import “awesomplete/awesomplete.css”; import Awesomplete from “awesomplete”;

// List of countries. Basically copied from the dataset cleanup notebook const countries = [“United States”, “India”, “Germany”, “United Kingdom”, “Canada”, “France”, “Brazil”, “Poland”, “Netherlands”, “Australia”, “Italy”, “Spain”, “Russia”, “Sweden”, “Switzerland”, “Turkey”, “Ukraine”, “Austria”, “Israel”, “Pakistan”, “Czech Republic”, “Belgium”, “Romania”, “Iran”, “China”, “Portugal”, “Greece”, “Finland”, “Norway”]

/** * Registers and initializes filter UI components. This function reads the current query string (for the parameters: * `country`, `age`, and `bubbleSetActive`) and pre-populates the respective * input elements. It then registers event listeners on these elements so * that any user changes update the query parameters in the URL via the History API. * The country input uses Awesomplete to enable autocomplete functionality. @returns {void} */ export const registerFilters = () => { const params = new URLSearchParams(window.location.search) const country = params.get(“country”) || null const age = params.get(“age”) || null const bubbleSetActive = params.get(“bubbleSetActive”) || null

```
const countryAutocompleteInput = document.getElementById("filter_country");
countryAutocompleteInput.value = country

const ageSelect = document.getElementById('filter_age')
ageSelect.value = age

const bubbleSetActiveInput = document.getElementById('filter_bubbleset')
bubbleSetActiveInput.checked = Boolean(bubbleSetActive)

new Awesomplete(countryAutocompleteInput, {
    list: countries,
    minChars: 1,
    autoFirst: true,

});

countryAutocompleteInput.addEventListener('awesomplete-selectcomplete', (event) => {
    event.preventDefault()
    const params = new URLSearchParams(window.location.search);
    if (event.target.value) {
        params.set('country', event.target.value);
    } else {
        params.delete('country')
    }

    window.history.pushState({}, '', `${window.location.pathname}?${params.toString()}`);
})

countryAutocompleteInput.addEventListener('blur', (event) => {
    event.preventDefault()
    if (!event.target.value) {
        params.delete('country')
        window.history.pushState({}, '', `${window.location.pathname}?${params.toString()}`);
    }
})

ageSelect.addEventListener('change', (event) => {
    event.preventDefault()
    const params = new URLSearchParams(window.location.search);
    if (event.target.value) {
        params.set('age', event.target.value);
    } else {
        params.delete('age')
    }

    window.history.pushState({}, '', `${window.location.pathname}?${params.toString()}`);
})

bubbleSetActiveInput.addEventListener('change', (event) => {
    event.preventDefault()
    const params = new URLSearchParams(window.location.search);
    if (event.target.checked) {
        params.set('bubbleSetActive', 'true');
    } else {
        params.delete('bubbleSetActive')
    }

    window.history.pushState({}, '', `${window.location.pathname}?${params.toString()}`);
})
```

}

// analytics

if (import.meta.env.PROD) { const ageSelect = document.getElementById(“filter_age”); const countryInput = document.getElementById(“filter_country”); const bubbleSetToggle = document.getElementById(“filter_bubbleset”);

```
ageSelect.addEventListener("change", () => {
    gtag('event', 'filter_used', {
        filter_type: 'age',
        filter_value: ageSelect.value || 'all'
    });
});

countryInput.addEventListener("change", () => {
    gtag('event', 'filter_used', {
        filter_type: 'country',
        filter_value: (countryInput.value || '').split(' ', '_') || 'all'
    });
});

bubbleSetToggle.addEventListener("change", () => {
    gtag('event', 'filter_used', {
        filter_type: 'bubbleset',
        filter_value: bubbleSetToggle.checked ? 'enabled' : 'disabled'
    });
});
```

}