notes-app index.html

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<h1>Notes App</h1>

<h2>Take notes and never forget</h2>

<input id="search-text" type="text" placeholder="Filter notes">

<select id="filter-by">

<option value="byEdited">Sort by last edited</option>

<option value="byCreated">Sort by recently created</option>

<option value="alphabetical">Sort alphabetically</option>

</select>

<div id="notes"></div>

<button id="create-note">Create Note</button>

<script src="uuidv4.js"></script>

<script src="notes-function.js"></script>

<script src="notes-app.js"></script>

</body>

</html>

*notes-app.js*

*let* notes = getSavedNotes()

*const* filters = {

searchText: ''

}

renderNotes(notes, filters)

document.querySelector('#create-note').addEventListener('click', *function* (*event*) {

*const* id = uuidv4()

notes.push({

id: id,

title: '',

body: ''

})

savedNotes(notes)

location.assign(`/edit.html#${id}`)

})

document.querySelector('#search-text').addEventListener('input', *function*(*event*) {

filters.searchText = *event*.target.value

renderNotes(notes, filters)

})

document.querySelector('#filter-by').addEventListener('change', *function*(*event*) {

console.log(*event*.target.value)

})

window.addEventListener('storage', *function* (*event*) {

if (*event*.key === 'notes') {

notes = JSON.parse(*event*.newValue)

renderNotes(notes, filters)

}

})

notes-function.js

// Read existing notes from localStorage

*const* getSavedNotes = *function*() {

*const* notesJSON = localStorage.getItem('notes')

if (notesJSON !== null) {

return JSON.parse(notesJSON)

} else {

return []

}

}

// Save the notes to localStorage

*const* savedNotes = *function* (*notes*) {

localStorage.setItem('notes', JSON.stringify(*notes*))

}

// Remove a note from the list

*const* removeNotes = *function* (*id*) {

*const* noteIndex = notes.findIndex(*function* (*note*) {

return *note*.id === *id*

})

if (noteIndex > -1) {

notes.splice(noteIndex, 1)

}

}

// Generate the DOM structure for a note

*const* generateNoteDom = *function* (*note*) {

*const* noteEl =document.createElement('div')

*const* textEl =document.createElement('a')

*const* button =document.createElement('button')

// Setup the remove note button

button.textContent = 'x'

noteEl.appendChild(button)

button.addEventListener('click', *function* () {

removeNotes(*note*.id)

savedNotes(notes)

renderNotes(notes, filters)

})

// Setup the note title text

if (*note*.title.length > 0) {

textEl.textContent = *note*.title

} else {

textEl.textContent = 'Unnamed note'

}

textEl.setAttribute('href', `/edit.html#${*note*.id}`)

noteEl.appendChild(textEl)

return noteEl

}

// Render application notes

*const* renderNotes = *function* (*notes*, *filters*) {

*const* filteredNotes = *notes*.filter(*function* (*note*) {

return *note*.title.toLowerCase().includes(*filters*.searchText.toLowerCase())

})

document.querySelector('#notes').innerHTML = ''

filteredNotes.forEach(*function* (*note*) {

*const* noteEl = generateNoteDom(*note*)

document.querySelector('#notes').appendChild(noteEl)

})

}

*notes-edit.js*

*const* titleElement =document.querySelector('#note-title')

*const* bodyElement =document.querySelector('#note-body')

*const* removeElement =document.querySelector('#remove-note')

*const* noteId = location.hash.substring(1)

*let* notes = getSavedNotes()

*let* note = notes.find(*function* (*note*) {

return *note*.id === noteId

})

if (note === undefined) {

location.assign('/index.html')

}

titleElement.value = note.title

bodyElement.value = note.body

titleElement.addEventListener('input', *function*(*event*) {

note.title = *event*.target.value

savedNotes(notes)

})

bodyElement.addEventListener('input', *function* (*event*) {

note.body = *event*.target.value

savedNotes(notes)

})

removeElement.addEventListener('click', *function* (*event*) {

removeNotes(noteId)

savedNotes(notes)

location.assign('/index.html')

})

window.addEventListener('storage', *function*(*event*) {

if (*event*.key === 'notes') {

notes = JSON.parse(*event*.newValue)

*const* note = notes.find(*function* (*note*) {

return *note*.id === noteId

})

if (note === undefined) {

location.assign('/index.html')

}

titleElement.value = note.title

bodyElement.value = note.body

}

})

[https://account.mapbox.com/confirm-email/?account=gbsoremekun&token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlbWFpbCI6Imdic29yZW1la3VuQGdtYWlsLmNvbSIsImV4cCI6MTYwODk1MTk5OX0.DY2uMbc3yBIp97AY\_uXmFonR1OtVvuNUJgzm2VlOWmw](https://account.mapbox.com/confirm-email/?account=gbsoremekun&token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlbWFpbCI6Imdic29yZW1la3VuQGdtYWlsLmNvbSIsImV4cCI6MTYwODk1MTk5OX0.DY2uMbc3yBIp97AY_uXmFonR1OtVvuNUJgzm2VlOWmw&utm_campaign=New+signups+-+verify+account%27s+email+address&utm_content=Signup+Confirmation+Email&utm_medium=email_action&utm_source=customer.io)

If this link has expired, [log in to your account](https://account.mapbox.com/auth/signin/?utm_campaign=New+signups+-+verify+account%27s+email+address&utm_content=Signup+Confirmation+Email&utm_medium=email_action&utm_source=customer.io) to resend your verification link. Please [contact support](https://docs.mapbox.com/accounts/guides/settings/?utm_campaign=New+signups+-+verify+account%27s+email+address&utm_content=Signup+Confirmation+Email&utm_medium=email_action&utm_source=customer.io#troubleshooting/?utm_source=confirmemail&utm_medium=email&utm_content=support-link&utm_campaign=confirmemail) with questions about email verification. Responses to this email are not monitored.

|  |  |
| --- | --- |
|  | ReplyForward |

https://api.mapbox.com/geocoding/v5/mapbox.places/Los%20Angeles.json?access\_token=pk.eyJ1IjoiZ2Jzb3JlbWVrdW4iLCJhIjoiY2tqM3AxY2ZpMjl6OTJ5bTA3MTJuYWN3bCJ9.gQo2Cps5SBIGUpLWl6SSQQ "