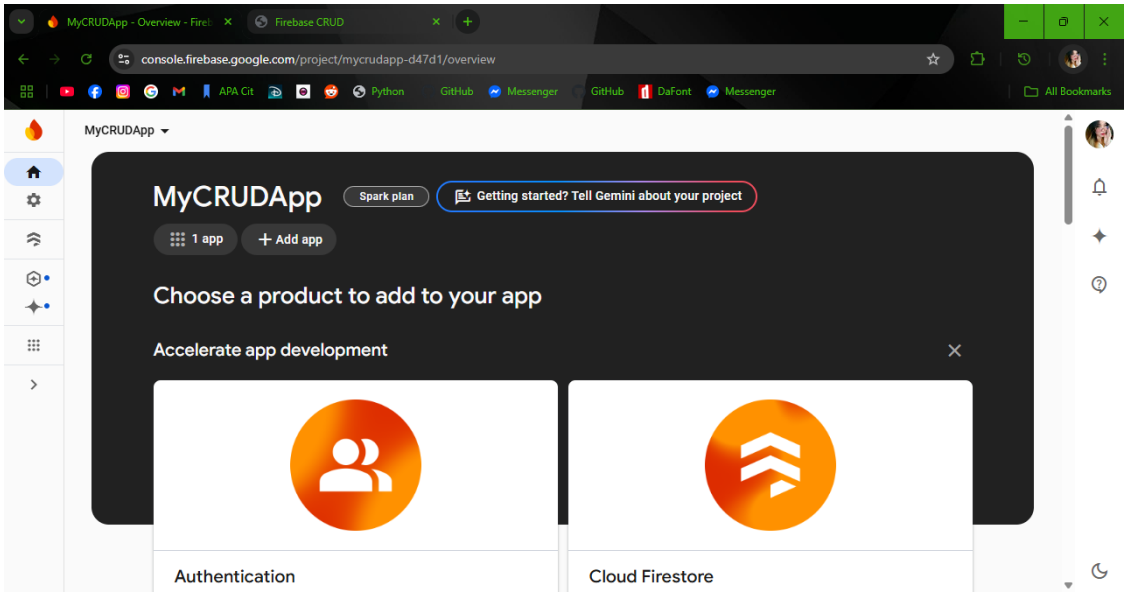
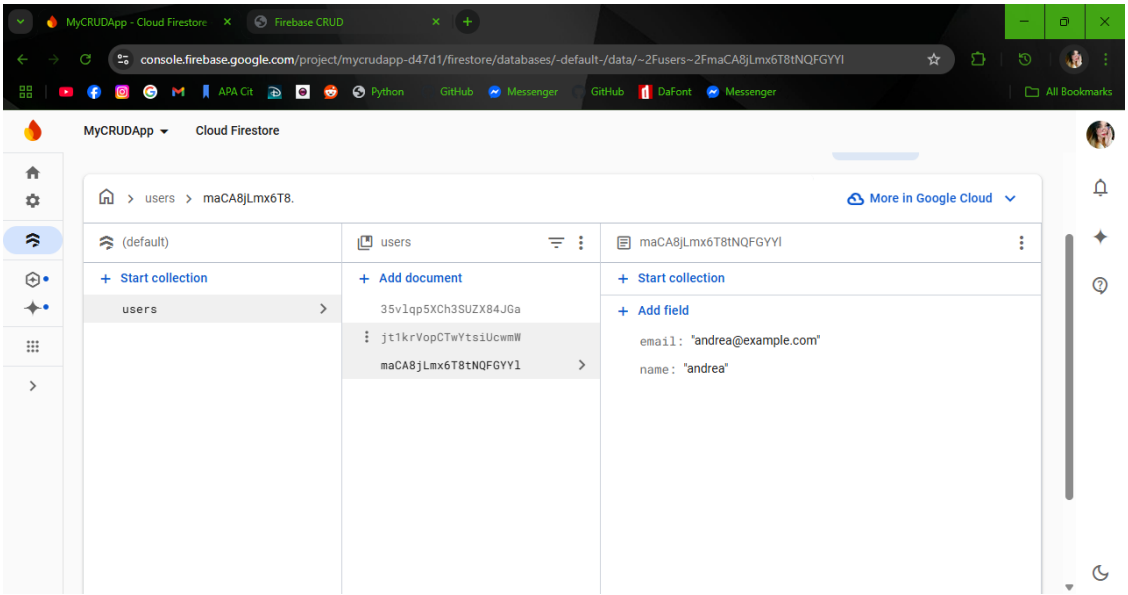
1. **Firebase Project Screenshot** (Project name (MyCRUDApp), Firebase database created, sample data under user collection)





2. **Complete html code**

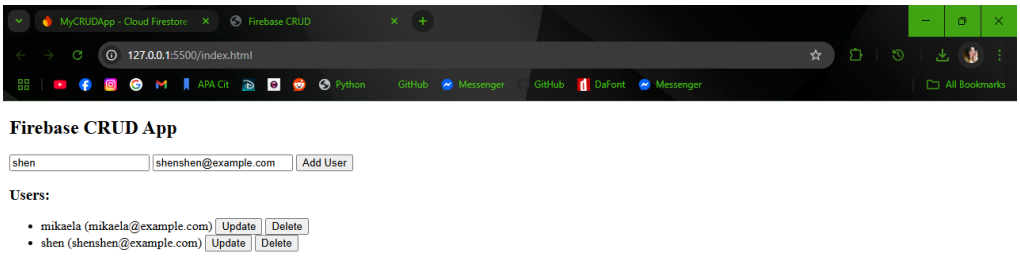
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Firebase CRUD</title>
</head>
<body>
<h2>Firebase CRUD App</h2>
<input type="text" id="name" placeholder="Enter name" />
<input type="email" id="email" placeholder="Enter email" />
<button onclick="addUser()">Add User</button>
<h3>Users:</h3>
<ul id="userList"></ul>
<script type="module">
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.6.1/firebase-app.js";
import { getFirestore, collection, addDoc, getDocs, updateDoc, deleteDoc, doc }
from "https://www.gstatic.com/firebasejs/9.6.1/firebase-firestore.js";
const firebaseConfig = {
apiKey: "AIzaSyAo2RL-PhB66w2pdgdn5Wiy-UOr5Gh9Ta4",
authDomain: "mycrudapp-d47d1.firebaseapp.com",
projectId: "mycrudapp-d47d1",
storageBucket: "mycrudapp-d47d1.firebasestorage.app",
messagingSenderId: "911889320489",
appId: "1:911889320489:web:5a5c125d84218525676a38"
};
//
const app = initializeApp(firebaseConfig);
const db = getFirestore(app);
// Add
async function addUser() {
const name = document.getElementById("name").value;
const email = document.getElementById("email").value;
if (!name || !email) {
alert("Please enter both name and email.");
return;
}
try {
await addDoc(collection(db, "users"), { name, email });
alert("User added successfully!");
fetchUsers();
} catch (error) {
console.error("Error adding user:", error);
```

```
}
}
// Fetch Users
async function fetchUsers() {
const querySnapshot = await getDocs(collection(db, "users"));
const userList = document.getElementById("userList");
userList.innerHTML = ""; // Clear previous list
querySnapshot.forEach((doc) => {
const li = document.createElement("li");
li.innerHTML = `${doc.data().name} (${doc.data().email})
<button onclick="updateUser('${doc.id}')">Update</button>
<button onclick="deleteUser('${doc.id}')">Delete</button>`;
userList.appendChild(li);
});
}
// Update
async function updateUser(userId) {
const newName = prompt("Enter new name:");
if (newName) {
try {
await updateDoc(doc(db, "users", userId), { name: newName });
alert("User updated successfully!");
fetchUsers();
} catch (error) {
console.error("Error updating user:", error);
}
}
}
// Delete
async function deleteUser(userId) {
try {
await deleteDoc(doc(db, "users", userId));
alert("User deleted successfully.");
fetchUsers();
} catch (error) {
console.error("Error deleting user:", error);
}
}
//Fetch users
window.fetchUsers = fetchUsers;
window.addUser = addUser;
window.updateUser = updateUser;
window.deleteUser = deleteUser;
fetchUsers();
```

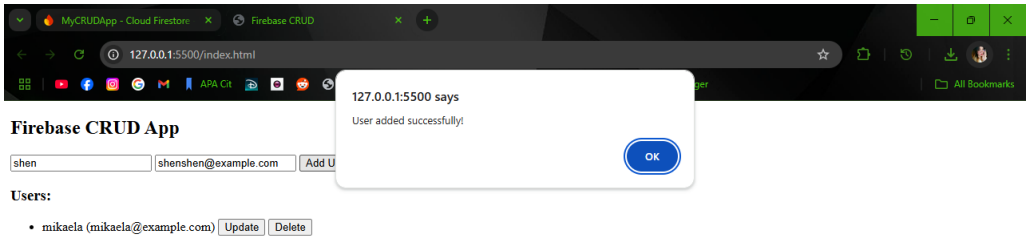
```
</script>
</body>
</html>
```

5. Screenshot of Browser Output

List of Users, Update & Delete buttons:



After Performing Update operation



After Performing Delete Operation