dayly":"3.13"}
```