Start with the provided „index.html“ template. All files you need are included in the CSS and JS folders. Always keep in mind to apply useful classes and ids for all elements. Figure out those elements that have CSS properties in common in order to apply the same classes.

# HTML Structure



**Step 1 (Basic):**

Create three basic sections for the main-view, detail-view, sidebar and block level element for the sidebar-fade-overlay on the same level as the predefined loading overlay.

**Step 2 (Basic):**

Divide the main-view into the four different block level elements header, datepicker, content and footer.

Divide the detail-view and sidebar into the two different block level elements header and content.

**Step 3 (MainView):**

Inside the main-view-header, create one button for opening the sidebar (left) and one button for the locate-me functionality (right). Create a headline element in order to display the selected city (center).

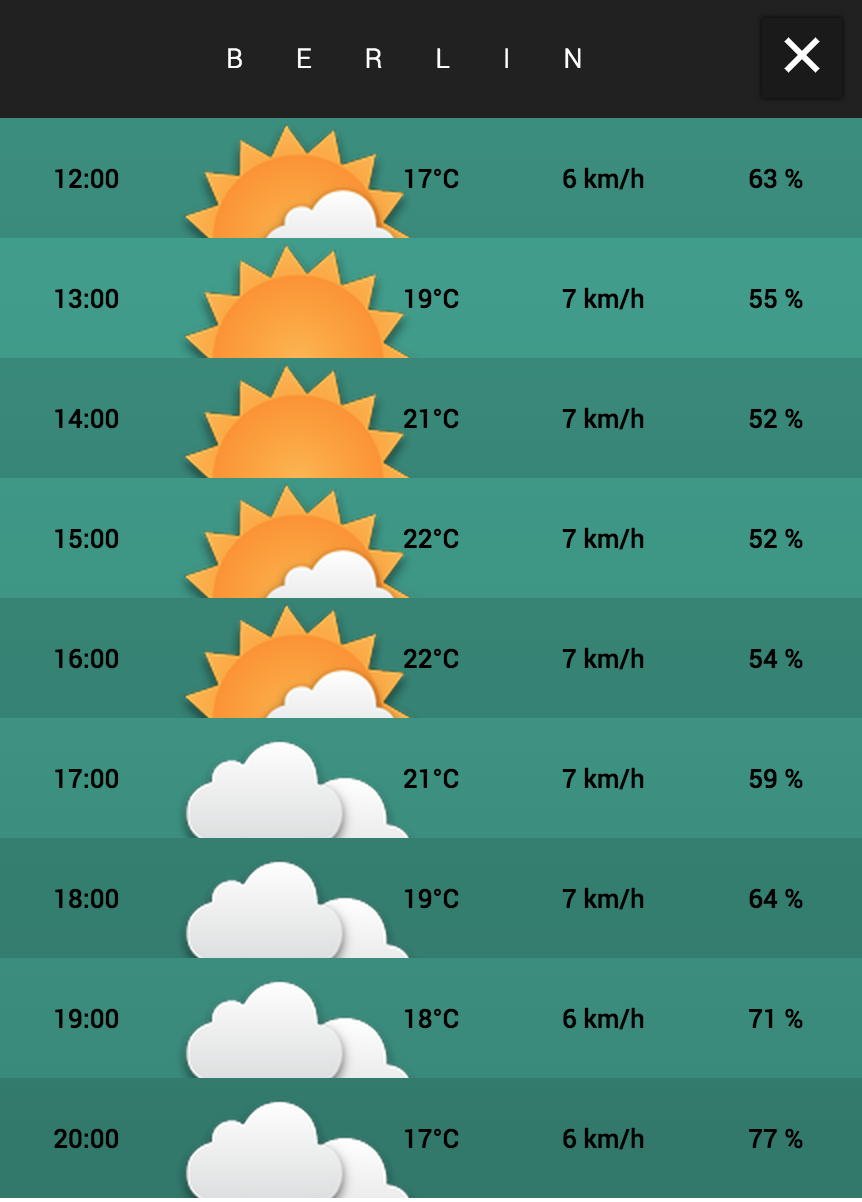
Use the same structure inside the main-view-datepicker for decreasing (left) and increasing the date (right) and displaying the current weekday (center).

Include meaningful icons (fontawesome is included) inside all button elements and add dummy text.

**Step 4 (MainView):**

Create a useful structure inside main-view-content consisting of block level and inline elements for displaying the icon image, summary text, temperature and additional information like wind speed and humidity. Add dummy text and choose a dummy weather icon (predefined in img folder).

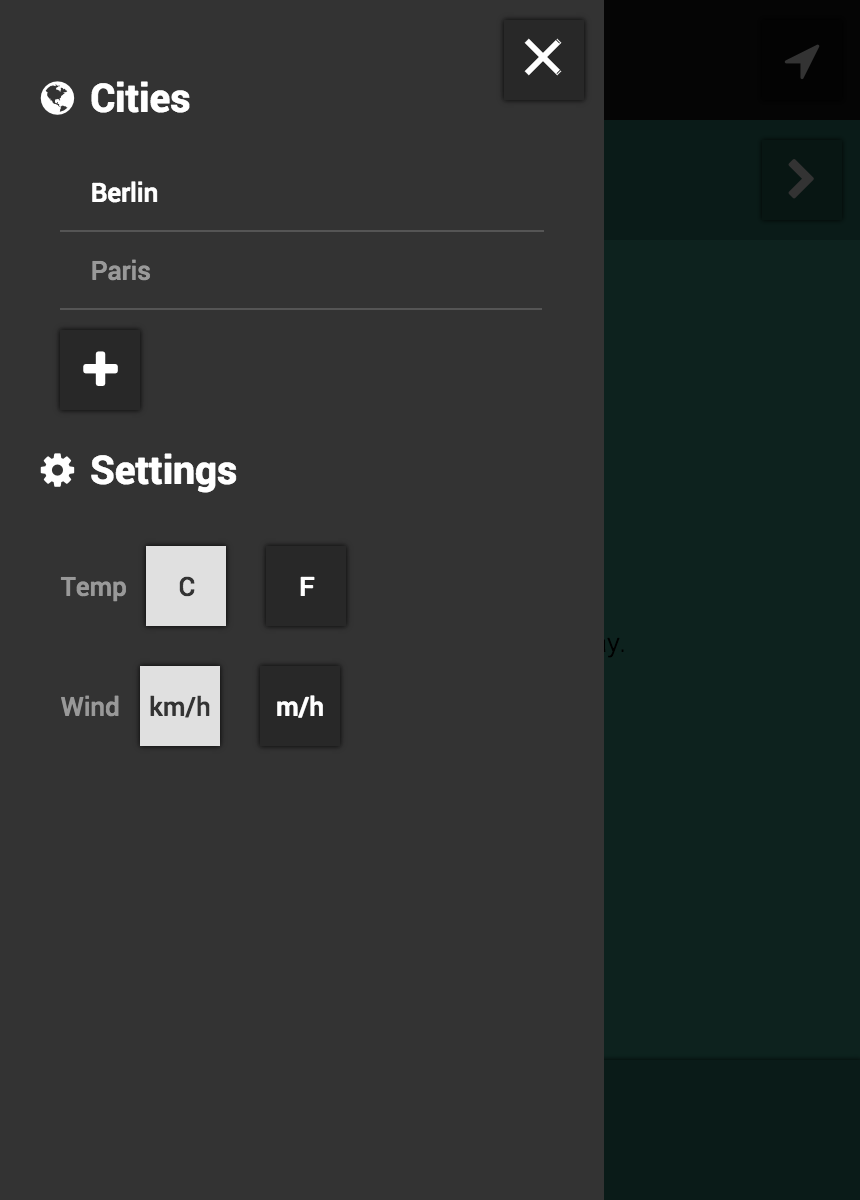
Inside the main-view-footer, provide a button to open the detail-view.



**Step 5 (DetailView):**

Inside the detail-view-header, create a headline element in order to display the selected city (center) and a button (right) to close the detail-view.

Create a table construction inside the detail-view-content without using the table element. Provide elements for time, icon, temperature, wind speed and humidity. Insert four rows with dummy content.



**Step 6 (Sidebar):**

Use unordered list to create a citylist and settingslist and provide headlines with icons.

For the citylist, additionally include a text input element with button. Add dummy content.

For the settingslist provide an inline description and two buttons in each list element. Active classes will be added later by JS.

# CSS Layout & Positioning

Work with the provided „main.css“ template to style your app. Always try to use appropriate CSS selectors that only affect those elements which are supposed to be affected. Too imprecise selectors may cause disarrangement of your elements. If necessary, improve your html structure by adding new classes or ids.

**Step 7 (MainView):**

* Use fixed heights for the main-view element (e.g. 600px), main-view-header, main-view-datepicker (both e.g. 40px) and main-view-footer (60px). Apply a width of 100% to all elements.
* Use relative positioning for all buttons and absolute positioning for the headlines inside main-view-header and main-view-datepicker.
* Use a fixed width and height for buttons in main-view-header and main-view-datepicker.
* Use relative positioning for all elements inside main-view-content. Set appropriate widths and center all elements or rather group of elements.
* Use absolute positioning to pin the main-view-footer to the bottom. Set a fixed height and full width to the main-view-footer button.

**Step 8 (DetailView):**

* Use same heights and widths for the detail-view elements as used before for the main-view. Set a fixed height for the detail-view-content (height detail-view – height detail-view-header) and make it scrollable by using the property overflow.
* Use absolute positioning for the detail-view-header to pin it to the top.
* Set a fixed height for all table rows and divide the table rows into columns using percental widths.

**Step 9 (Sidebar):**

* Use same heights for sidebar and sidebar-fade-overlay as used before.
* Use a width of 70% for the sidebar and a dark background-color.
* Use a width of 100% for the sidebar-fade-overlay and add a dark half-transparent background-color.

# CSS Styling

Now add styling to your WeatherApp layout.

**Step 10 (MainView):**

* Add a dark, half-transparent background for all buttons, main-view-datepicker and main-view-footer.
* Create :hover and :active styles for all buttons to highlight the user interaction.
* Use font properties to style color and size of the icons and appearance of headlines.
* Use padding & margin to set enough space between the elements.
* Provide a background-color transition for the main-view element when the color is changed. Test the transition with your browser developer tools by changing the background-color manually.

**Step 11 (DetailView):**

* Use the CSS selector :nth-child() to set a half-transparent darker background-color to odd rows and lighter background-color to even rows.
* If the button classes where applied correctly in the HTML part, the close button and header should be already styled.
* Create a half-transparent background gradient for the table.

**Step 12 (Sidebar):**

* Edit the color and size of the headlines and icons.
* Style the lists with margins, paddings and borders.
* If the button classes where applied correctly in the HTML part, all buttons should be already styled. Create an active class to highlight currently selected buttons.
* Add a :hover class to the city select list item using the image draggable.png inside the img folder.

# CSS Final Positioning

This step hides all sections except the main-view. Make sure that all CSS Styling is completed before you proceed.

**Step 13:**

* Use absolute positioning for all sections and overlays:
  + loading overlay: top 0 left 0
  + main-view: top 0 left 0
  + detail-view: top 100% left 0
  + sidebar: top: 0 left -70%
  + sidebar fade overlay: top 0 left 0
* Use z-indices to arrange all sections.
* Hide all overlays with display property.

# JavaScript

**Step 14:**

* Open the main.js file in your js/ folder
* Add an onclick-Handler for #open-sidebar and #close-sidebar HTML elements
* Your event functions should show and hide #sidebar by manipulating the css property ‘display’.
* Add another onclick-Handler for the Details-Button. Give the button an id-attribute in order to get access to the DOM element.
* The event function should toggle the display view. Use an if..else statement to show the view if its hidden and to hide it if its shown

Toggle sidebar, Toggle Details view - Javascript

7) UpdateWeatherMainHTML()- Javascript innerHTML, Objects

8) UpdateWeatherDetailHTML()- Javascript HTML String creation

9) ChangeCurrentDay() – Javascript Logik, DateStrings

10) UpdateColorScheme() – Javascript CSS Manipulation

11) ConvertTemperature, Wind, Humidity Functions – Javascript Functions, Returns

12) getAPIData() – jQuery, getJSON

13) Show / Hide Loading overlay - CSS Transitions, Animations

14) ChangeTempUnit, changeWindUnit – jQuery

15) Show city List - jQuery DOM Insertion

16) addCity + Sortable - jQuery Interaction, jQueryUI Widgets

17) LocateMe, getLocation, getCoordinates, getCityname – Javascript Geolocation, Geocoding

18) setCityClickable - jQuery Plugins