Internet Programming

**(Pagination, Local data sync)**

**Name: Sophart Chhordaphea**

**ID: e20191055**

**Group: I4 GIC C**

Link: <https://github.com/chhordaphea-27/I4-TP_Internet_Programming/tree/TP6>

Table of Contents

[Exercise 1: Pagination (infinite scroll) 2](#_Toc130851483)

[1. Source Code 2](#_Toc130851484)

[1.1. Index.html 2](#_Toc130851485)

[1.2. Styles.css 2](#_Toc130851486)

[1.3. Script.js 4](#_Toc130851487)

[2. Result 6](#_Toc130851488)

[Exercise 2: Local data syncIndex.html 6](#_Toc130851489)

[1. Source Code 6](#_Toc130851490)

[1.1. Index.html 6](#_Toc130851491)

[1.2. app.js 7](#_Toc130851492)

[2. Result 9](#_Toc130851493)

[Exercise 3: List and Viewer 10](#_Toc130851494)

[1. Source Code 10](#_Toc130851495)

[1.1. Index.html 10](#_Toc130851496)

[1.2 Styles.css 11](#_Toc130851497)

[1.3 Script.js 13](#_Toc130851498)

[2 Result 16](#_Toc130851499)

# **Exercise 1: Pagination (infinite scroll)**

## Source Code

### Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Pagination</title>

<link rel="stylesheet" href="styles.css">

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">

</head>

<body onload="fetchData()">

<p id="pageNumber">Page: 1</p>

<div id="scoller-wrapper" onscroll="containerScroll()">

<div id="loading" hidden>

<div class="lds-ring"><div></div><div></div><div></div><div></div></div>

</div>

</div>

</body>

</html>

### Styles.css

body {

font-family: 'Roboto', sans-serif;

font-size: medium;

}

#scoller-wrapper {

/\* display: inline-block; \*/

width: 90%;

height: 93vh;

margin: auto;

overflow-y: auto;

background-color: rgb(151, 151, 151);

}

#pageNumber {

/\* display: inline-block; \*/

}

.data {

width: 90%;

height: 100px;

background-color: white;

padding: 21px;

margin: auto;

margin-top: 10px;

border-radius: 7px;

}

.data h6 {

display: inline;

font-size: x-large;

}

.data p {

display: inline;

font-size: x-large;

}

#loading {

text-align: center;

}

.lds-ring {

display: inline-block;

position: relative;

width: 80px;

height: 80px;

}

.lds-ring div {

box-sizing: border-box;

display: block;

position: absolute;

width: 64px;

height: 64px;

margin: 8px;

border: 8px solid #fff;

border-radius: 50%;

animation: lds-ring 1.2s cubic-bezier(0.5, 0, 0.5, 1) infinite;

border-color: #ffffff transparent transparent transparent;

}

.lds-ring div:nth-child(1) {

animation-delay: -0.45s;

}

.lds-ring div:nth-child(2) {

animation-delay: -0.3s;

}

.lds-ring div:nth-child(3) {

animation-delay: -0.15s;

}

@keyframes lds-ring {

0% {

transform: rotate(0deg);

}

100% {

transform: rotate(360deg);

}

}

### Script.js

let pageNumber = 1;

const numberPerPage = 20;

let dataLoading = false;

let currentPage = 1;

let scrollWrapper = null;

let pageHeight = null;

function fetchData() {

scrollWrapper = document.getElementById("scoller-wrapper");

const loading = document.getElementById("loading");

loading.hidden = false;

fetch(

`https://api.instantwebtools.net/v1/passenger?page=${pageNumber}&size=${numberPerPage}`

).then(async (response) => {

let datas = await response.json();

loading.hidden = true;

datas.data.forEach((element) => {

scrollWrapper.innerHTML += renderData(element);

});

pageNumber++;

dataLoading = true

});

}

function renderData(data) {

let dataCell = `

<div class="data">

<h6>✈️: </h6>

<h6>${data.airline[0].name} - ${data.airline[0].country}</h6>

<br>

<p>🙂: </p>

<p>${data.name}</p>

</div>

`;

return dataCell;

}

function containerScroll() {

const pageNumberEle = document.getElementById("pageNumber")

const { scrollTop, clientHeight, scrollHeight } = scrollWrapper;

if (pageHeight == null) {pageHeight = scrollHeight}

// console.log(scrollWrapper.scrollTop)

if (clientHeight + scrollTop >= scrollHeight \* 0.95) {

pageNumber += 1;

if (dataLoading) {

fetchData(pageNumber);

console.log("loading")

}

dataLoading = false

}

let page = Math.round((scrollTop) / pageHeight) + 1

pageNumberEle.innerHTML = `Page: ${page}`

console.log(page)

}

## Result



# **Exercise 2: Local data sync**Index.html

## Source Code

### Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Local data sync</title>

<script src="./assets/tailwindcss.js"></script>

<style>

@import url('https://fonts.googleapis.com/css2?family=Julee&family=Lato:ital,wght@1,100&family=Oswald:wght@200&family=Raleway:wght@800&family=Roboto:wght@100&family=Shadows+Into+Light&display=swap');

</style>

</head>

<body>

<div class="w-[500px] h-[80vh] p-6 absolute top-[50%] left-[50%] border-2

translate-x-[-50%] translate-y-[-50%] rounded-lg">

<div class="h-full rounded-lg overflow-y-scroll" id="scroller">

</div>

</div>

<script src="./assets/app.js"></script>

</body>

</html>

### app.js

let scrollerEle = document.getElementById("scroller");

let pageNum = 0;

let isDataLoading = false;

var passengers = localStorage.getItem("passengers") != null ? JSON.parse(localStorage.getItem("passengers")) : [];

// input as array

function displayAPage(dataAPage) {

dataAPage.forEach(element => {

scrollerEle.innerHTML += `

<div class="w-full border-2 rounded-lg px-5 py-2 mb-2 bg-[#F4F4F4]">

<div class="font-bold">✈️ : ${element.airline[0].name} - ${element.airline[0].country}</div>

<div>😊 : ${element.name}</div>

</div>

`

});

}

function fetchDataFromLocal(pageNumber) {

displayAPage(passengers[pageNumber]);

}

function fetchData(pageNumber) {

scrollerEle.innerHTML += ` <div class="font-bold text-[1.5rem] px-2">${pageNumber + 1}</div>`;

pageNum >= passengers.length ? fetchAPI(pageNumber) : fetchDataFromLocal(pageNumber);

pageNum++;

}

function fetchAPI(pageNumber) {

isDataLoading = true;

fetch(`https://api.instantwebtools.net/v1/passenger?page=${pageNumber}&size=15`)

.then(async (res) => {

let dataInfors = await res.json();

passengers.push(dataInfors.data);

saveToLocalstorage(passengers);

displayAPage(dataInfors.data);

isDataLoading = false;

})

.catch((err) => {

alert(err);

});

}

const saveToLocalstorage = (data) => {

localStorage.setItem("passengers", JSON.stringify(data));

}

fetchData(pageNum);

scrollerEle.addEventListener("scroll", () => {

const { scrollTop, clientHeight, scrollHeight } = scrollerEle;

if ((clientHeight + scrollTop) >= (scrollHeight \* 0.95)) {

if (!isDataLoading) {

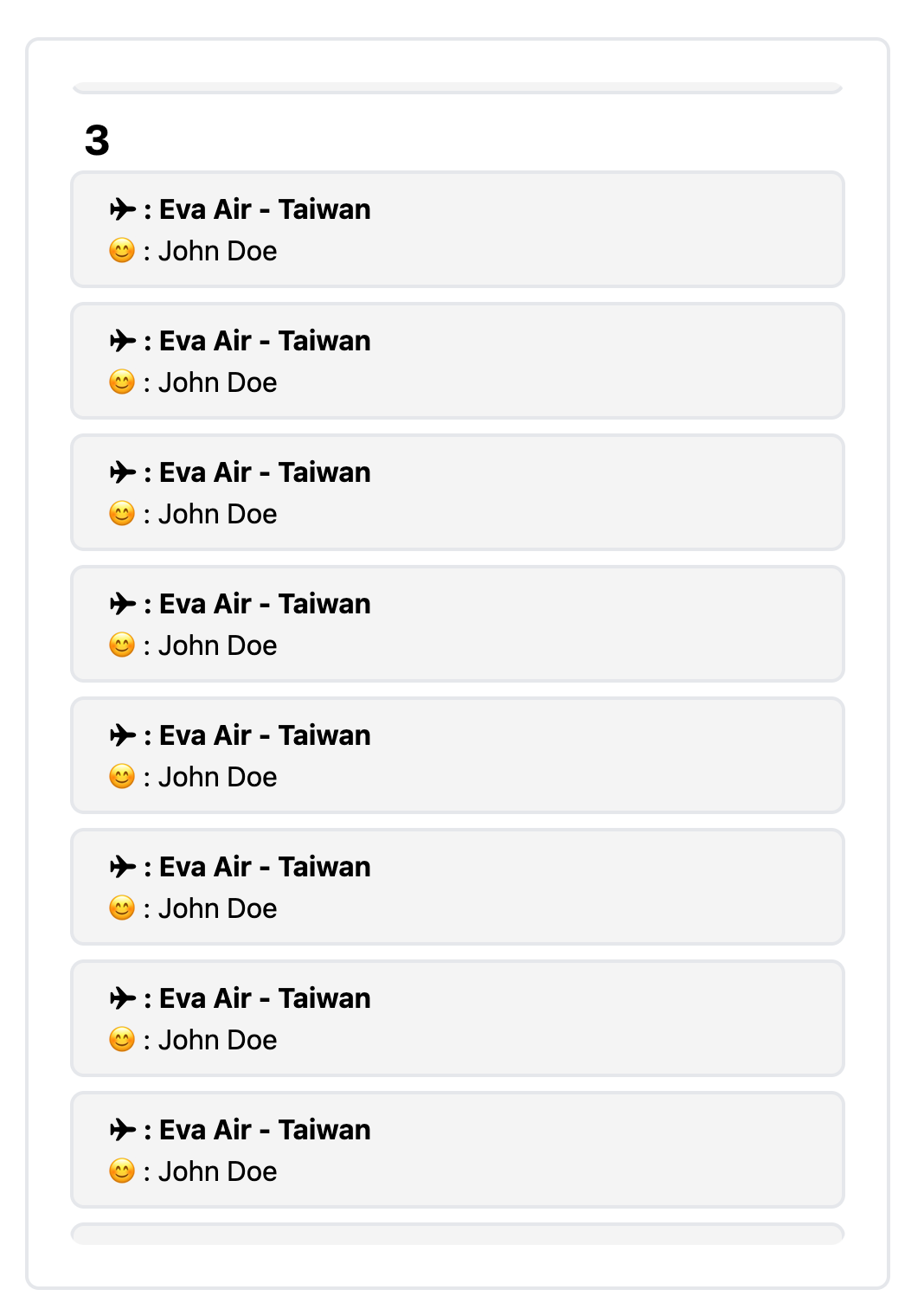
fetchData(pageNum);

}

}

});

## Result



# **Exercise 3: List and Viewer**

## Source Code

### Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="./styles.css">

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">

</head>

<body>

<h1>Find your favorite 🥂</h1>

<br>

<div class="wrapper">

<div id="dataList">

<!-- <div id="dataCell" onclick="">

<img src="http://cleanswifter.com/wp-content/uploads/2016/07/beer.png" alt="" srcset="">

<div id="info">

<h2>Arcade Nation</h2>

<p>The brainchild of our. small batch brewer...</p>

</div>

</div> -->

</div>

<div class="dataDetail">

<img src="" alt="" srcset="" id="beerImage">

<h2 id="selectedName">Arcade Nation</h2>

<p id="date">04/2013</p>

<p id="detailInfo">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Corrupti natus suscipit sequi tempora vero incidunt fuga quibusdam repellat cupiditate molestiae!</p>

<br>

<p>contributor</p>

<p><b>(Sam Mason)</b></p>

</div>

</div>

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

</body>

</html>

### Styles.css

body {

text-align: center;

font-family: 'Roboto', sans-serif;

/\* width: 50%; \*/

margin: auto;

display: table;

}

h1 {

margin-bottom: 10px;

}

.wrapper {

display: grid;

grid-template-columns: auto auto;

}

/\* Left-Side \*/

#dataList {

background-color: #cfcfcf;

overflow-x: auto;

width: 450px;

height: 89vh;

padding: 4px;

padding-top: 2px;

border-radius: 10px;

border-color: #CACACA;

border-style: solid;

}

#dataCell {

background-color: white;

border-radius: 10px;

padding-top: 10px;

padding-left: 10px;

padding-right: 10px;

text-align: left;

margin-top: 3px;

}

#dataCell :hover {

cursor: pointer;

}

#dataCell img {

display: inline-block;

width: 60px;

height: 60px;

margin-bottom: 7px;

}

#dataCell #info {

display: inline-block;

margin-left: 10px;

}

#dataCell #info p {

color: #CACACA;

font-size: small;

overflow: hidden;

white-space: nowrap;

text-overflow: ellipsis;

max-width: 300px;

}

#dataCell h2 {

margin-top: 0px;

margin-bottom: -10px;

font-size: x-large;

overflow: hidden;

white-space: nowrap;

text-overflow: ellipsis;

max-width: 300px;

}

/\* Right-Side \*/

.dataDetail {

/\* display: inline-block; \*/

width: 450px;

height: 89vh;

border-radius: 10px;

border-style: solid;

border-color: #E6E6E6;

margin-left: 5px;

}

.dataDetail #date {

color: #CACACA;

}

.dataDetail #detailInfo {

margin-left: 80px;

margin-right: 80px;

}

#beerImage {

height: 300px;

background-repeat: no-repeat;

background-size: contain;

}

### Script.js

//MARK: Element

// Left-side

const dataListEle = document.getElementById("dataList")

// Right-side

const nameDetailProduct = document.getElementById("selectedName")

const dateDetailProduct = document.getElementById("date")

const descriptionDetailProduct = document.getElementById("detailInfo")

const beerImageProdect = document.getElementById("beerImage")

let allPage = 1;

const pageSize = 20;

let dataLoading = false;

const localDataArray = [];

function init() {

if (isHaveLocalData()) {

readLocalStorage()

renderAllData()

} else {

fetchData()

}

}

function fetchData() {

console.log("Fetch Data")

fetch(`https://api.punkapi.com/v2/beers?page=${allPage}&per\_page=${pageSize}`).then(async (res) => {

let data = await res.json()

console.log(data);

data.forEach(element => {

localDataArray.push(element)

});

renderAllData()

writeLocalStorage()

})

}

function renderAllData() {

localDataArray.forEach(element => {

dataListEle.innerHTML += `

<div id="dataCell" onclick="clickHandling(${element.id})">

<img src="http://cleanswifter.com/wp-content/uploads/2016/07/beer.png" alt="" srcset="">

<div id="info">

<h2>${element.name}</h2>

<p>${element.description}</p>

</div>

</div>

`

});

dataLoading = true

}

function writeLocalStorage() {

let localData = JSON.stringify(localDataArray)

localStorage.setItem("beerLocalData", localData)

}

function readLocalStorage() {

console.log("Read Local Data")

let local = localStorage.getItem("beerLocalData")

let data = JSON.parse(local)

console.log(data)

data.forEach(element => {

localDataArray.push(element)

})

}

function isHaveLocalData() {

let local = localStorage.getItem("beerLocalData")

let data = JSON.parse(local)

if (data === null) return false

else return true

}

function clickHandling(selecrtedID) {

localDataArray.forEach(element => {

if (element.id === selecrtedID) {

renderDetailData(element)

}

});

}

function renderDetailData(data) {

console.log("Clicked Data: ", data)

beerImageProdect.src = data.image\_url

nameDetailProduct.innerHTML = data.name

dateDetailProduct.innerHTML = data.first\_brewed

descriptionDetailProduct.innerHTML = data.description

}

dataListEle.addEventListener("scroll", (event) => {

const pageNumberEle = document.getElementById("pageNumber")

const { scrollTop, clientHeight, scrollHeight } = dataListEle;

// console.log(scrollWrapper.scrollTop)

if (clientHeight + scrollTop >= scrollHeight \* 0.95) {

allPage += 1;

if (dataLoading) {

fetchData();

console.log("Loading new page")

}

dataLoading = false

}

})

init();

## Result

