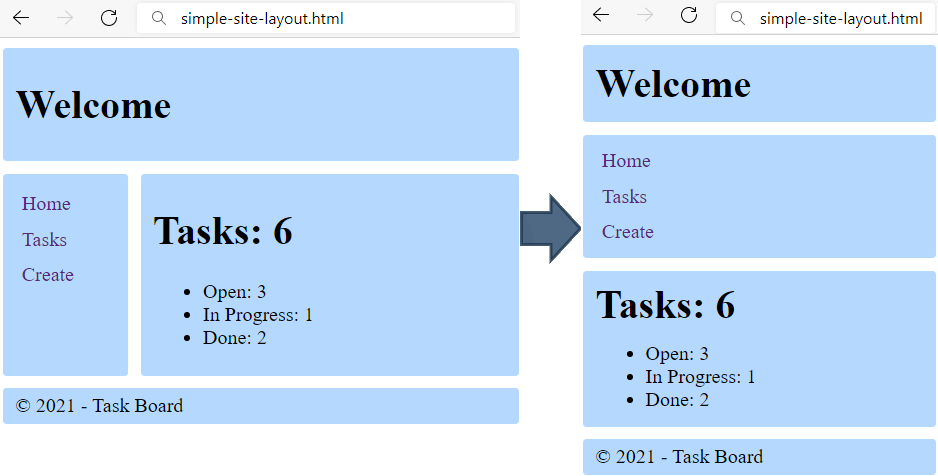
# Exercise: Media Queries

Problems for exercises and homework for the ["HTML and CSS" course @ SoftUni](https://softuni.bg/courses/html-and-css). Submit your solutions in the SoftUni Judge system at <https://judge.softuni.org/Contests/3336/Media-Queries>.

## Responsive Site Layout

Use the solution from the previous task and create responsive layout by using media query.

The behavior of the Web page should be like the following:



Define a maximum width of 500px, if the screen is less than that, set in the body new grid template areas: "header header", "aside aside", "main main", "footer footer", and display the h1 tag in the header as block, and with margin bottom 5px

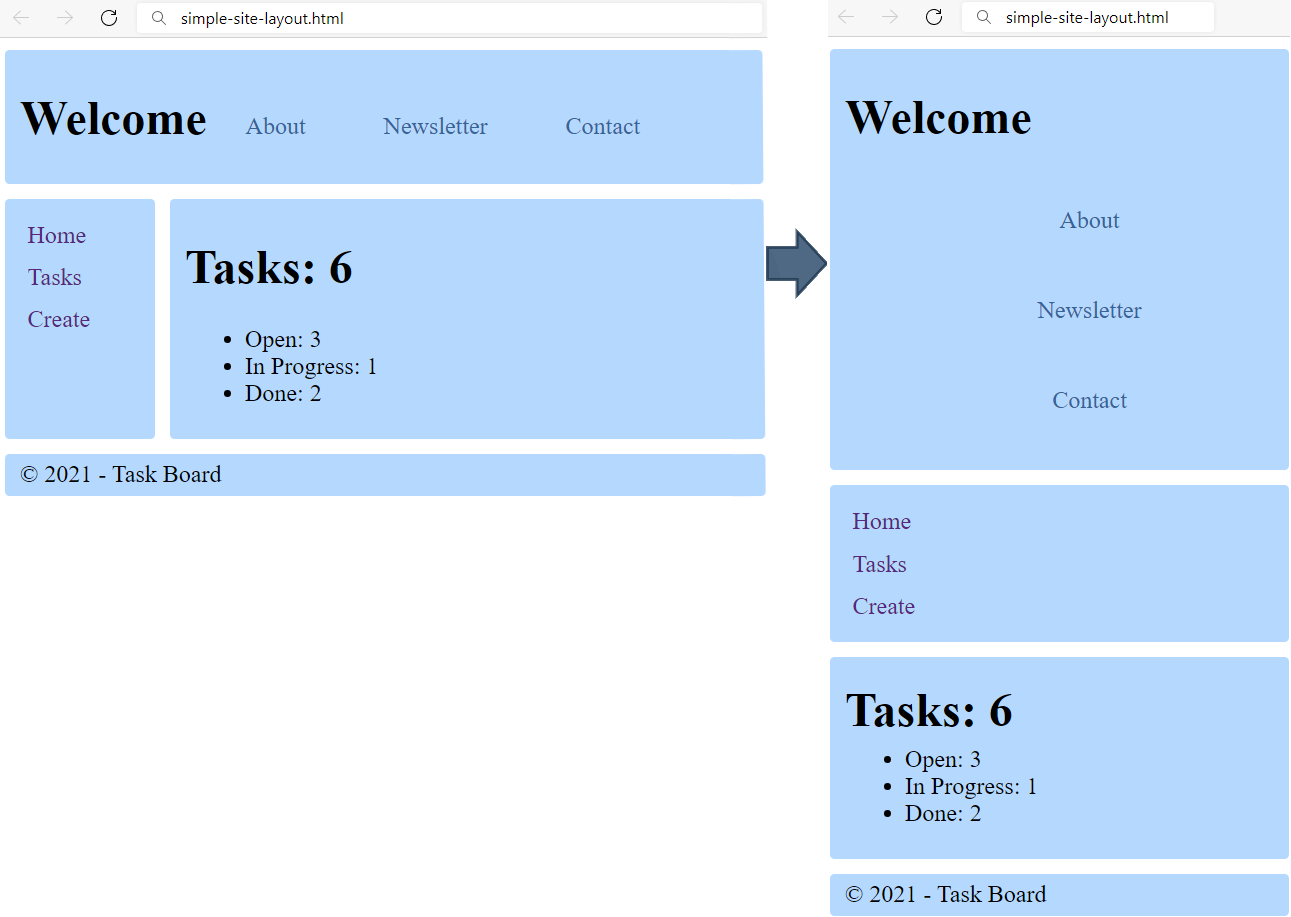
### Hints



## Responsive Top Menu

Add to the previous solution navigation and make it responsive by using media query.

The behavior of the Web page should be like the following:



### Requirements

Add in the header:

* <nav>tag with:
  + display: **inline-block;**
  + vertical-align: **middle;**
  + margin-top: **8px;**
    - <ul> tag for unordered list
      * <li> tag for list item
        + list-style: none;
        + display: inline;
        + <a> tag for hyperlink

text-decoration: none;

color: rgb(53, 100, 153);;

padding: 5px 15px;

Make the navigation responsive by using media query. Define a maximum width of 500px, if the screen is less than that:

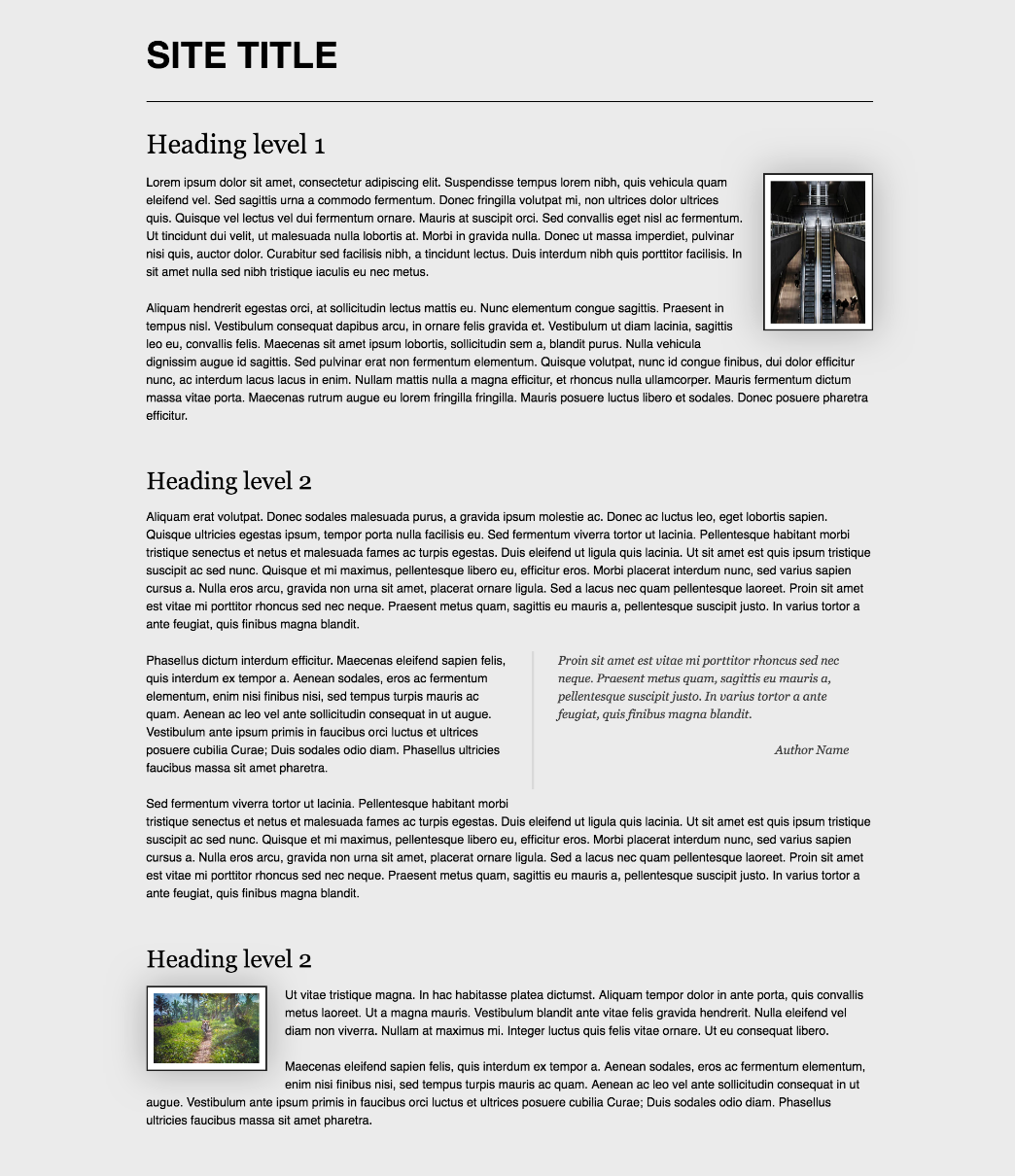
* the <a> tag in the <nav> need to be displayed as block and with padding-left: 5px
* the <ul> tag in the <nav> need to be left with **0** margin and padding

Example for bad behavior and wrong solution:

## Media Queries in Typography

Upgrade your typograpgy.css file from previous homework’s with Media Queries. Create the following page for different screen sizes:

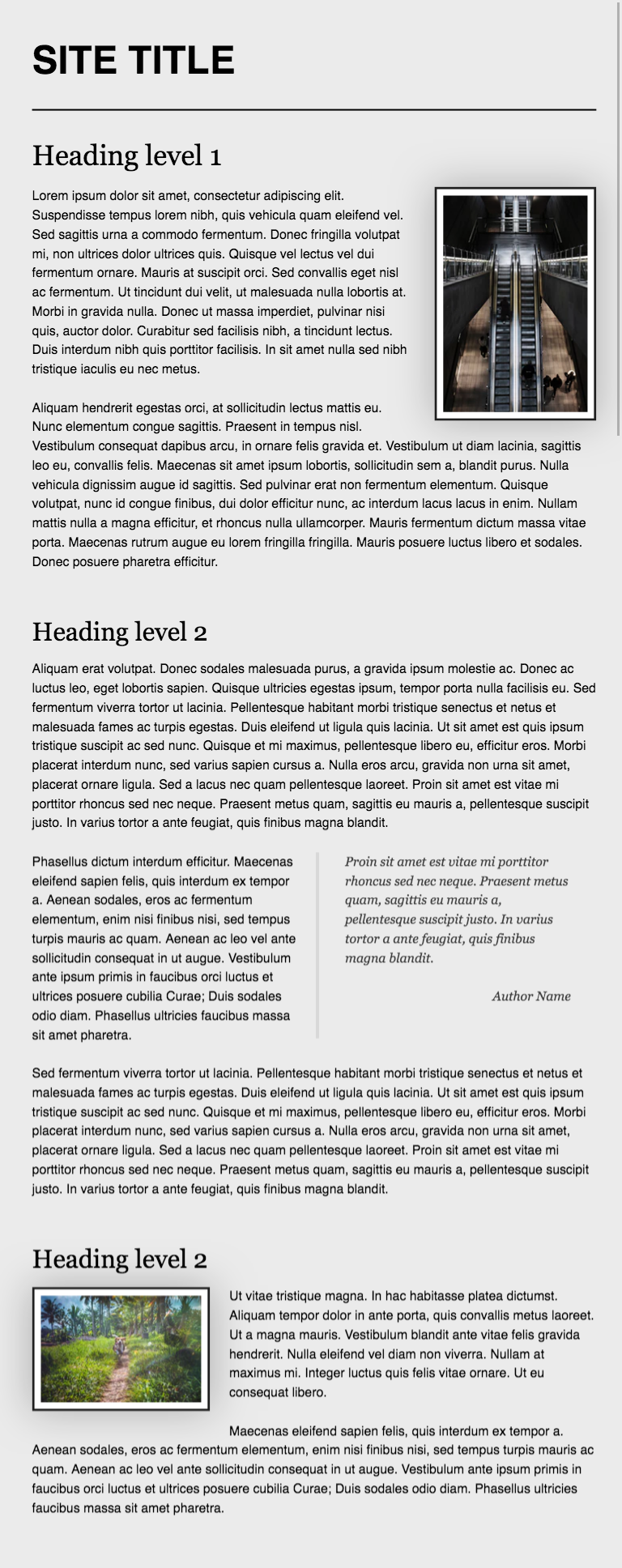
**Desktop:**



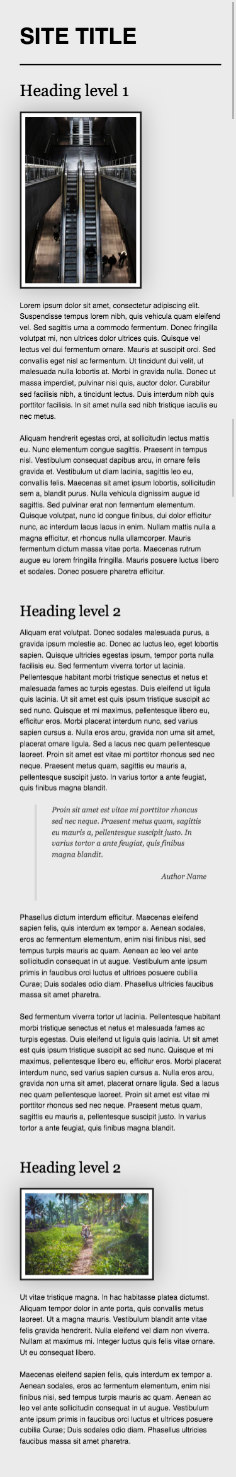
* Laptop



* Tablet



* Phone

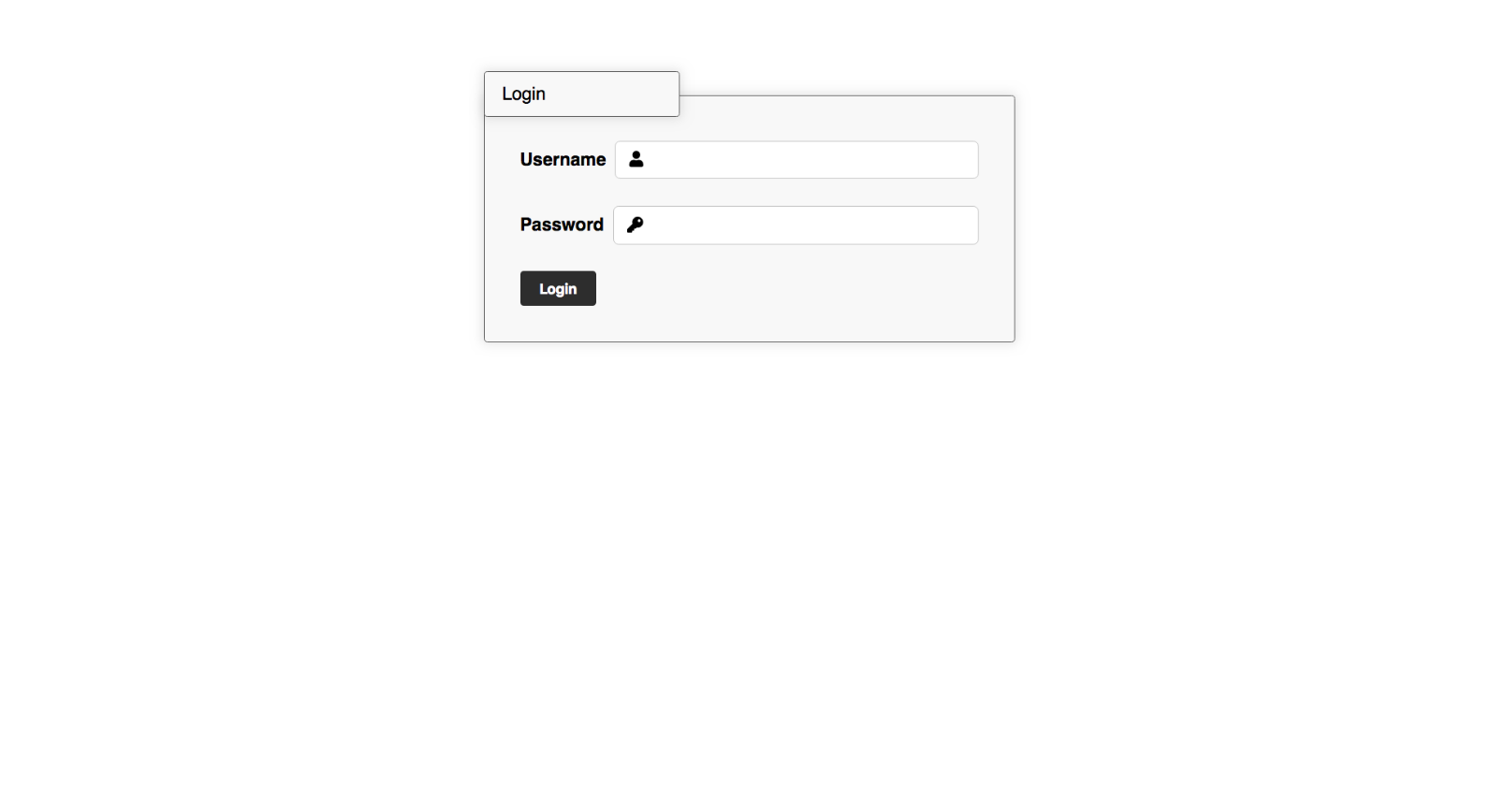


### Requirements

* Change the document **title** to "**Media Queries in Typography"**
* Make the font size for **phones** 12px
* Make the font size for screens **larger** than 500 14px
* Make the font size for screens **larger** than 600 16px
* Make the font size for screens **larger** than 800 18px
* Make the font size for screens **larger** than 1300 21px
* Make the font size for screens **larger** than 1800 23px

## Responsive Forms

Create a web page like the following:



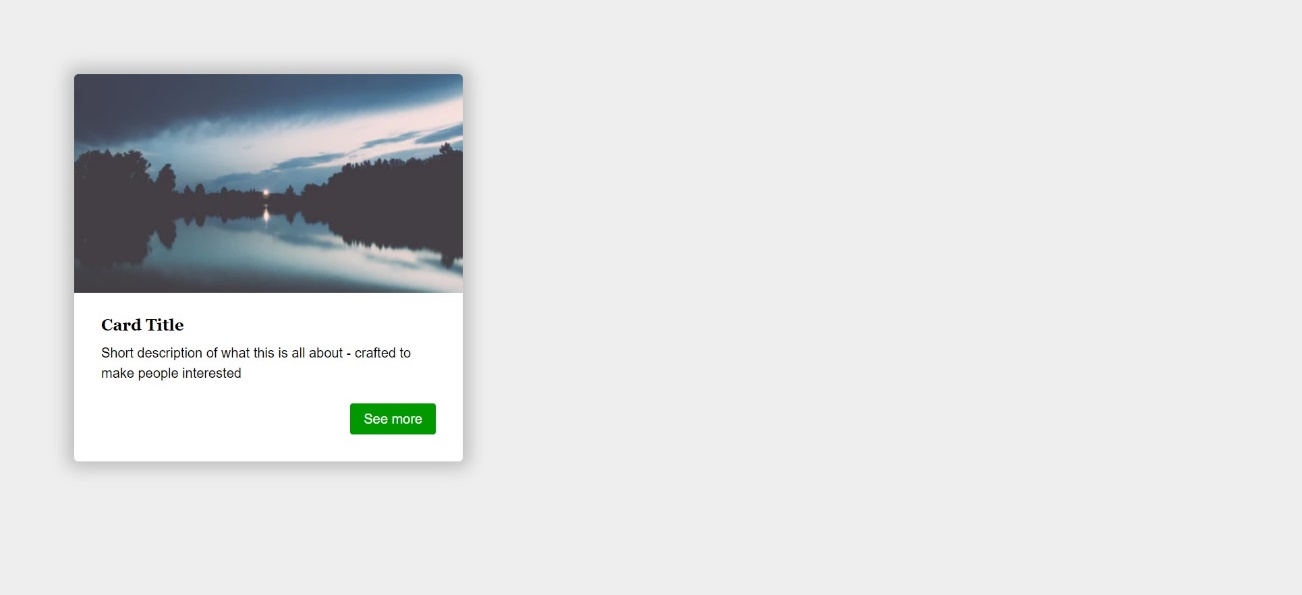
* Look at the provided screenshots in **resources** **folder** for different screen size’s view

### Requirements

* Change the document title to "**Responsive Forms**"
* Create a simple login form using **semantic** html
  + Add icons to the input fields using **FontAwesome**
  + Add focus state that changes the **input** and the **icon**
  + User **form** and two **p** tags

## Responsive Media Boxes

Create a web page like the following:

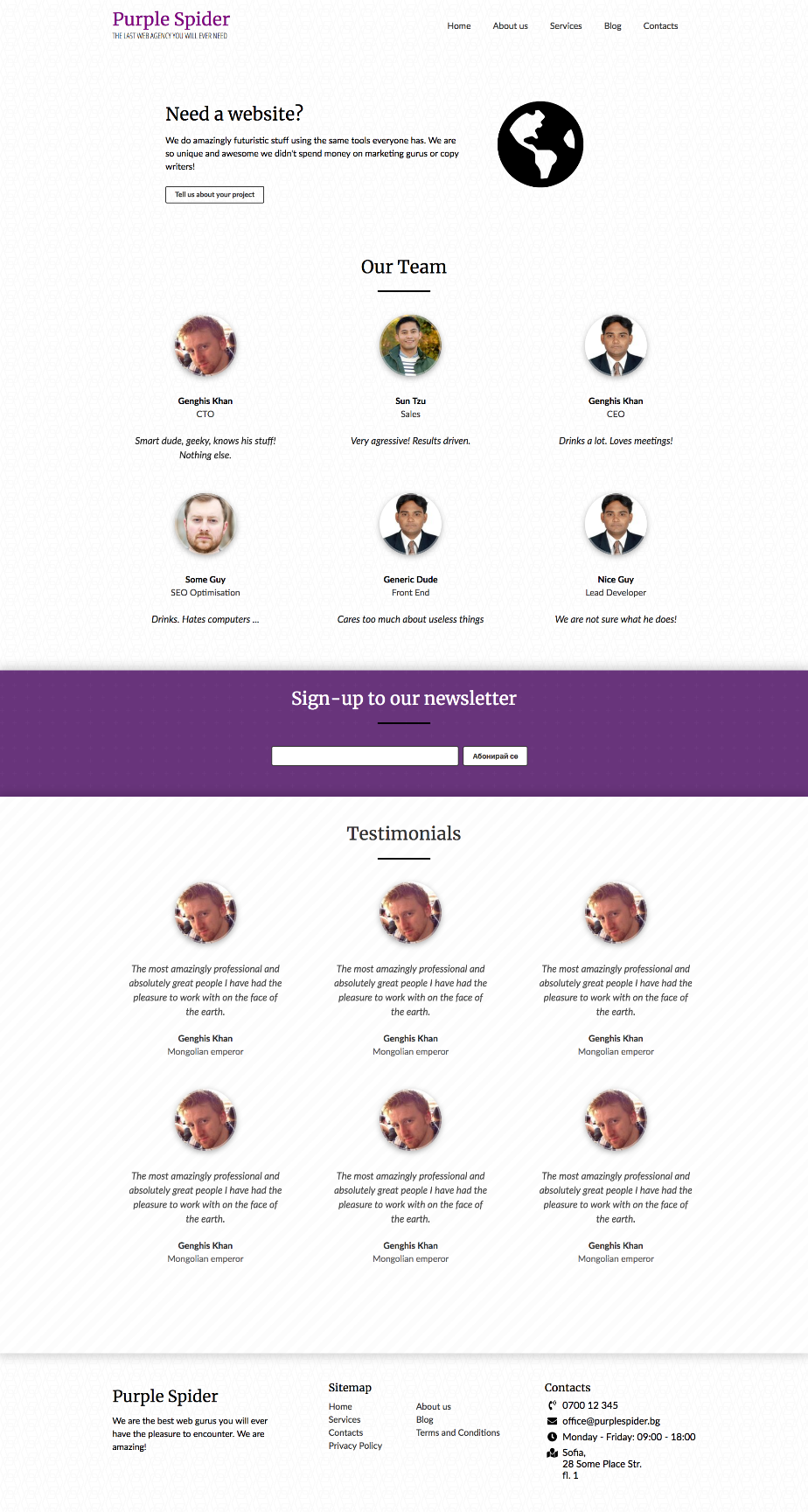


### Requirements

* Change the document title to "**Responsive Media Boxes**"
* Create a section with two div elements
* You can access the image [here](https://images.unsplash.com/photo-1538230942844-e89414a0692d?ixlib=rb-0.3.5&ixid=eyJhcHBfaWQiOjEyMDd9&s=34dcc54db235ce71b59c1d5ada02bdc4&auto=format&fit=crop&w=400&q=80)
* Use rgb(238, 238, 238) for body background
* Use rgb(255, 255, 255) for card background
* Use rgb(0, 153, 0) for card button background
* Use font Georgia, serif with size 1em/1.2 for the headings
* Use font Helvetica, sans-serif for the document

## Landing Page

Create a web page like the following:



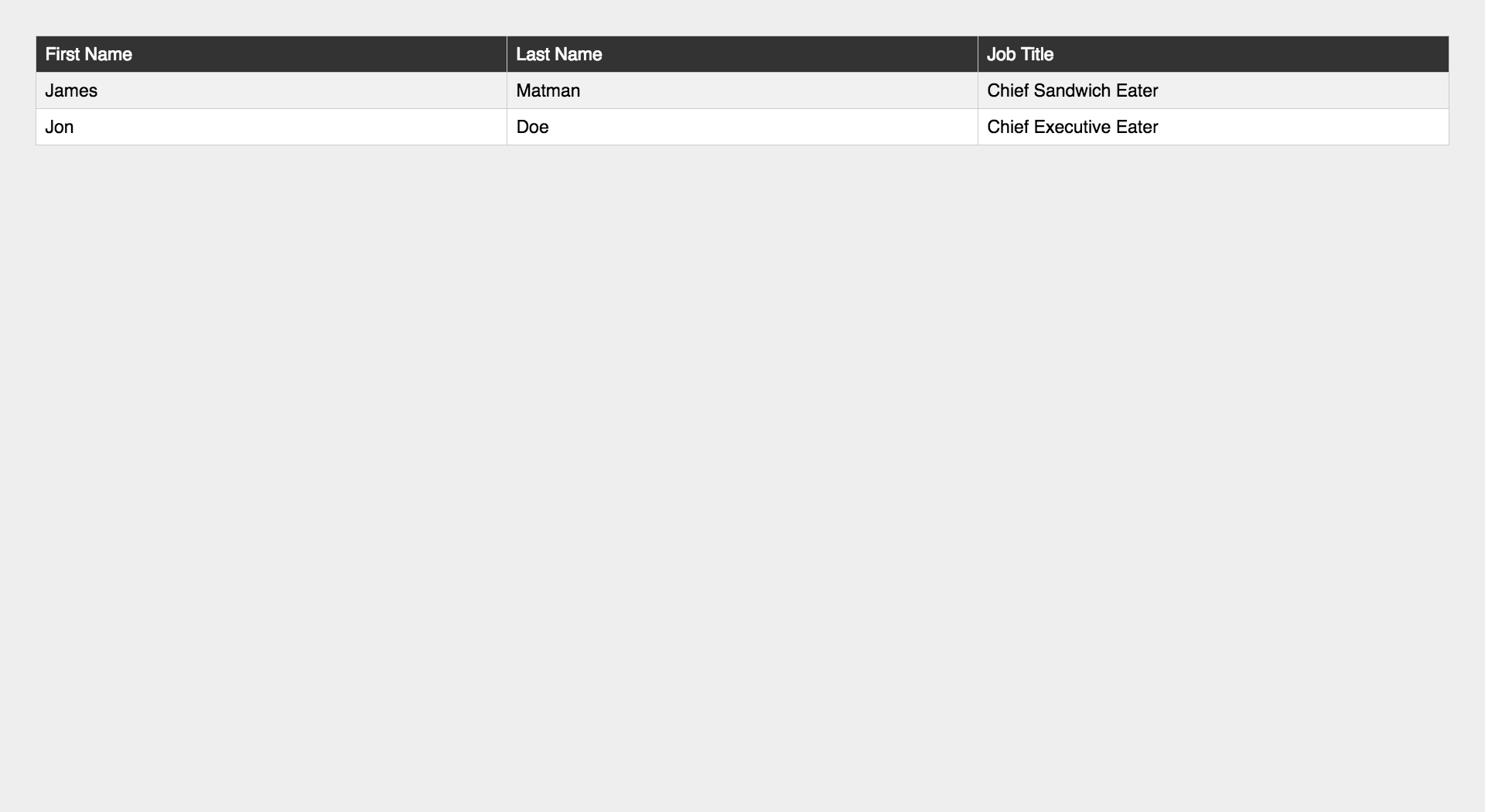
* Look at the provided screenshots in **resources** **folder** for different screen size’s view

### Requirements

* Change the document title to "**Landing Page**"
* Get the latest reset.css
* Get the latest typography.css
* Get the latest buttons.css
* Get the latest forms.css
* Create a section with a title Need a website?
* Create a section with a title Our Team
* Create a section with a title Sign-up to our Newsletter
* Create a section with a title Testimonials
* Create a footer
* Make the page **responsive** per the screenshots attached

## Responsive Tables

Create a web page like the following:



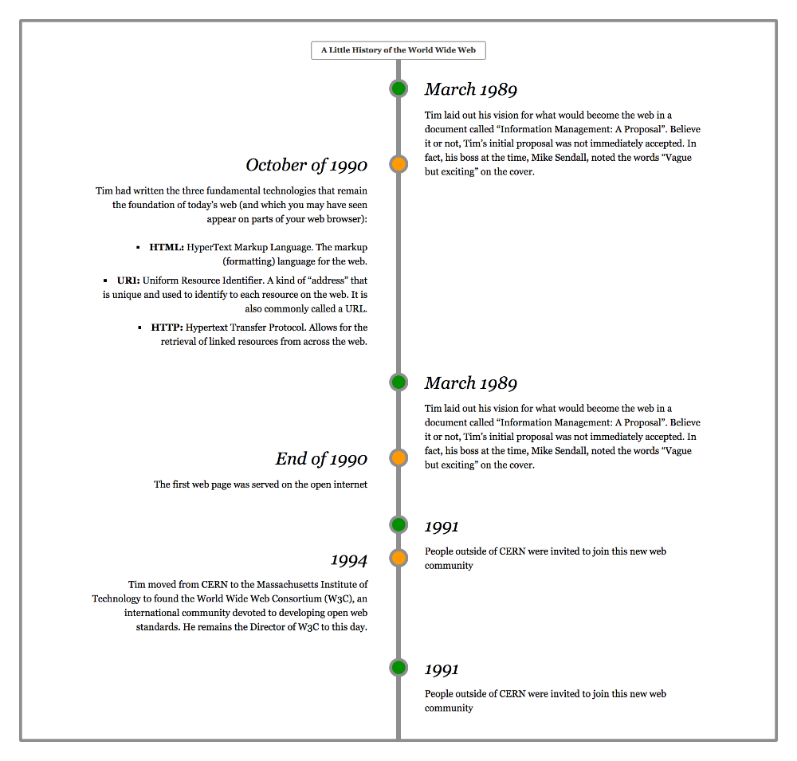
* Look at the provided screenshots in **resources** **folder** for different screen sizes view

### Requirements

* Change the document title to "**Responsive Tables**"
* Create a table with 3 columns: First Name, Last Name, Job Title
* Get the latest reset.css
* Get the latest typography.css
* Make the table **responsive**

## Responsive Vertical Timeline

Create a web page like the following:



* Look at the provided screenshots in **resources** **folder** for different screen sizes view

### Requirements

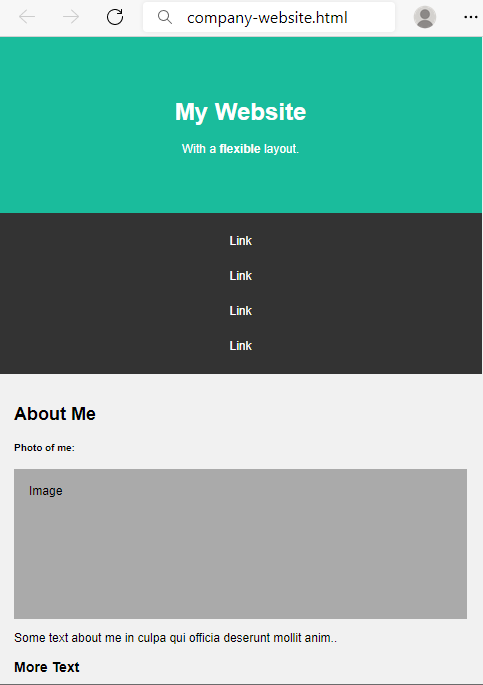
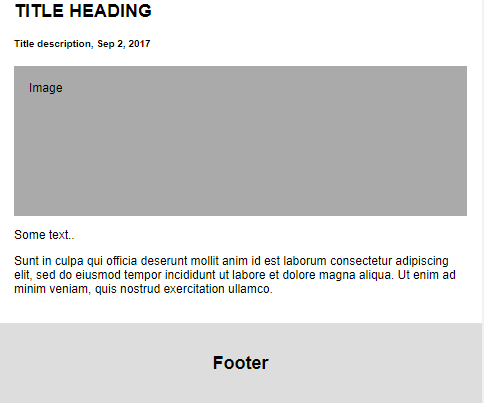
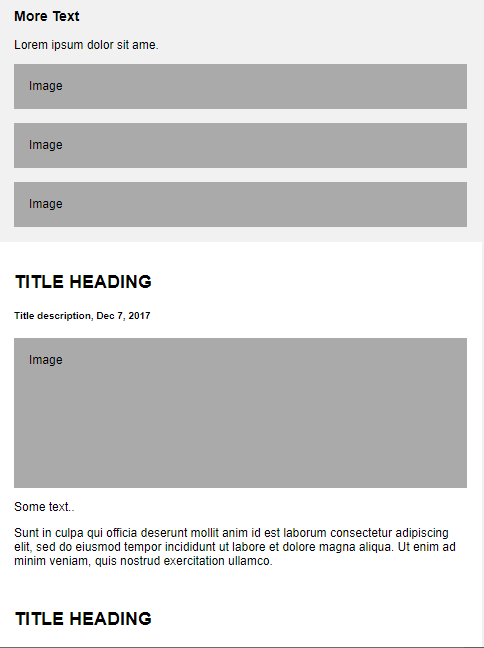
* Change the document title to "**Responsive Vertical Timeline**"
* Create a section with class timeline
  + The section must have
    - Border width - 0.3rem
    - Border color - rgb(153, 153, 153)
    - Border style - solid
* Create seven articles (article) in the section
  + Divide **article content** into header and section
  + The header has h3 heading and the section has a paragraph with text inside
  + Style h3 font to italic
* Add media queries for the list to look good on **mobile** and **desktop**

## Company Web Site

You are **given** a company web site as **HTML**. Using CSS grid, **write** the **missing** **styles**, to make the site look like this:



Ensure the site behaves correctly on small screens:

### Styles

* body
  + Margin: 0px
  + Font family: Arial
* header
  + Center the text
  + Padding: 60px
  + Margin: 0px
  + Background color: #1abc9c
  + Text color: white
* nav
  + Background color: #333
  + Top and bottom padding: 14px and 26px from left and right
* nav a
  + Text color: white
  + Top and bottom padding 14px and 20px from left and right
  + Align the text to left
  + Remove the text decoration
* aside
  + Background color: #f1f1f1
  + Padding: 20px
* main
  + Background color: white
  + Padding: 20px
* footer
  + Center the text
  + Padding: 20px
  + Background color: #ddd

### CSS Grid

Make the body grid container by displaying the grid.

* Define two grid columns and their size as ratio (**units**) 3:7.
  + For the first column 3 units and 7 units for the second.
* Define the grid areas: header, nav, aside, main, footer.

Reference the grid areas:

* the header to take the entire row
* the nav to take the entire row
* the aside to be **3 units** at the **left side of its row**
* the main to be **7 units** at the **right side of its row**, next to the aside
* the footer to take the entire row

### CSS Media Queries

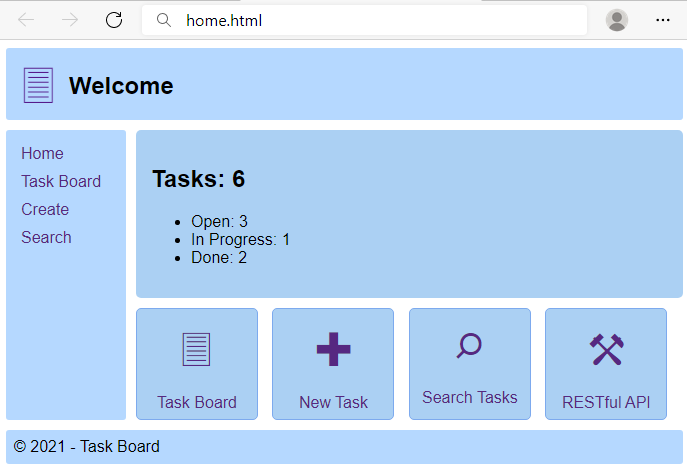
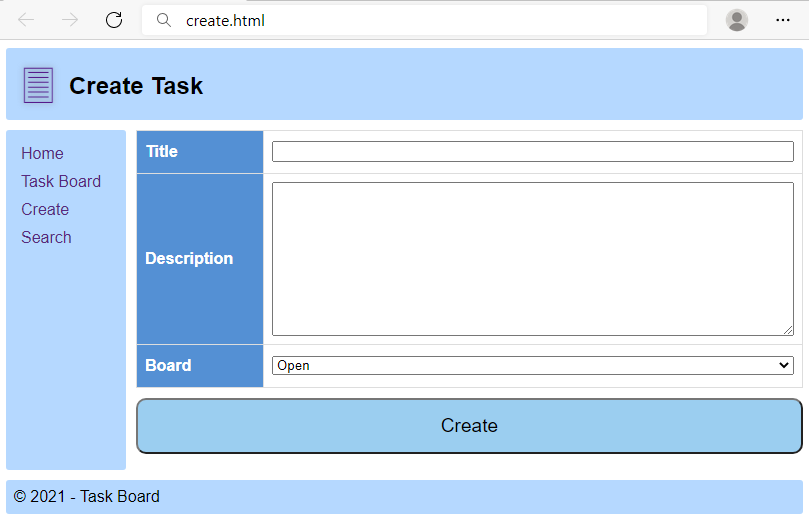
To make the layout responsive, you need **two** **media** **queries**.

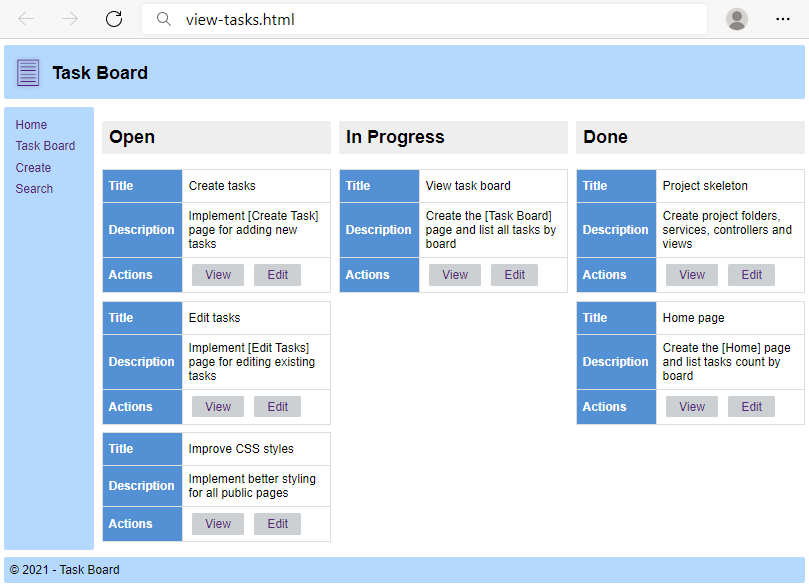
The **first** **query** concerns the rearrangement of the body**.** If the screen is **smaller** **than** 700px, display the body as block and set **the areas one after another, in that order:** header, nav, aside, main, footer.

The second query concerns the style of the navigation bar. If the screen is smaller than 700px, display the <a> tag in the nav tag as block and center the text.

## Task Board Web Site

In these exercises we shall build a **complete responsive Web site layouts** for a task board app, which looks like this:

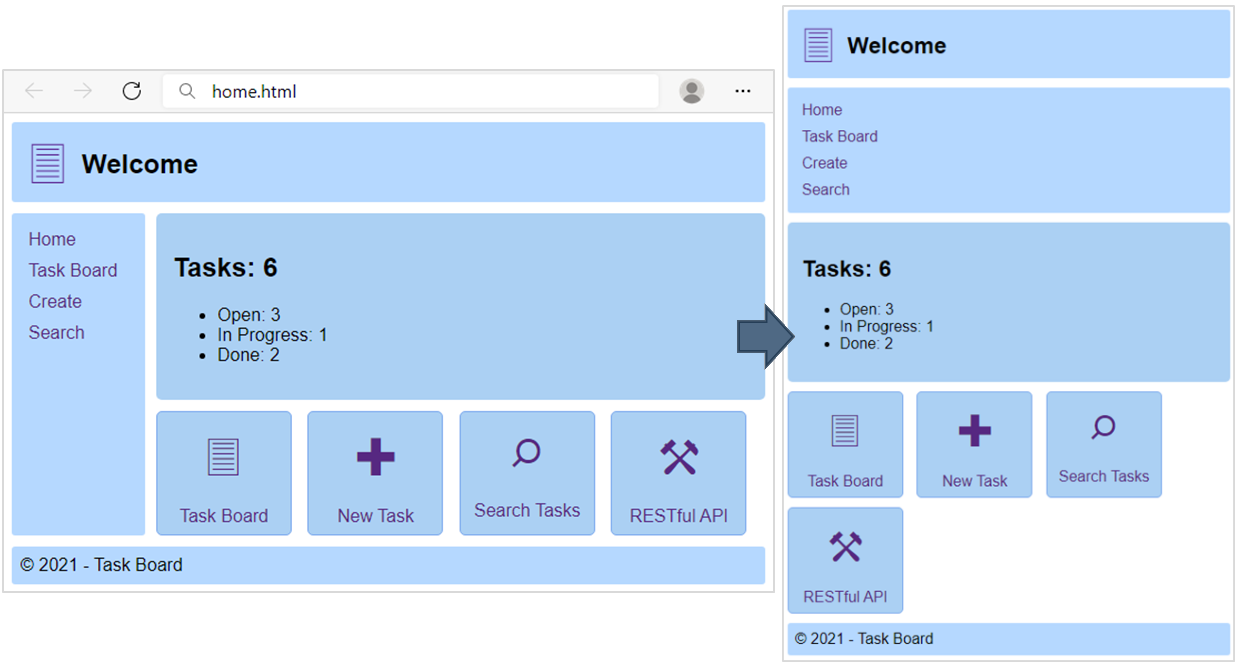
You can browse the app **source code** and play with the app at: <https://repl.it/@nakov/taskboard>.

## Task Board – "Home" Page

You will be given a **skeleton** with an **empty** **HTML** and **CSS** with styles. Your tasks are:

* create a **semantic** HTML in the empty file
* create a valid **CSS grid layout**
* make the **layout** **responsive** by using **media** **queries**

Using the layout, build the following home page:



Use the texts from the filesite-texts.txt.

### HTML Constraints

* Create body that contains: header, aside, main, footer.
* header that contains:
  + a tag with this icon: 🖹
  + h1 tag with text
* aside that contains:
  + ul tag for unordered list
    - li tag for list item
      * a tag for hyperlink
* maintag that contains:
  + section tag with:
    - h1tag for the sum of the tasks
    - ul tag with:
      * li for the tasks with their status
  + div tag with the following a tags for each icon:
    - a tag that contains:
      * span tag with this icon: 🖹
      * p tag with text “Task Board”
    - a tag that contains:
      * span tag with this icon: ✚
      * p tag with text “New Task”
    - a tag that contains:
      * span tag with this icon: ⌕
      * p tag with text “Search Tasks”
    - a tag that contains:
      * span tag with this icon: ⚒
      * p tag with text “RESTful API”
* footer that contains:
  + div tag with text © 2021 - Task Board

### CSS Grid

In this task we will define two grid containers: for the page body and for the header in the body.

#### CSS Grid for the Page Body

Make the body grid container by displaying the grid.

* Define two grid columns and their size.
  + For the first column 120px and auto for the second.
* Define the grid areas: header, aside, main, footer.
* Add **gap** between the rows and columns in the grid 10px.

Reference the grid areas:

* header to take the entire row
* aside and main to be next to each other
* the footer to take the entire row

#### CSS Grid for the Header

Make the header **in the body** to be a grid container.

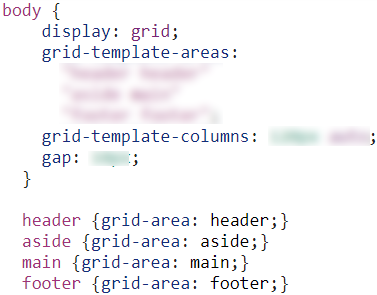
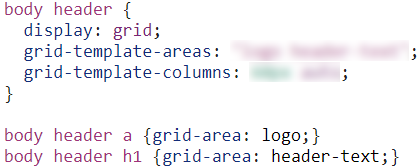
* Define two grid columns and their size.
  + For the first column 60px and auto for the second.
* Define the grid areas: logo and header-text.
* Reference the grid areas:
  + logo and header-text next to each other

### CSS Media Queries

Write a **media query** that rearranges the HTML areas in the body vertically one after another when the screen size is **below** 500px. Use the grid template areas for this purpose.

### Hints

You can use the following **grid container** and **grid area** definitions for your page layout:

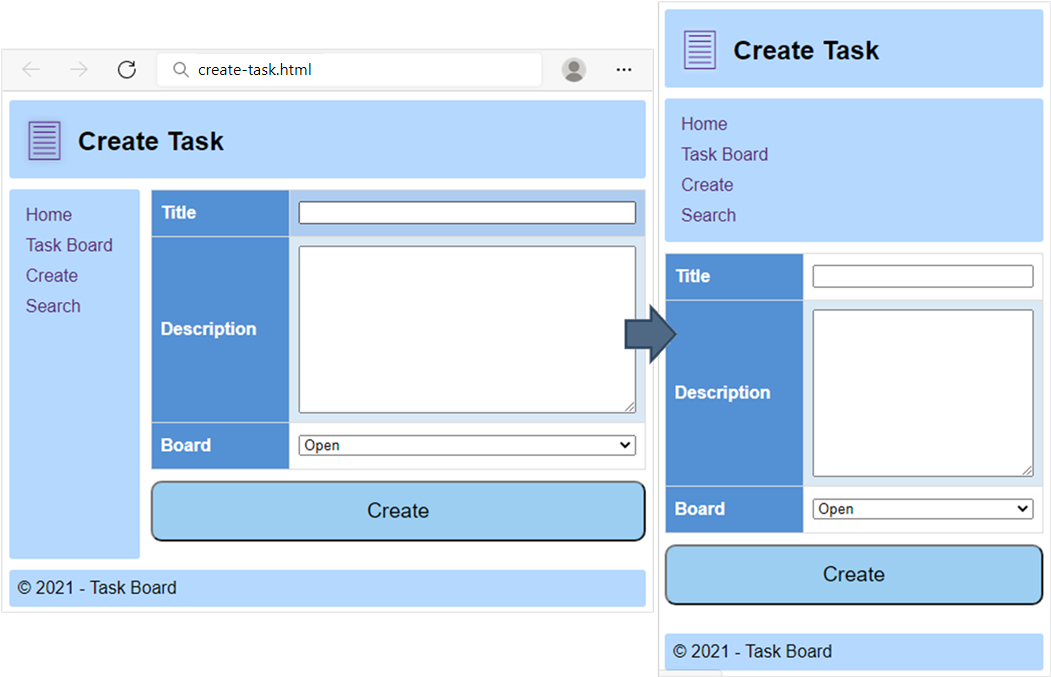
At smaller screens, you can change the **grid-template areas** (using media queries).

## Task Board – "Create Task" Page

You will be given a **skeleton** with an **empty** **HTML** and **CSS** with styles. Your tasks are:

* create a **semantic** HTML in the empty file
* create a valid **CSS grid layout**
* make the **layout** **responsive** by using **media** **queries**

Using the layout, build the following page:



Use the texts from the filesite-texts.txt.

### HTML Constraints

Like in the previous problem, create body that contains:

* header, aside, main, footer

The elements into the body:

* header: the same like in the previous problem
* aside: the same like in the previous problem
* main that contains:
  + form with table and button:
    - table with:
      * tbody tag with following rows:
        + tr with class="title" that contains:

th for the title

td with:

input field

* + - * + tr with class="description" that contains:

th for the title

td with:

textarea with 10 rows

* + - * + tr with class="board" that contains:

th for the title

td with:

select and options: Open, In Progress, Done

* + - button with type for submit
* footer: the same like in the previous problem

### CSS Grid

Like in the previous problem, define two grid containers: for the page body and for the header in the body. Define the grid areas and other grid setting to display the layout correctly.

### CSS Media Queries

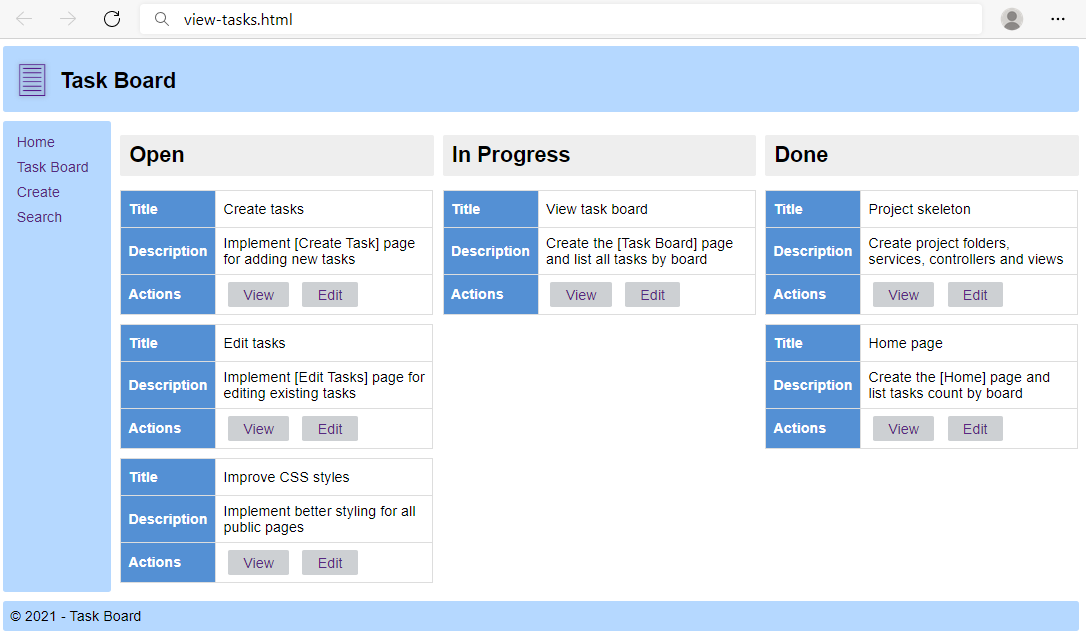
Like in the previous problem, write a **media query** that rearranges the grid areas in the **body** one after another when the screen size is **below** 500px.

## Task Board – "View Tasks" Page

You will be given a **skeleton** with an **empty** HTML and CSS with styles. Your tasks are:

* create a **semantic** HTML in the empty file
* create a valid **CSS grid layout**
* make the **layout** **responsive** by using **media** **queries**

Using the layout, build the following page:

 Ensure the site behaves correctly on small screens:

Use the texts from the filesite-texts.txt.

### HTML Constraints

Like in the previous problem, create body that contains:

* header, aside, main, footer

The elements into the body:

* header: the same like in the previous problem
* aside: the same like in the previous problem
* main contains:
  + divwith class="tasks-grid", which will serve as a grid container. It contains three **sections** arranged in columns (Open, In Progress, Done).
    - each section contains class="task-board" and:
      * h1 with one of the texts: Open, In Progress or Done
      * article for each **table** in the column:
        + table with:

tbody tag with following rows:

tr that contains:

th with text: **Title**

td for the name of the task

tr that contains:

th with text: **Description**

td with the description:

tr for the actions and buttons:

th with text: **Actions**

td with two buttons

button for the **View**

button for the **Edit**

* + - * You need to **write** **6 articles** with t**able**. Three for Open, one for In Progress and two for Done.
* footer: the same like in the previous problem

### CSS Grid

In this task we will have 3 grid containers: body, the **header** in the body and the div tasks-grid in the body.

* The **body grid container** will be the same like in the previous problem.
* The **header grid container** will be the same like in the previous problem.
* Make the div tag with class="tasks-grid" grid container by displaying the grid. You can select it like this: section.tasks-grid {/\* TODO \*/}. Define three grid columns and set their size to be three **equal** ratios (**units**). Add gap 10px.

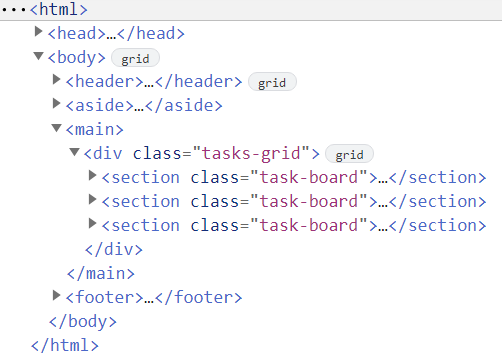
### CSS Media Queries

Like in the previous problem, write a **media query** that rearranges the grid areas in the **body** one after another when the screen size is **below** 500px.

Write a **media query** that rearranges the grid areas in the div **container** "tasks-grid", so that the boards with tasks to stay vertically one after another. Just use **one single grid column**, instead of 3 grid columns. Since we did not make references of grid areas, you can rearrange the tables by setting **new** **values** to the grid **template** **columns** CSS styling rules.

### Hints

This is how part of the HTML structure should look like:

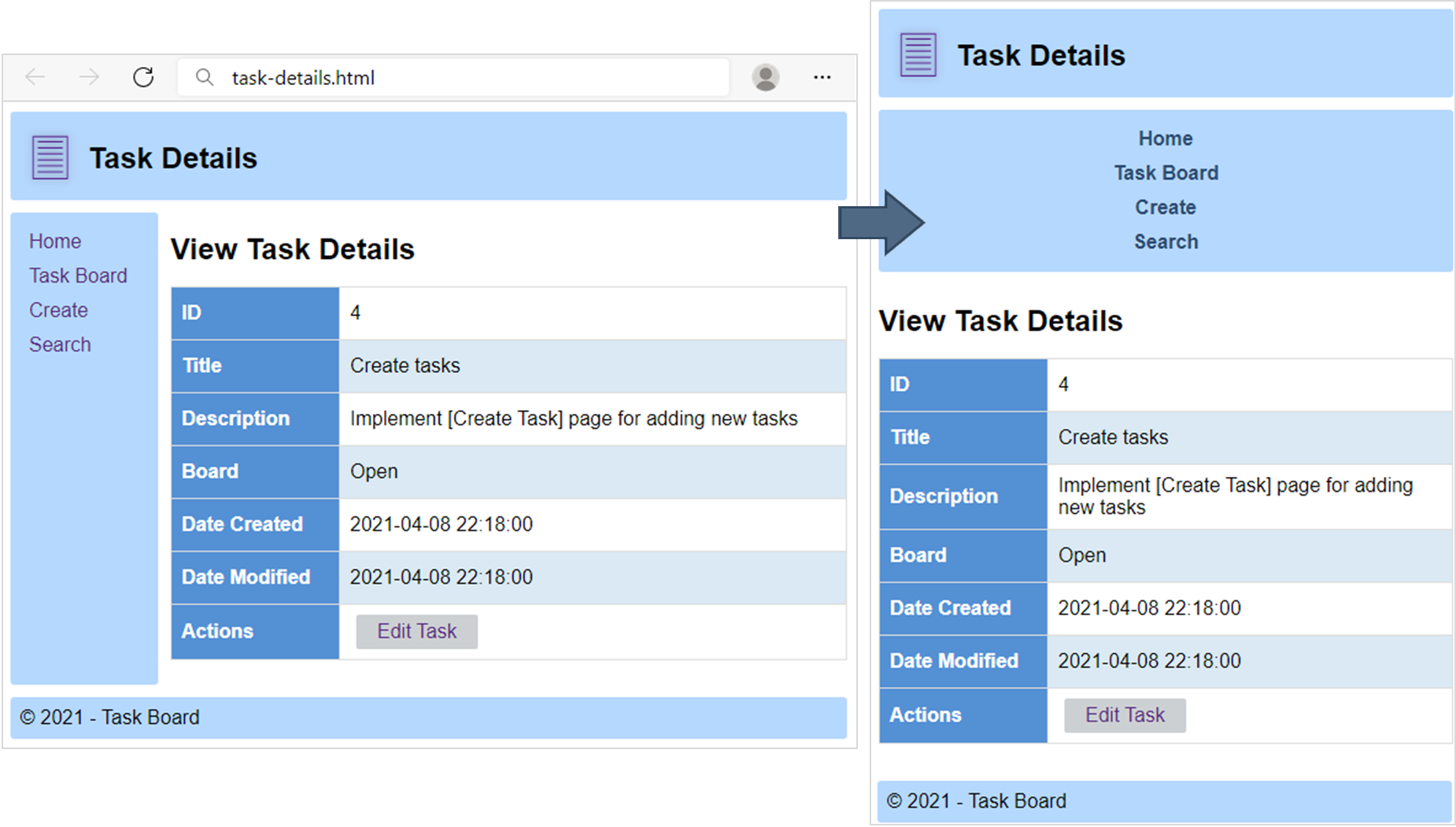


## Task Board – "Task Details" Page

You will be given a **skeleton** with an **empty** HTML and CSS with styles. Your tasks are:

* create a **semantic** HTML in the empty file
* create a valid **CSS grid layout**
* make the **layout** **responsive** by using **media** **queries**

Using the layout, build the following page:



Use the texts from the filesite-texts.txt.

### HTML Constraints

Like in the previous problem, create body that contains:

* header, aside, main, footer

The elements into the body:

* header: the same like in the previous problem
* aside: the same like in the previous problem
* main that contains:
  + h1 tag for the title
  + table tag with:
    - tbody tag with following rows:
      * tr for the ID row, with:
        + th as label
        + td with the ID
      * tr for the Title row, with:
        + th as label
        + td with the title
      * tr for the Description row, with:
        + th as label
        + td with the description
      * tr for the Board row, with:
        + th as label
        + td with the board status
      * tr for the Date Created row, with:
        + th as label
        + td with the date
      * tr for the Date Modified row, with:
        + th as label
        + td with the date
      * tr for the Actions row, with:
        + th as label
        + td with:

button for Edit Task

* footer: the same like in the previous problem

### CSS Grid

Like in the previous problem, define two grid containers: for the page body and for the header in the body. Define the grid areas and other grid setting to display the layout correctly.

### CSS Media Queries

Like in the previous problem, write a **media query** that rearranges the grid areas in the **body** one after another when the screen size is **below** 500px.

Write an additional **media query** that when the screen size is **below** 500px **arranges** the **text** of the **navigation links** from the aside area in the middle, makes the text **bold** and gives it the following **color**: rgb(44, 76, 114).