Activity 7 : Backend

Group #	11

Participating group members

Student ID	Name
6431309721	ชนกันต์ วิริยสถาพรพงศ์
6431345221	ศุภกฤษ อยู่เย็น
6432154921	วรันธร จันทร์สว่าง

Synopsis

Here is a list of what you are expected to have already learned from the preparation.

- How to start a Firebase project
- How to create a Firestore in the project
- How to setup a Firestore in the project
- How to connect Firestore to a frontend application
- How to do CREATE, DELETE, and GET data to/from database
- All these in your last project!

In the activity, you will:

- Learn some Javascript concepts used in the starter project
- Try to implement it to the Frontend Web Application

Activity time table

Time	Activities
9:30 - 10:00	Activity briefing, Attendance checking (at 9:45)
10:00 - 12.20	Working in groups (Part A, Part B, Part C, Outstanding)
12:20 - 12:30	Activity debrief

Part A: Setup a Firestore Collection

- Setup your Firestore to store **Documents** of items to buy
- Add a new **Collection** in you Firestore

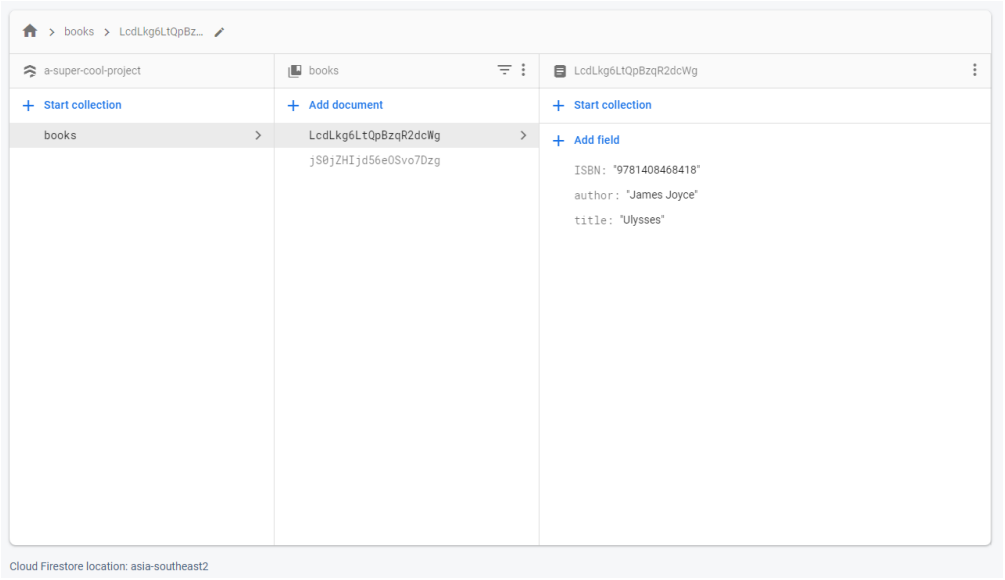
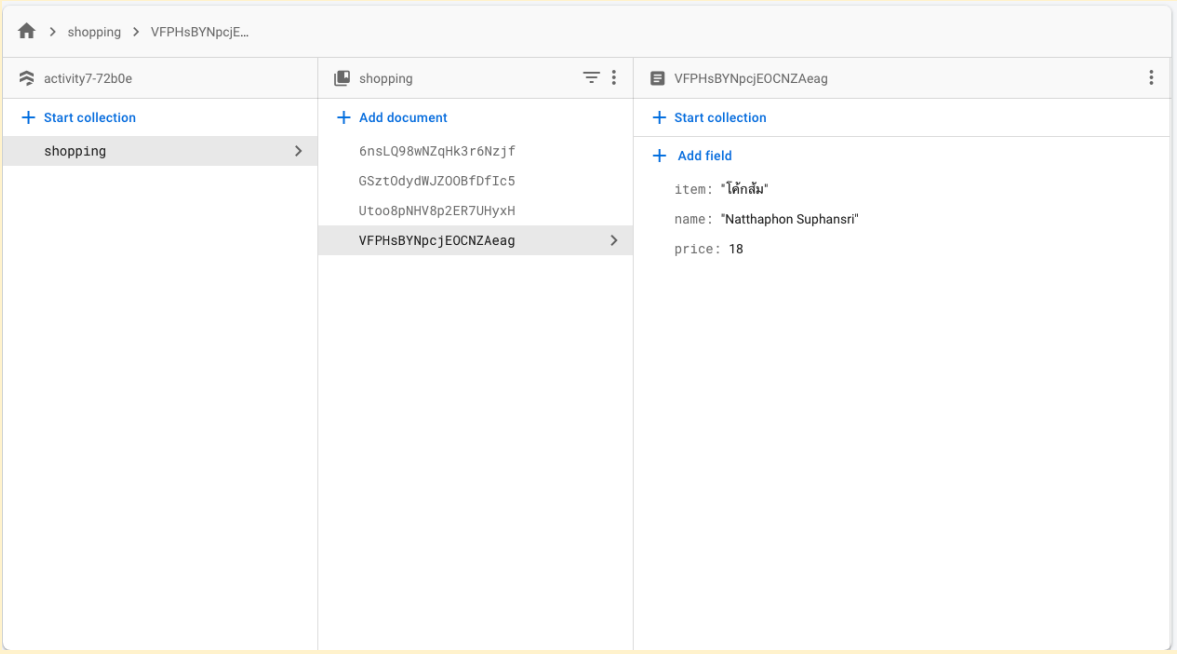


Fig. 1 Firestore page that you will add a new Collection of items to buy

What is the structure of a Document that represents an item to buy

{
item: string
name: string
price: number
}

Show a screenshot of a Firestore page with your new Collection


Part B: Connecting Firebase to the Web App

- Connect your Firebase project to the Web App

Show a screenshot of how you did that

```
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.6.8/firebase-app.js";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCBp906DcmpYJiEEp-8dS9dEhyQcFF3JJw",
  authDomain: "activity7-72b0e.firebaseapp.com",
  projectId: "activity7-72b0e",
  storageBucket: "activity7-72b0e.appspot.com",
  messagingSenderId: "100453663532",
  appId: "1:100453663532:web:b531e68012bf9f76672289"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Part C: Sync Firestore to the Web App

- (i) The items shown in the table should come from the items in Firestore
- (ii) Adding a new item in the table should also add that item to the Firestore
- (iii) Removing an item in the table should also remove that item in the Firestore
- See this sample of the end result

Useful JavaScript How-to

- Creating and triggering events
  - window.document.dispatchEvent
  - document.addEventListener

You can do (i), (ii), (iii) with your own methods, but this snippet code may be useful

```
const items_ref = collection(db, '{???????????}');

async function showItemsInTable() {
  const table_body = document.getElementById('main-table-body')
```

```
    table_body.innerHTML = ''

    const collection = '{???????????}'
    const items = '{???????????}'
    items.map((item) => {
        table_body.innerHTML += `
        <tr id="${item.docId}">
            <td>${'{???????????}'}</td>
            <td>${'{???????????}'}</td>
            <td>${'{???????????}'}</td>
            <td><button class="delete-row" onclick="deleteItem('${item.docId}')">ลบ
</button></td>
        </tr>
        `
    })
}

document.addEventListener("DOMContentLoaded", function(event) {
    console.log('showing items from database')
    showItemsInTable()
});

function redrawDOM() {
    window.document.dispatchEvent(new Event("DOMContentLoaded", {
        bubbles: true,
        cancelable: true
    }));
    document.getElementById("item-to-add").value = "";
    document.getElementById("name-to-add").value = "0";
    document.getElementById("price-to-add").value = "";
}

async function addItem() {

}

var addButton = document.querySelector("#add-newrow");
addButton.onclick = async () => {
    addItem().then(() => {
        redrawDOM()
    })
}

async function deleteItem(docId) {

}
```

Describe your code how to do (i)

```
const db = getFirestore();
const items_ref = collection(db, 'shopping');

async function showItemsInTable() {
  const table_body = document.getElementById('main-table-body')
  table_body.innerHTML = ''
  const collection = await getDocs(items_ref);
  const items = collection.docs.map((item) => ({
    docId: item.id,
    ...item.data(),
}));
console.log(items)
  items.map((item) => {
    table_body.innerHTML += `
    <tr id="${item.docId}">
        <td>${item.item}</td>
        <td>${item.name}</td>
        <td>${item.price}</td>
        <td><button class="delete-row" onclick="deleteItem('${item.docId}')">ลบ</button></td>
    </tr>
    `
  })
}
```

ดึงข้อมูลจาก FireStore ที่ collection ชื่อ shopping จากนั้นสร้างตารางโดยดึงมาจาก main-table-body แล้วเอาข้อมูลที่ดึงมาใส่ในตาราง โดยเลือกให้ id ตรงกัน

Describe your code how to do (ii)

```
async function addItem() {
  console.log('addItem')
  const item = document.getElementById("item-to-add").value;
  const name = document.getElementById("name-to-add").value;
  const price = document.getElementById("price-to-add").value;
  if(item == ""){
    return;
  }
  if(name == ""){
    return;
  }
  if(price == ""){
    return;
  }

  addDoc(items_ref, {
    item,
    name,
    price,
  });
}
```

ดึงข้อมูลมาสร้างเป็นตัวแปรโดยการใช้ getElementById().value จากนั้น ใช้ addDoc เพื่อเพิ่มข้อมูลเข้าใน FireStore

Describe your code how to do (iii)

```
async function deleteItem(docId) {
  const docRef = doc(db, `shopping/${docId}`);
  await deleteDoc(docRef);
  console.log("deleteItem")
  redrawDOM()
}
window.deleteItem = deleteItem;
```

ใช้ doc ในการดึงข้อมูลของ shopping ตาม docId เพื่อลบจากนั้น redrawDOM() ตารางใหม่

- ส่งงาน Part A, B, C ใน Activity 9: Submission ภายใน 12:20
    - ส่งไฟล์ Worksheet นี้ในรูปแบบ PDF โดยแนบไปยัง Attachment Slot ชื่อ Worksheet PDF
    - Zip โฟลเดอร์และส่งไฟล์ Zip โดยแนบไปยัง Attachment Slot ชื่อ Zipped Folder for Part A to C

Outstanding Factor

- ทำส่วนนี้เพื่อรับการพิจารณา Outstanding Factor สำหรับกิจกรรมนี้ โดยเลือกทำอย่างใดอย่างหนึ่งระหว่าง

Host your frontend at the Firebase Hosting, you will need to download NodeJS to be able to use *npm*
A Link to Your Website

หรือ

Add Update functionality to the Frontend
Describe your code how to do it

- ส่งงานเพื่อ Outstanding Factor ใน Activity 9: Submission ภายใน 12:20
    - Zip โฟลเดอร์และส่งไฟล์ Zip โดยแนบไปยัง Attachment Slot ชื่อ Zipped Folder for Outstanding Factor