

Programme <b>Applied Computer Science</b>	Trajectory <b>AIBP / CCCP / IBC / SSE</b>
Module <b>Web, Mobile and Security</b>	
Lecturer <b>Frédéric Vlummens</b>	Verified by <b>N/A</b>
Academic year <b>2019 – 2020</b>	Semester <b>2</b>
Date and time <b>06/05/2020, 13u30 – 17u30</b>	Period <b>N/A</b>
<b>MOCK EXAM “WEB AND MOBILE”</b>	<b>RESULT /20</b>

## SOURCE FILES

All source files (images, SQL dump, ...) can be found on Leho via <https://leho-howest.instructure.com/courses/6943> > **A Web and Mobile > 98 Mock exam.**

## LOCAL FILE STRUCTURE

Create a new Laravel project on your virtual machine with the name `purrr-palace-firstnamelastname`, replacing `firstname` and `lastname` by your actual `firstname` and `lastname`. Do not use any spaces or special characters (accented letters, punctuation, ...) in your project name.

Example: Frédéric Vlummens → **purrr-palace-fredericvlummens**

**Before continuing, make sure to modify your .gitignore file so that the .env file is also uploaded to the Gitlab server. To do so, remove the .env entry from your .gitignore file.**

## IMPORTANT

Every form on (online) communication between students or with other parties is strictly forbidden.

This is an individual assignment. Every observation of irregularity (a.o. cell phone, cheating, copying, hacking, usage of social media clients, ...) will conforming to the education and exam regulations (*OER*) result in notification of both the student(s) involved and the chairperson of the exam board.

After the exam has been completed, it is forbidden to publish the assignment or solution in any form whatsoever.

## ALLOWED SOURCES

You are allowed to use your own class notes, slides, books, syllabi and other materials, including the internet. Communication with fellow students or third parties is strictly forbidden (see above). Make sure that all applications that might popup/launch automatically of may be interpreted as an attempt of fraud (Outlook, Messenger, Facebook, ...) are closed!

## ELECTRONIC MONITORING

**Since this is an online exam, you are required to activate HOSA (Howest Screenshot Application) for the entire duration of the exam. Failure to comply with this requirement or terminating HOSA prematurely will be interpreted as an irregularity.**

HOSA can be found on <https://hosa.toegepasteinformatica.be/>.

Before you start working on your exam, launch the HOSA script, enter your login username (without @student.howest.be suffix) and the exam duration, which is 4 hours, therefore 240 minutes.

## HANDING IN

---

Hand in your complete project using op-gitlab. Create an individual repository under your personal Web and Mobile group with the name **99-mock-exam-purr-palace**.

**Perform a first commit and push with the following text:**

**I, *lastname firstname*, hereby declare that I will activate HOSA during the entire duration of the exam. I have read the assignment and know what the consequences are of any form of irregularity.**

(replace *lastname firstname* by your own lastname and firstname)

Make sure you commit on remote master, as this is the branch that will be reviewed by the lecturers!

**You are required to commit and push on a regular basis:**

- At least one commit every 20 minutes
- At least one push per hour

**Your final commit and push will be with the following text:**

**I, *lastname firstname*, hereby declare that this is my final version for the exam and that I can no longer modify this.**

(again, replace *lastname firstname* by your own lastname and firstname).

Finally, **zip the folder with screenshots created by HOSA and upload** this file through the corresponding assignment in Leho: <https://leho-howest.instructure.com/courses/6943/assignments/35222>.

**Failure to comply with the requirements above will be interpreted as an irregularity.**

## EXAM ASSIGNMENT

### 1. INTRODUCTION

You have been hired by the Purr Palace cat hotel to build a responsive server-side website which allows cat owners to book an arrangement for their feline friends.

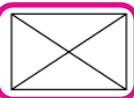
### 2. LAYOUT

The graphics department has created the wireframes for you for two form factors: desktop and smartphone in portrait mode.

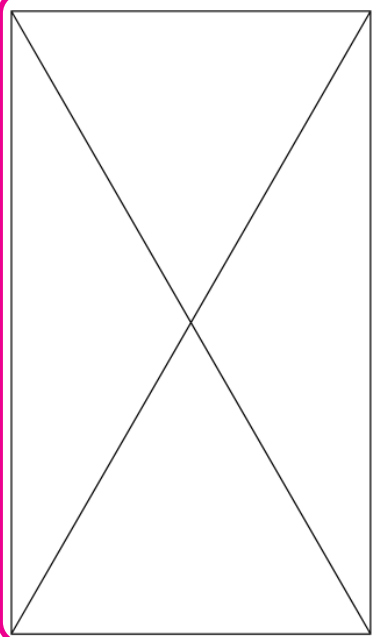
The wireframe shows a desktop browser window with the URL `http://moqups.com`. The page layout includes a header with a logo placeholder (a square with an 'X') and the text "Purr Palace". Below the logo is the text "Choose from our arrangements today!". The main content area is divided into two columns. The left column contains a large rectangular placeholder for a "theme image" (a square with an 'X'). Below this image is the text "Kittens enjoying a furry weekend". The right column contains a form with the following fields: "Your name:" (text input), "Email address:" (text input), "Cat name:" (text input), "Arrangement type:" (radio buttons for "Feline Happiness €19.99", "Gourmet Midweek €25.99", and "Furry Weekend €29.99"), "Date:" (text input with a calendar icon), and a "Book" button.

Moqzilla

← → ↻ `http://moqups.com`

 **logo** *Purr Palace*

Choose from our arrangements today!

 **theme image**


Your name:

Email address:

Cat name:

Arrangement type:

☒ Feline Happiness €19.99 ☐ Gourmet Midweek €25.99 ☐ Furry Weekend €29.99

Date:  

Kittens enjoying a furry weekend

Figure 1: booking form in desktop mode

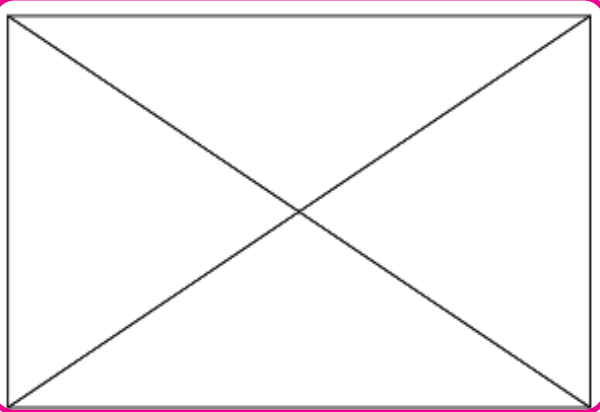
Mozilla

← → ↻ <http://moqups.com>



*Purr Palace*  
logo

Choose from our arrangements today!



Kittens enjoying a furry weekend

Your name:

Email address:

Cat name:


Arrangement type:

☐ Feline Happiness €19.99

☐ Gourmet Midweek €25.99

☐ Furry Weekend €29.99

Date:



Book

theme image

Figure 2: booking form in smartphone mode

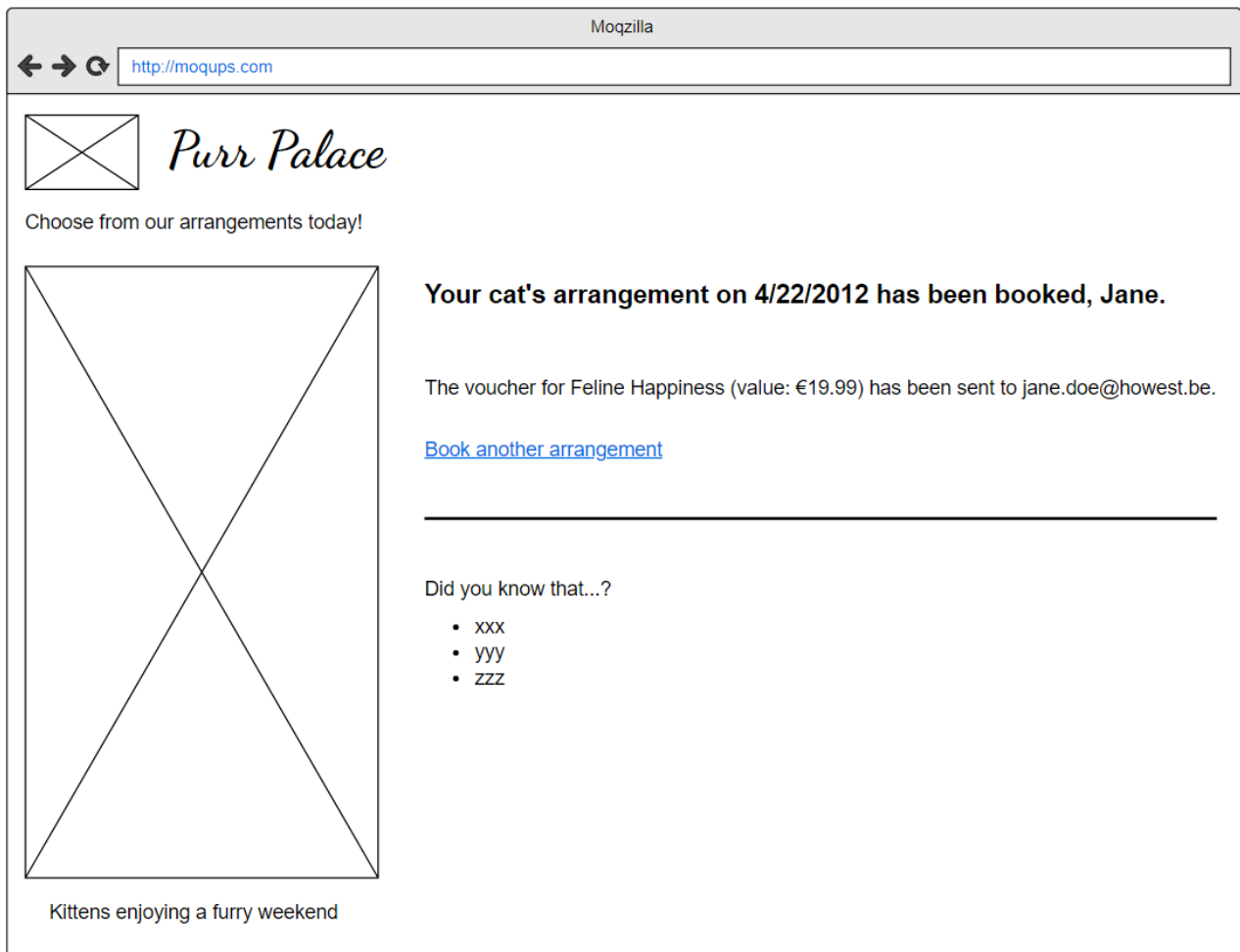


Figure 3: confirmation page in desktop mode



Figure 4: confirmation page in smartphone mode

### 3. GRAPHICAL ELEMENTS

You have freedom to create the various pages/forms/elements, but you should take the following into account:

- Logo (see first wireframe): provided on the exam folder
- Theme image (see first wireframe): provided on the exam folder
- Default font:
  - o Arial for the body text (make sure to include suitable fallback fonts)
  - o Google Font 'Dancing Script' for the logo text (make sure to include suitable fallback fonts)
- Main colors:
  - o Black
  - o Gray
  - o White
  - o Howest pink: rgb(236, 0, 140)
  - o Soft pink for logo text: rgb(255, 172, 178)

#### 4. DATABASE

- You are provided with a SQL dump of a MySQL database. Import this into your MySQL instance using SQLyog, without changing the database or table names.
- The database contains two tables:
  - o **arrangements**: contains all different types of arrangements
  - o **bookings**: will be populated with the individual bookings made for the individual cats
- Connect to the database using user **user** and corresponding password **user**. No other user/password combinations are allowed!

#### 5. FORM STRUCTURE

The reservation form contains the following fields. Implement as much validation as possible, both on client and server.

- Your name
  - o Text field
  - o Required
  - o Max length 50 characters
  - o Corresponds to the column "customername" in the database
- Email address
  - o Email field
  - o Required
  - o Max length 255 characters
- Cat name
  - o Text field
  - o Required
  - o Max length 50 characters
- Date
  - o Date field
  - o Required
  - o Date cannot lie in the past
- Arrangement
  - o Radio buttons
  - o Required
  - o Dynamically filled in with the various arrangement types retrieved from the database, includes both title and price
- Submit button
  - o Submits the reservation

#### 6. RESERVATION FLOW

When the user fills in the form correctly and clicks the submit button:

- the reservation is added to the database table **booking** (see §4)
- a confirmation mail is sent to the specified email address, containing the various details (see §7)
- a thank you message summarizing the booking is displayed in the browser (see wireframes)

Should the form not be filled in completely (or in case of errors), error messages are displayed (in Howest pink) at the top of the form as a bulleted list. Of course, the reservation process will not complete successfully in this situation. Instead the user is to correct any invalid input and try submitting again.

## 7. CONFIRMATION MAIL

No real-life messages are to be sent. Instead, use the Mailtrap service as was done during the course of the year. Make sure you correctly configure Laravel so your Mailtrap account is used.

**Remember: on Howest infrastructure, use port 465 for Mailtrap. The default port will not work!**

The confirmation mail should contain the following information and is to be sent to the mail address submitted in the form:

- Customer name
- Cat name
- Arrangement type
- Arrangement price
- Reservation date

## 8. FELINE FACTS

As you can see on the confirmation pages, at the bottom a list of “Did you know that...?” facts are displayed regarding cats in general.

The list of facts is retrieved dynamically through JavaScript from the following API:

<https://fredericvlummen.be/howest/cats.php>.

Not all facts are taken into account: only facts with **5 or more upvotes** are displayed. Only the **text** needs to be displayed, which additionally needs to be brought back to **lowercase** prior to being displayed.

Example:

Did you know that...?

- cats can hear the ultrasonic noises that rodents and dolphins make to communicate.
- by the time a cat is 9 years old, it will only have been awake for three years of its life.
- the frequency of a domestic cat's purr is the same at which muscles and bones repair themselves.

*(continued for all facts with 5 or more upvotes)*



## 9. LOGIN FUNCTIONALITY

Once your registration mechanism is up and running, implement standard Laravel authentication, allowing users to register and login.

Add a registration and login link to your main view. There is no need to re-style the generated views they will point to.

Make sure only logged in users can actually add a new booking. Update your view(s) accordingly and protect the necessary routes as well.

Since a logged-in user has an e-mail address, make sure it is filled in in the Email address field of the booking form automatically. However, the user should be able to modify it.

**The feline facts should also be displayed if a user is not logged in.**

## 10. ATTENTION POINTS

- Pay attention to clean and readable code.
- Place your files in the correct folders, honouring the Laravel conventions.
- Responsiveness is to be obtained using vanilla CSS. **Do not** use Bootstrap or any other libraries/frameworks.
- Validate your HTML on a regular basis.
- For client-side scripting, do not use jQuery. The requirements can be obtained using vanilla JavaScript.
- Make sure to use any best practices studied during the classes. This applies not only to the Laravel and JavaScript code, but also to HTML and CSS.

## 10. FINISHED?

Re-read the instructions at the beginning of this document thoroughly. Make sure all required files (including the .env file) are uploaded into your repo.

Also, do not forget to upload the HOSA zip file onto Leho through the appropriate assignment.