Multiple choice

Question	Question title	Marks	Question type
ℹ	Frontpage 2020		Document
1	HTML guess 2020	2	Multiple Choice
2	HTML guess 2 2020	2	Multiple Choice
3	HTML guess 3 2020	2	Multiple Choice
4	CSS guess 2020	2	Multiple Choice
5	CSS guess 3 2020	2	Multiple Choice
6	POST guess 2020	2	Multiple Choice
7	HTTP request guess 2020	2	Multiple Choice
8	What Vue Router	2	Text Entry

Programming exercises

Question	Question title	Marks	Question type
ℹ	Info coding 2020t		Document
9	HTML Form Coding 2020	10	Text area
10	CSS Coding 2020	10	Text area
11	JS Hide 2020	6	Text area
12	JS Numbers 2020	15	Text area
13	Jinja pages 2020	18	Text area
14	Vue List 2020	15	Text area
15	Vue categories 2020	5	Text area
16	AJAX 2020	5	Text area
17	Upload	0	Upload Assignment

i Frontpage 2020

Exam: DAT310 webprogramming

Available time: 4 hours

Supporting material: All technical support material is permitted. You are not allowed to get help from other people when working on your exam assignment. We want to remind you that you, when registering for the semester, signed that you have read and understood the rules for cheating and plagiarism in the Exam Rules and Regulations at the University of Stavanger. Plagiarism control will be carried out.

Important contact information during the exam: If something should happen during the exam and you need help, you can call one of the phone numbers below, both if you need technical support, administrative support or have questions for the course responsible.

- Course responsible: Leander Jehl 51 83 20 62
- Administrative support: 51 83 17 15 / 51 83 31 33 / 92 81 65 97 / 91 78 67 16

NB! If something goes wrong in Inspera, and you are not able to hand in your assignment, you can send your file to eksamenTN@uis.no. **Please write candidate No and course code in e-mail.**

About the exam

This exam consists of two parts.

Part 1 contains 8 multiple choice questions.

Answers should be given using the available forms.
The scoring of multiple choice questions is as follows

- The correct solution is 2 points
- Wrong answer is -1 point

Part 2 contains 9 programming exercises.

Starter files for all exercises are provided as a zip archive.

Download the starter files and complete them.

Carefully read all requirements stated in the exercises.

To submit your solutions to part 2, submit a single zip file, containing the starter files and your added code.

This zip file can be uploaded on the last page of the exam.

Remember to upload your code before the exam deadline!
The exam will automatically close for uploading when the time is up. Remember that the time given includes the time it will take you to make a zip archive and upload your code.

About Inspera

If you have been given extra time, it has been added to your user. You can see the count down on the top of the page. Here you will also find your candidate number.

Good luck!

1 HTML guess 2020

Why are hidden form fields useful? *(2 points)*

Select an alternative:

○ because they stay out of the way

○ because they can hold onto data for use later on ✓

○ because their values are unique

○ because what you don't know can't hurt you

Maximum marks: 2

2 HTML guess 2 2020

Consider an image that is 1024 pixels in width and 768 pixels in height. What is the best practice for displaying a 50 pixel by 50 pixel thumbnail version of the image that, when clicked, will cause the full size image to be displayed?

Select an alternative:

○ use JavaScript to resize the image

○ create a new thumbnail image that links to the larger image ✓

○ set the width and height attributes of the tag to 50

○ set the width and height properties of the CSS to 50px

Maximum marks: 2

3 HTML guess 3 2020

Is it possible to link to a specific location within the current page? *(2 points)*
Select an alternative:

○ Yes ✓

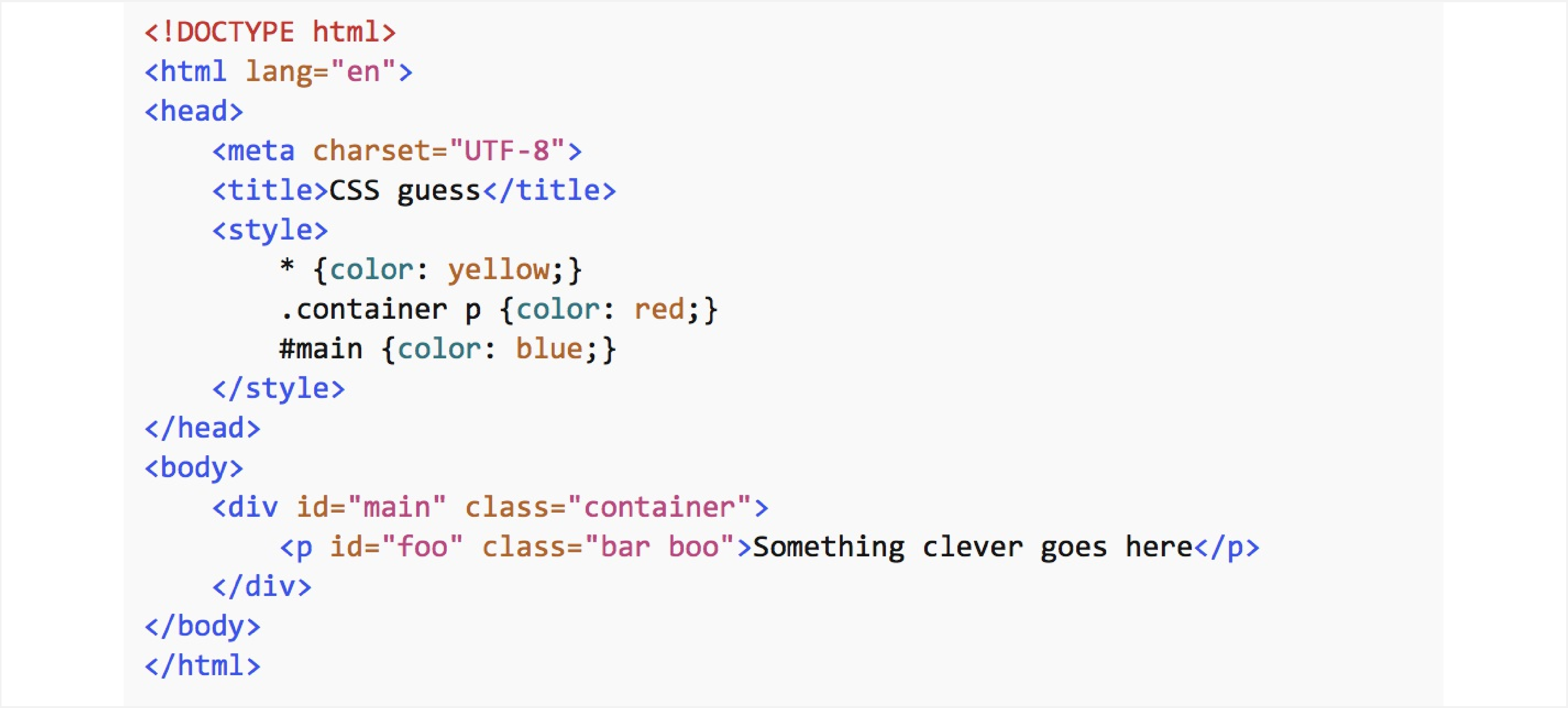
○ Only if the page is generated from Python

○ No

○ Only in framesets

Maximum marks: 2

4 CSS guess 2020



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS guess</title>
    <style>
        * {color: yellow;}
        .container p {color: red;}
        #main {color: blue;}
    </style>
</head>
<body>
    <div id="main" class="container">
        <p id="foo" class="bar boo">Something clever goes here</p>
    </div>
</body>
</html>
```

What color will the text be? *(2 points)*

Select an alternative:

○ black

○ yellow

○ blue

○ red ✓

Maximum marks: 2

5 CSS guess 3 2020

Assume this element
<div id="controls" style="position:fixed; height:100px; width:100px; bottom:0; right:20px;">...</div>
is present on a page that is 2000px tall being viewed in a browser that has a viewport of 960px by 600px. How far does the user have to scroll down the page before they see the element?

Select an alternative:

○ 2000 pixels

○ 0 pixels ✓

○ 500 pixels

○ 600 pixels

Maximum marks: 2

6 POST guess 2020

Which of the following is true for HTTP **POST** requests:
Select one alternative:

○ Data is displayed in the URL

○ Should be used when dealing with sensitive data ✓

○ Have length restrictions

○ Can be bookmarked

Maximum marks: 2

7 HTTP request guess 2020

What is sent along as part of the HTTP request? *(2 points)*
Select an alternative:

○ Cookie ✓

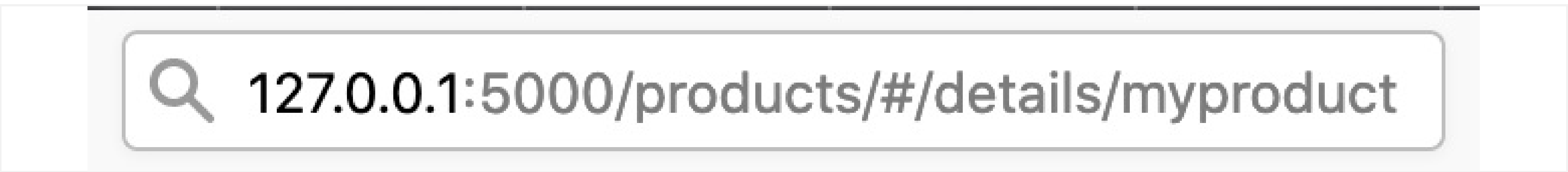
○ Session

○ Both of them

○ Neither of them

Maximum marks: 2

8 What Vue Router



127.0.0.1:5000/products/#/details/myproduct

The URL above is from a Vue application.

Write the part of the URL that is processed by the vue router: (/details/myproduct, details/myproduct)

Maximum marks: 2

Info coding 2020t

Part 2: In the next exercises, you have to complete the code given as a zip arkive below.

Some of the exercises have several minor tasks. Read the complete exercise description before starting on the tasks. Make sure you do not forget one of the tasks.

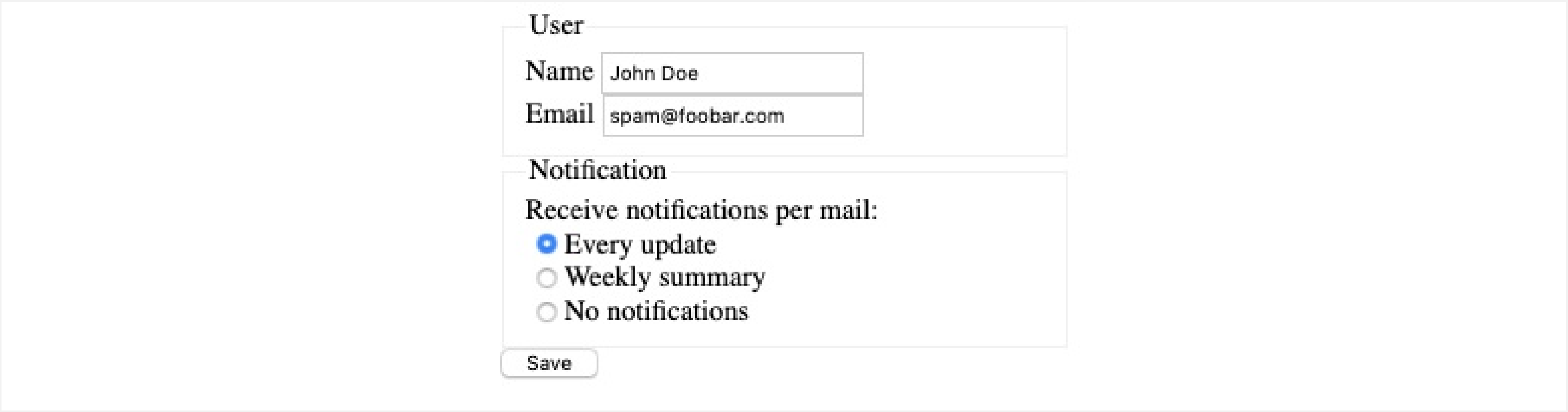
You can download all start files as a zip archive below.
The zip file includes the following folders:

- html (exercise 9)
- css (exercise 10)
- js_hide (exercise 11)
- js_memory (exercise 12)
- flask_pages (exercise 13)
- vue_list (exercise 14)
- vue_category (exercise 15)
- ajax (exercise 16)

Each folder contains starting code for one exam question.
Please solve exercises in place and upload your code in question 17.

You are not allowed to use/include additional frameworks, e.g. jQuery, ...
Unless noted differently, you may change the existing code.
Download starting files here: start_files.zip

9 HTML Form Coding 2020



User

Name John Doe

Email spam@foobar.com

Notification

Receive notifications per mail:

● Every update

○ Weekly summary

○ No notifications

In this exercise you need to create the HTML form shown above.
Please create the form in the file **start_files/html/index.html** in the start_files folder.

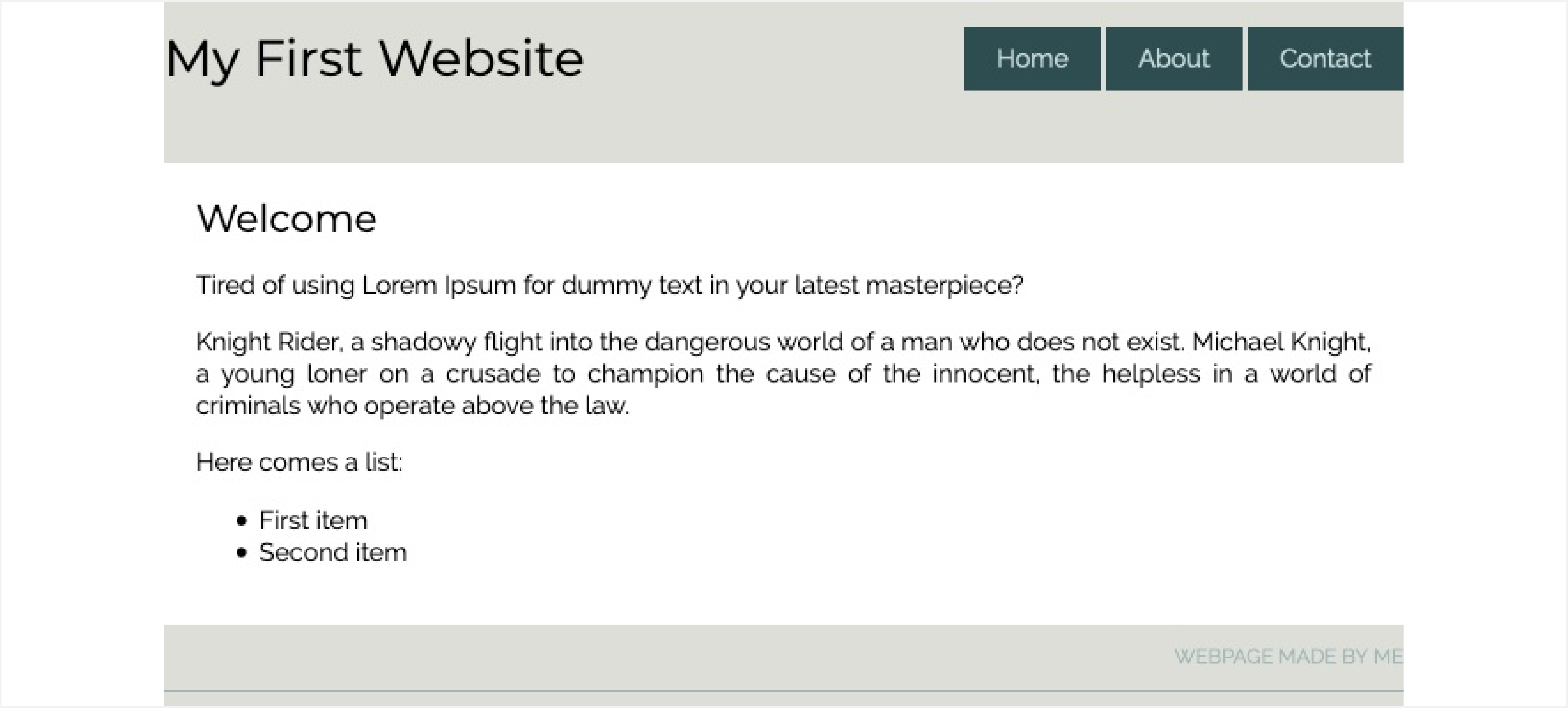
You need to meet the following criteria:

1. Form inputs are separated into two groups **User** and **Notification**.
    - Use appropriate tags.
2. Use <label> tags where appropriate.
3. The form has the following inputs for name, email and three radio buttons, as shown.
4. The form is prefilled as shown, e.g. name: "John Doe", email: "spam@foobar.com", every update selected.
5. When the add button is clicked with the prefilled form (no changes), a get request is sent with the following address:
    - **/user?name=John+Doe&email=spam%40foobar.com&notification=all**

Fill in your answer here

Maximum marks: 10

10 CSS Coding 2020



My First Website

Welcome

Tired of using Lorem Ipsum for dummy text in your latest masterpiece?

Knight Rider, a shadowy flight into the dangerous world of a man who does not exist. Michael Knight, a young loner on a crusade to champion the cause of the innocent, the helpless in a world of criminals who operate above the law.

Here comes a list:

- First item
- Second item

WEBPAGE MADE BY ME

Write an external CSS file that displays the given HTML page as shown. *(10 points)*
You are not allowed to make changes to the HTML source.

You can find the html side at **start_files/css/index.html**.
Write your css queries into **start_files/css/style.css**.

Specific requirements:

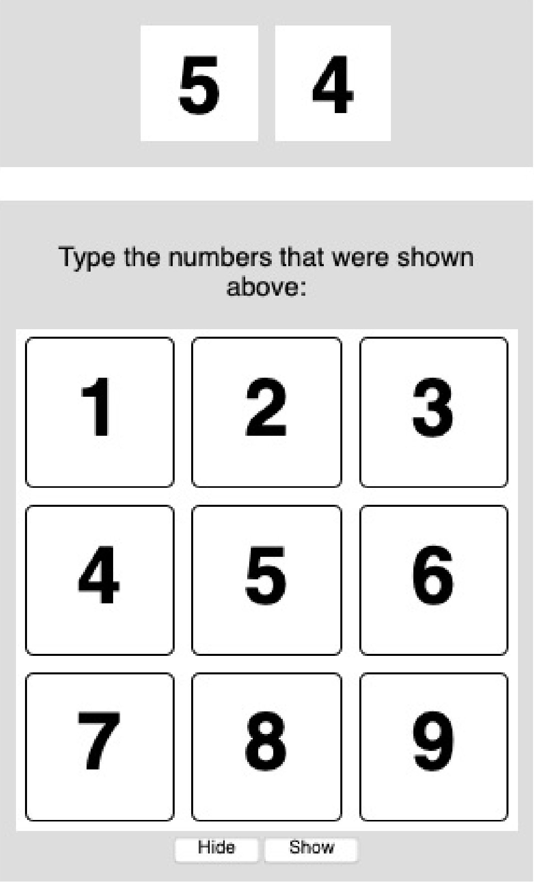
1. The background of the entire page is #DDDED7.
2. Headers are written with the font Montserrat, all other text uses the font Railway,
   Both cases use sans-serif as backup font.
   - Fonts Montserrat and Raleway need to be imported:
     - 'https://fonts.googleapis.com/css2?family=Montserrat'
     - 'https://fonts.googleapis.com/css2?family=Raleway'
3. Both headings have normal font weight, 0 margin and 0 padding (on all sides)
4. Navigation menu is aligned on one line with the "logo" div. List items are aligned horizontally next to each other.
   - Hints:
     - change the display property of the navigation list items to inline
     - set the display property of the header to flex
     - ensure #logo and nav are positioned at the left and right end of the header
   - (you can also use float instead of a flexbox)
5. Navigation menu items have:
   - 10px padding on the top and bottom and 20px padding left and right,
   - background color #2D4D50 and text color #C6D8DA,
   - no underlining
6. Main content:
   - Background is white
   - Padding is 20px
   - Top margin is 30px, other margins are 0px.
7. Footer
   - Set 1px wide solid bottom border in #A1B7B9
   - Set bottom margin to 10px
   - Text is right aligned, font size set to 0.8em, colored #A1B7B9, and set to uppercase.

You can add comments here:

Fill in your answer here

Maximum marks: 10

11 JS Hide 2020



Type the numbers that were shown above:

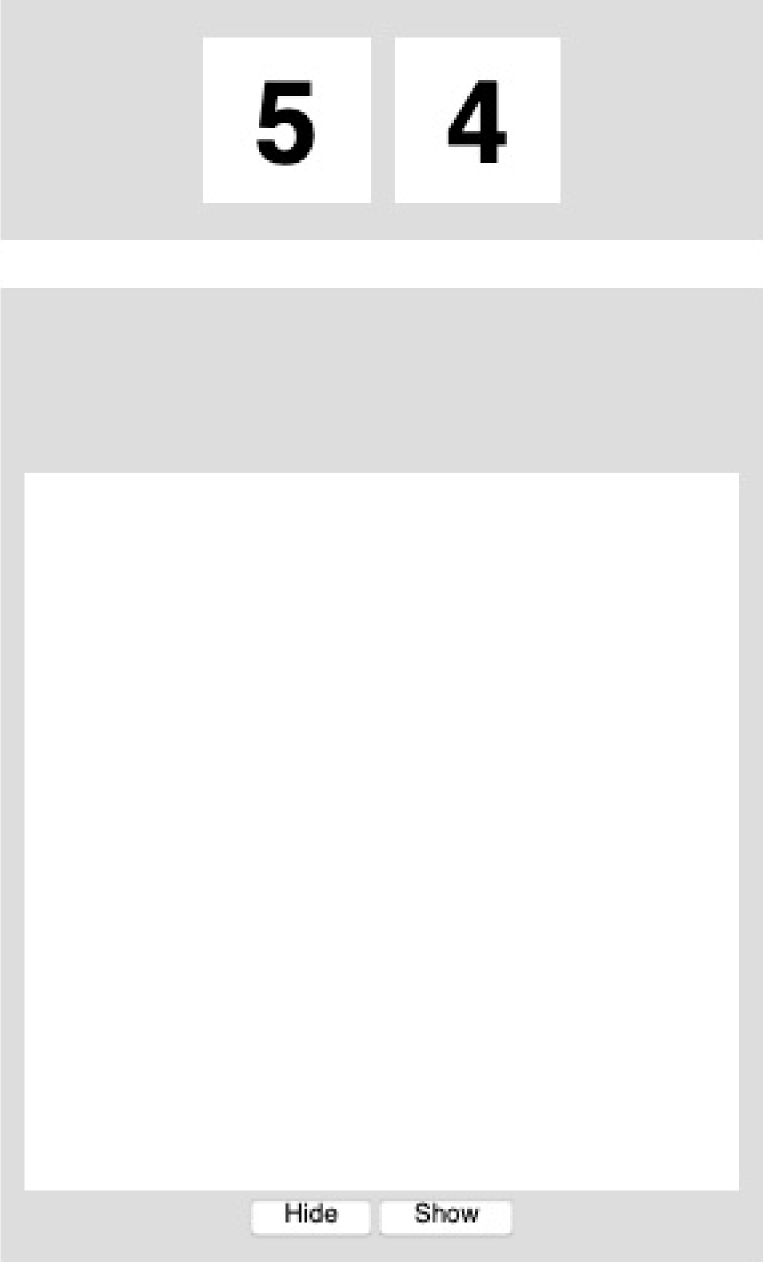
In this exercise you have to write some functions in Javascript *(6 points)*

You can find starting files in **start_files/js_hide/**.
You have to implement some functions in **start_files/js_hide/script.js**.
Do not change the index.html file.

You have to implement the *hide( )* and *show( )* functions.

- The *hide( )* function hides the buttons with class "number" and the paragraph with id "padinstructions".
  - Make sure you only hide these elements.
  - Elements should be hidden but still take up space.
- The *show( )* function undoes the effects of the *hide* function.

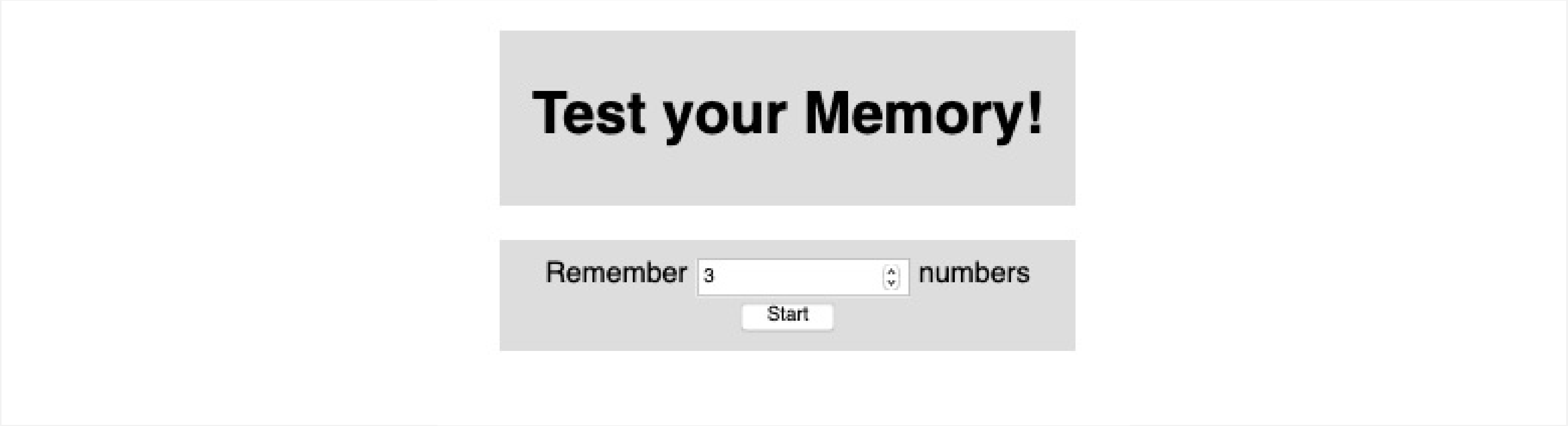
Below you can see the page, after hiding elements.



Add comments here.

Maximum marks: 6

12 JS Numbers 2020



In this exercise you have to complete a Javascript application *(15 points)*

You can find starting files in **start_files/js_memory/**.
You have to implement some functions in **start_files/js_memory/script.js**.
You are not allowed to use third party frameworks, e.g. Vue, jQuery, ...

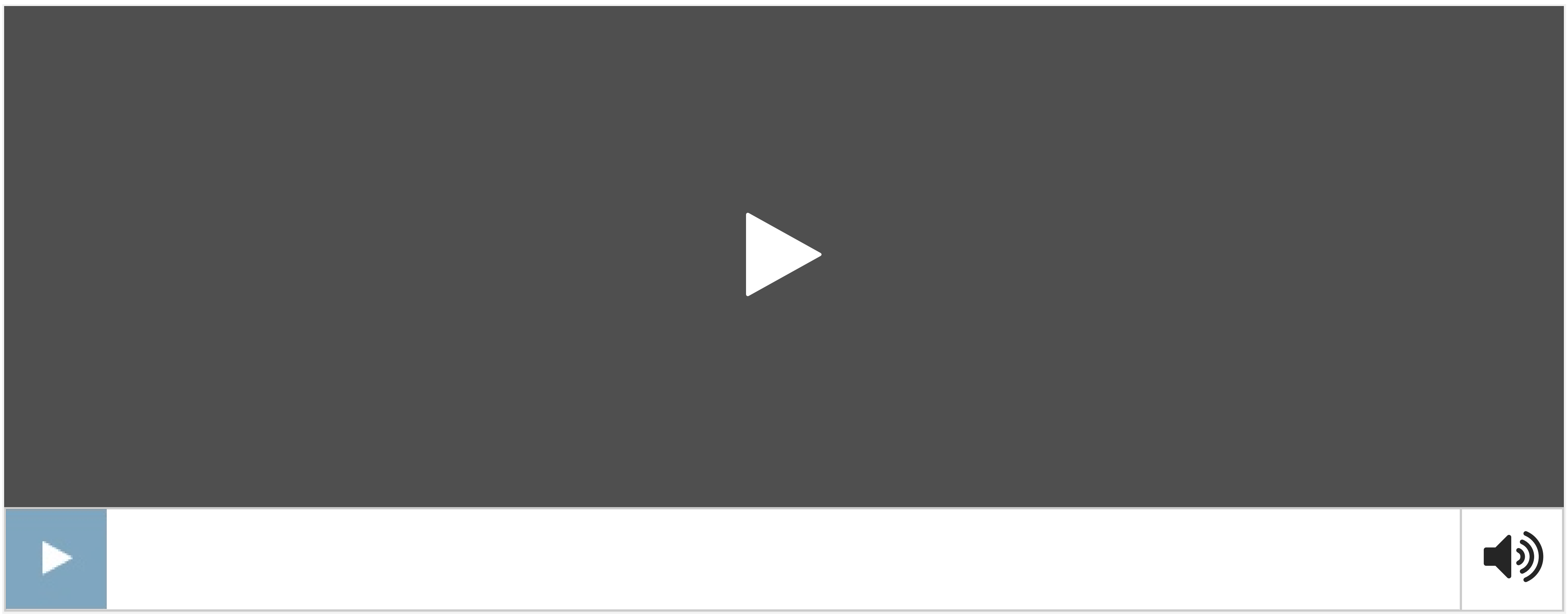
Desired functionality:
The application is a memory training game.

- The user can select how many numbers to remember and start the game.
- The initial form disappears and a number pad appears.
- After 1 second the app shows randomly selected numbers, each between 1 and 9.
    - How many numbers there are is based on the users input.
- Numbers show for half a second (i.e. 500 milliseconds).
    - For each number, create a div with class "number" and the number as content e.g. <div class="number">5</div>
    - Add these elements inside the div with id numberbox.
- The user then has to type the numbers on the number pad (in order).
- Correctly typed numbers appear.
- After adding all numbers correctly, an alert message is shown congratulating the user.
- Then the user can start a new game.
- If the user types a false number, the correct number appears in red, the game is over and the user is alerted.

The video below also shows how the app should work.

Hints:

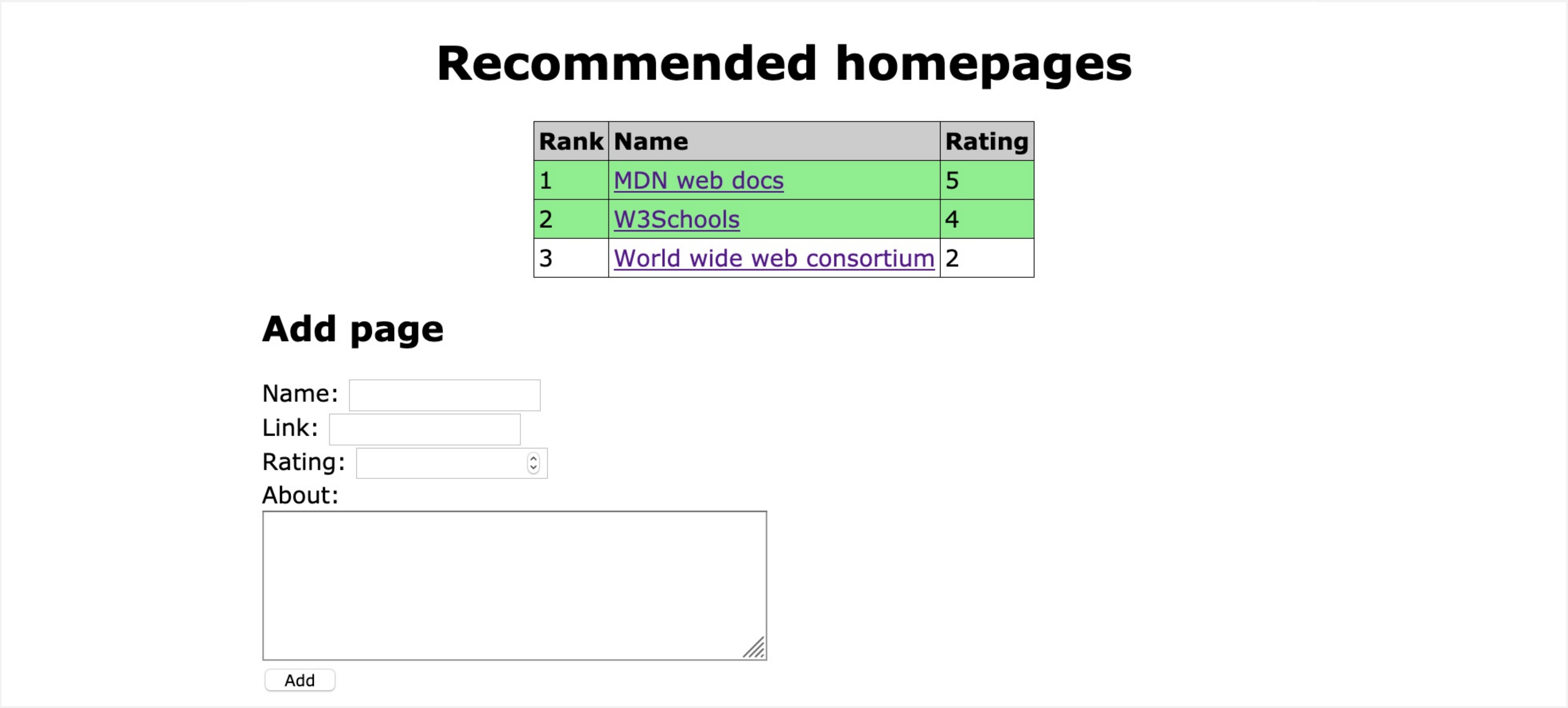
- The start files contain functions for hiding and showing the different elements.
- The *random( )* function in the start code gives a random number between 1 and 9.
- **Use** *setTimeout* **to show or hide elements after a specific time.**
- You can use global variables to remember the created numbers and check if they are entered correctly.
- More hints are added as comments in the start files.



You can add comments here.

Maximum marks: 15

13 Jinja pages 2020



Recommended homepages

Rank	Name	Rating
1	MDN web docs	5
2	W3Schools	4
3	World wide web consortium	2

Add page

Name:
Link:
Rating:
About:

In this exercise you have to complete a Flask application *(18 points)*

You can find starting files in **start_files/flask_pages/**.

The application lists info and ranking for several useful web pages.
You have to:
A: complete the *index.html* template in **start_files/flask_pages/templates.**
B: implement function in *app.py* to add and edit pages
C: write a new template *page.html* to display page details

Specific requirements
A: In *index.html* replace the static table with one created dynamically. Including:

- Use a for loop to display one row for every entry in the PAGES array.
- Rows are **ordered by rating**, in descending order.
- Rows with rating 4 or higher have class **good**.
- The first column (Rank) shows increasing numbers.

B: The *index.html* page contains a form. Write the *add( )* function in app.py to process the form.

- Retrieve data from the form.
- If the name is non-empty, add the data to the PAGES array.
- Add a new index to the new page.
- Redirect to the index page.

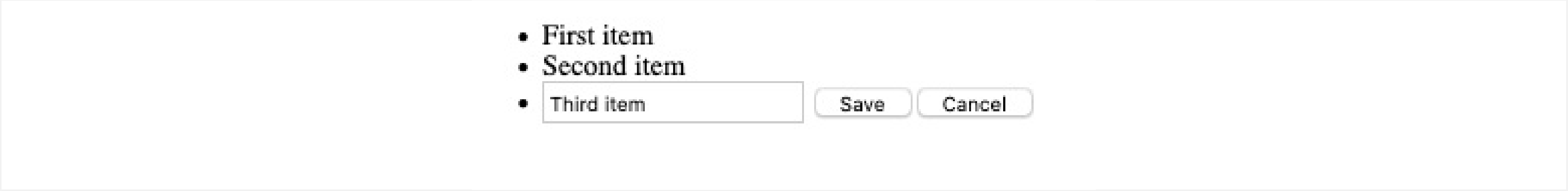
C: Create a new route and template that shows details for a page.

- Create a new route in *app.py,* that takes the index of the page as parameter.
- Make the names of pages in the table links to that route.
    - E.g. */details/0* could be the route for details for the entry with index 0 in PAGES.
- Create a template that displays details of a page
- An example for a rendered details page is in **start_files/flask_pages/static/details.html.**
- It
    - Extends base template
    - Page name displayed as header
    - Includes link to page
    - Displays rating and description

Add comments here

Maximum marks: 18

14 Vue List 2020



- First item
- Second item
- Third item Save Cancel

In this exercise you have to complete a Vue application *(15 points)*

You can find starting files in **start_files/vue_list/**.
You have to write a new component in **start_files/vue_list/listentry.js**.
You will also have to make changes to **start_files/vue_list/index.html**.

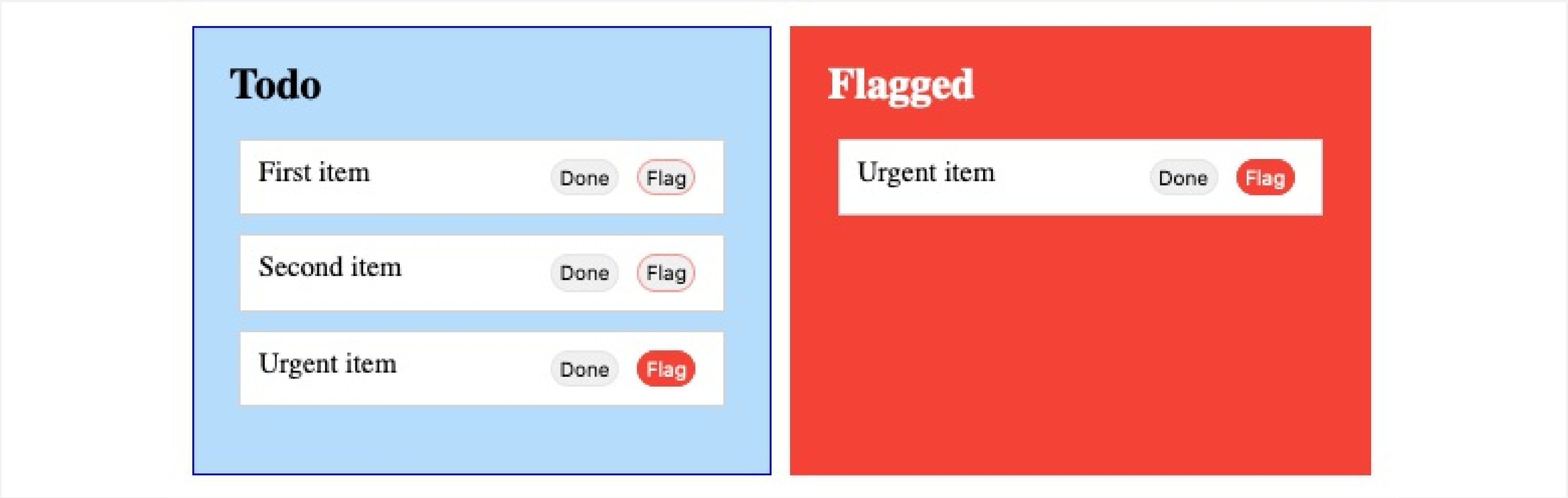
Implement the following:

- Implement the **list-entry** component in *listentry.js.*
- Instead of *item* the component should show one of the elements from *datalist* (see index.html)
- The component only shows the first <span> element (with class "item"). On doubleclick, the second <span> element is shown instead (class "form").
- When first shown, the <input> element contains the *item* value.
- When cancel is clicked, the form <span> disappears and the "item" <span> shows the initial value.
- When the save button is clicked the form disappears and the component emits an event with the new value.
  - The main app (in *index.html*) should process that event and change the entry in the list.
  - After this change, the new value should be displayed.

Add comments here

Maximum marks: 15

15 Vue categories 2020



In this exercise you have to complete a Vue application *(5 points)*

You can find starting files in **start_files/vue_category/**.
Start the application by opening **start_files/vue_category/index.html** in the browser.

The application shows a list of items. Items are located in **data.js.**

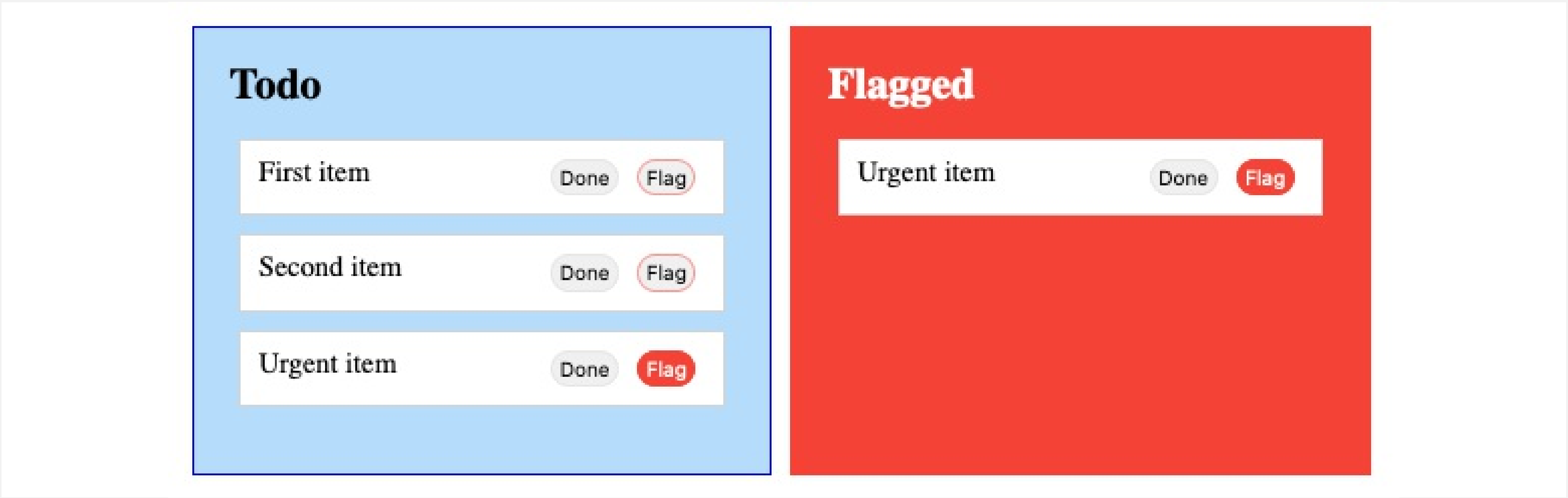
You need to implement the following:

- For elements that are flagged (*flagged*==**true**), the button with label "Flag" should have the class "flagged".
- Complete the *computed* functions.
  - *todo* should return all elements from the list, that are not done (*done*==**false**).
  - *flagged* should return all elements from the list, that are not done (*done*==**false**). and flagged (*flagged*==**true**).

Add comments here.

Maximum marks: 5

16 AJAX 2020



In this exercise you have to write an AJAX call *(5 points)*

You can find starting files in **start_files/ajax/**.
The application simulates roling a dice.
You need to implement function *role( )* in *script.js*.

You need to implement the following:

- Remove old content from result div.
- Retrieve a random number between 1 and 6 from route /random.
- While retrieving, display a loading symbol by adding class "loading" to the result div.
- Add the resulting number as content to the div.
- Ensure at most one number is requested at a time.

Add comments here.

Maximum marks: 5

17 Upload

Rename the start-files folder, including your changes to files-{candidate-number}, including your candidate-number.

Create a zip archive from the whole folder and upload it here:

The file must be a zip archive, with file extension .zip.
Create a zip archive, do not use gzip, 7zip.

Maximum marks: 0