This Flask app is a simple user management system that allows you to:

View all users

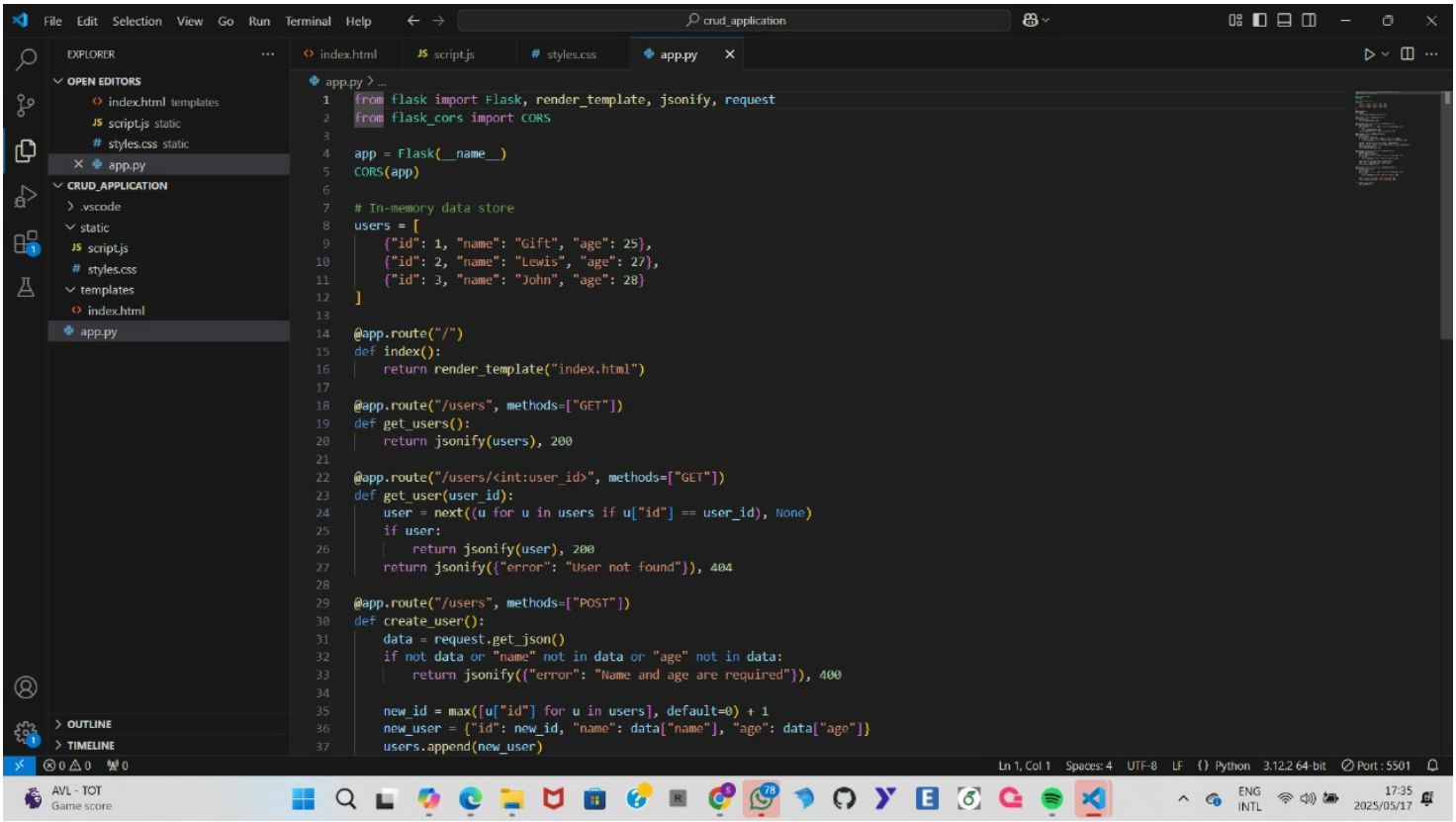
View a single user by ID

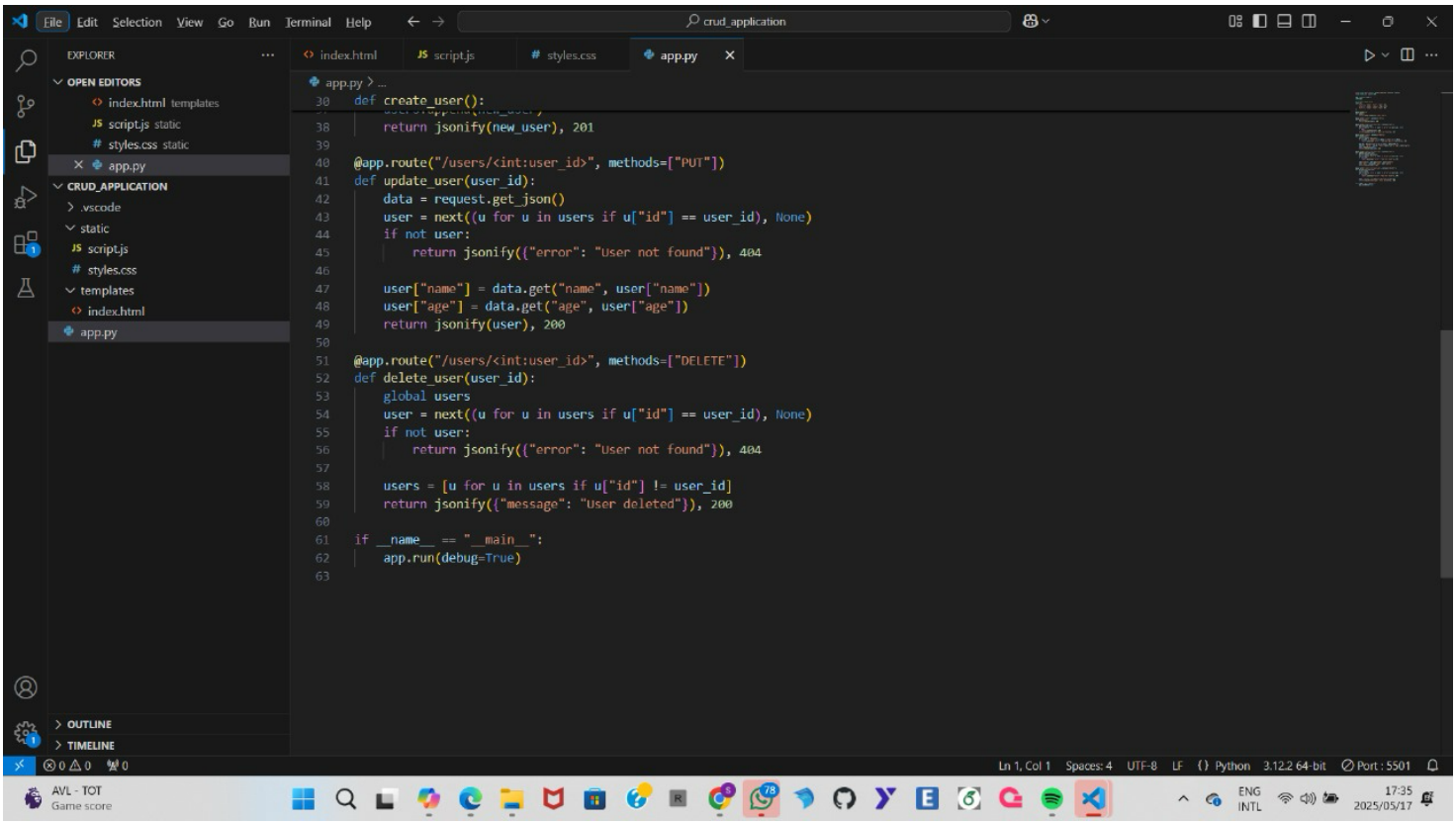
Add a new user

Update a user's details

Delete a user

It uses an in-memory list to store users (no database), and handles all the logic through different URL routes (/users, /users/<id>). The app responds with JSON data and supports frontend interaction using CORS. The main page is rendered using an index.html file. It’s great for learning how basic web APIs and CRUD operations work using Python and Flask.





Summary of What the HTML Code Does:
This HTML file is the user interface for a Flask-based CRUD application. It allows users to:

Add a new user with a name and age

Edit existing user information

Delete users from the list

Search for users by name

Display all users in a table format

The frontend connects to the backend (app.py) using JavaScript (from script.js) and dynamically updates the content without refreshing the page.

✨ Few Important Code Sections:
Search Field and Button
Allows the user to search for names:

<input type="text" id="searchInput" placeholder="Search by name...">
<button id="searchBtn">Search</button>
User Form (Add/Edit)

A form that captures name and age:

```
<form id="userForm">
   <input type="hidden" id="userId" value="">
   <input type="text" id="name" required>
   <input type="number" id="age" min="1" required>
</form>
```

Users Table

Displays all users and their actions:

```
<table>
   <thead>
      <tr>
         <th>ID</th>
         <th>Name</th>
         <th>Age</th>
         <th>Actions</th>
      </tr>
   </thead>
   <tbody id="userTable"></tbody>
</table>
```

Alert Message Box

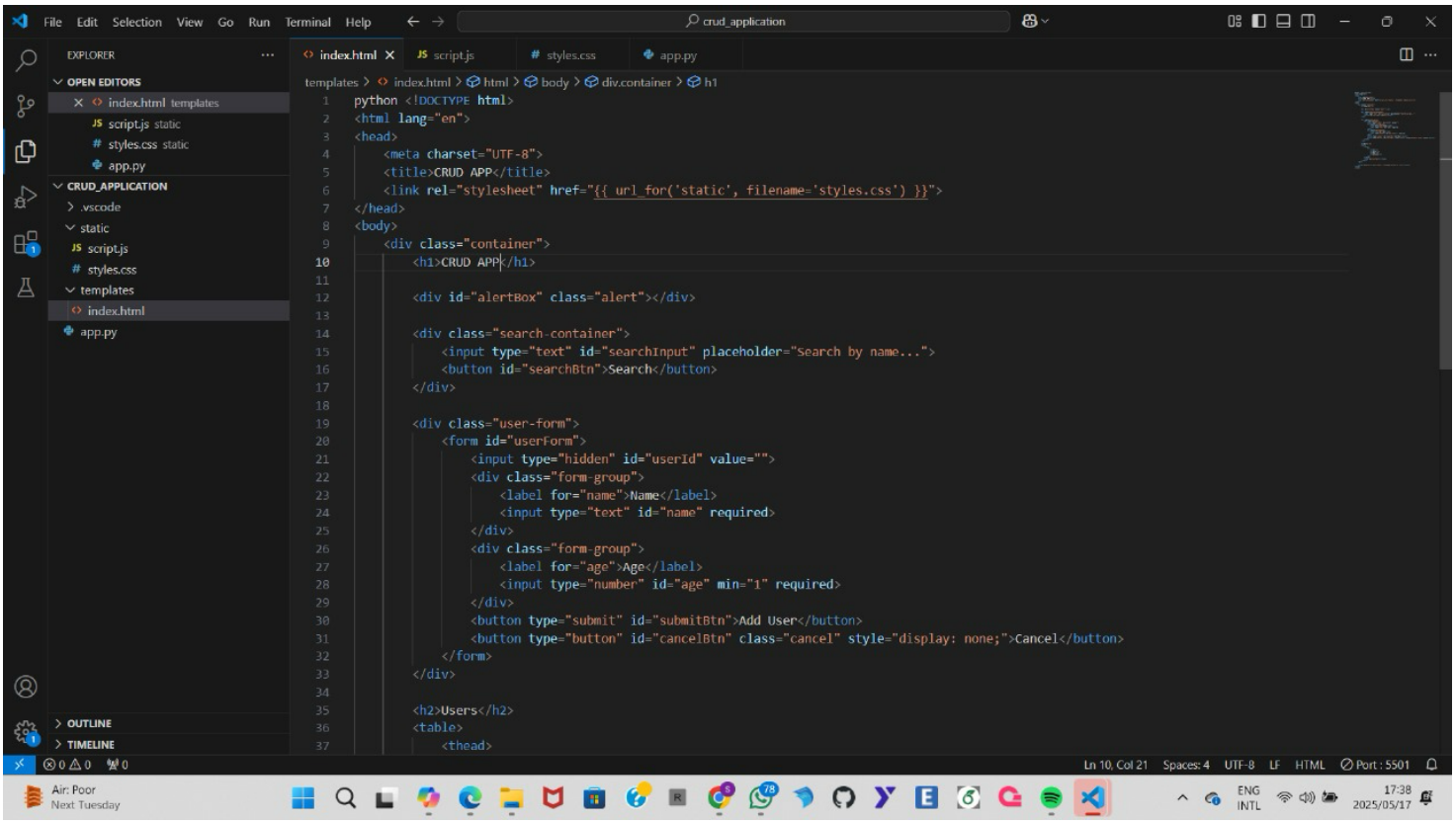
Used to show success or error messages:

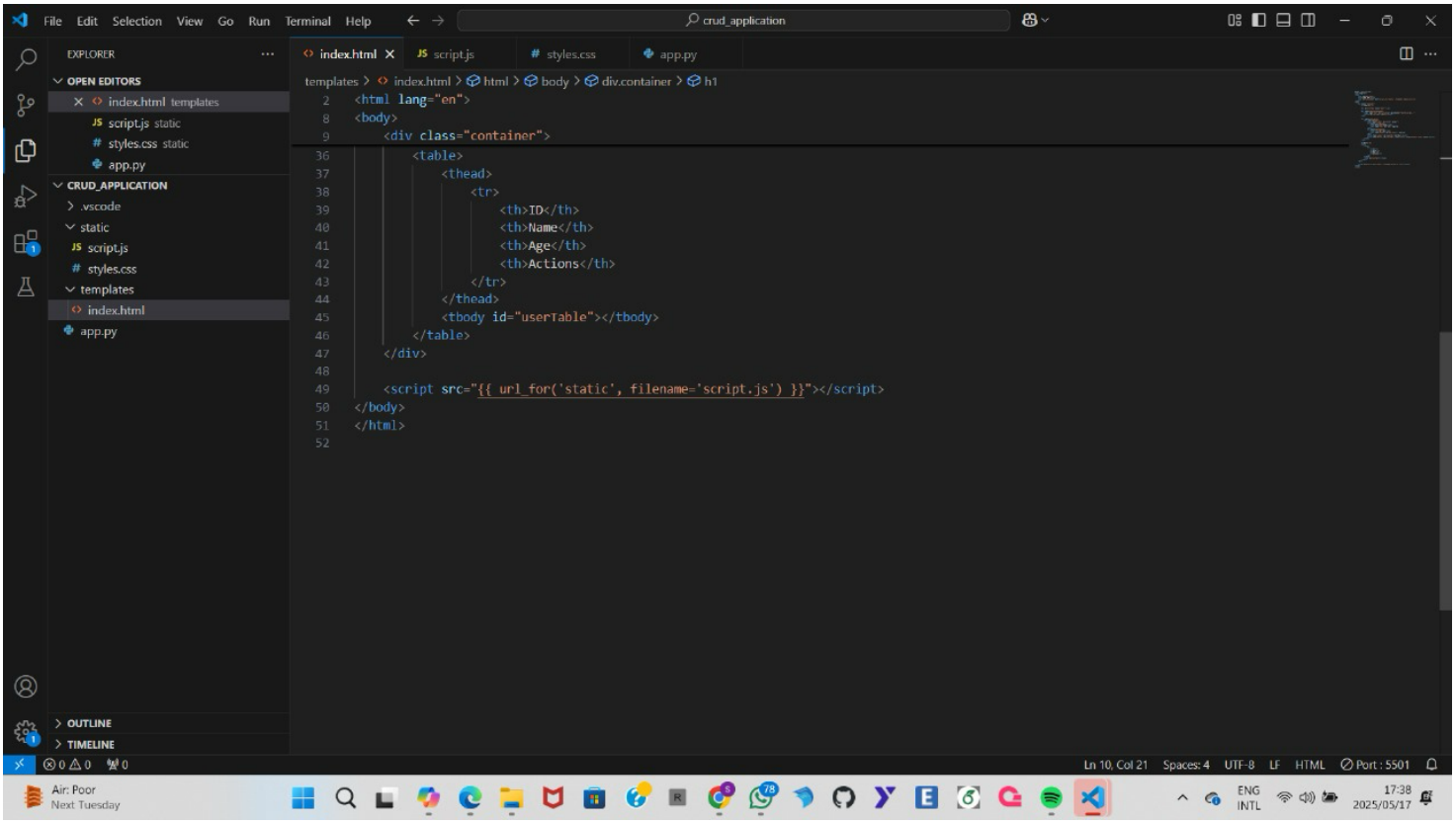
```
<div id="alertBox" class="alert"></div>
```

CSS & JS Linking

Links to external styles and scripts:

```
<link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
<script src="{{ url_for('static', filename='script.js') }}"></script>
```





This CSS file styles the CRUD App interface to make it clean, readable, and visually appealing. It:

Sets a light background and modern font across the app.

Centers the content using a styled .container.

Styles inputs and buttons to be user-friendly.

Adds hover effects for buttons to improve interactivity.

Formats the table for displaying user data neatly.

Defines alert message styles for success and error feedback.

Few Important Code Sections (Core Styling):
Centering and Styling the App Container:

```
.container {
    max-width: 800px;
    margin: auto;
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
}
```

Input and Button Styling:

```
input, button {
    padding: 10px;
    margin: 5px;
    font-size: 14px;
    border-radius: 4px;
    border: 1px solid #ccc;
}
```

Hover Effect for Buttons:

```
button:hover {
    transform: scale(1.05);
    background-color: #45a049;
}
```

Error and Success Alert Boxes:

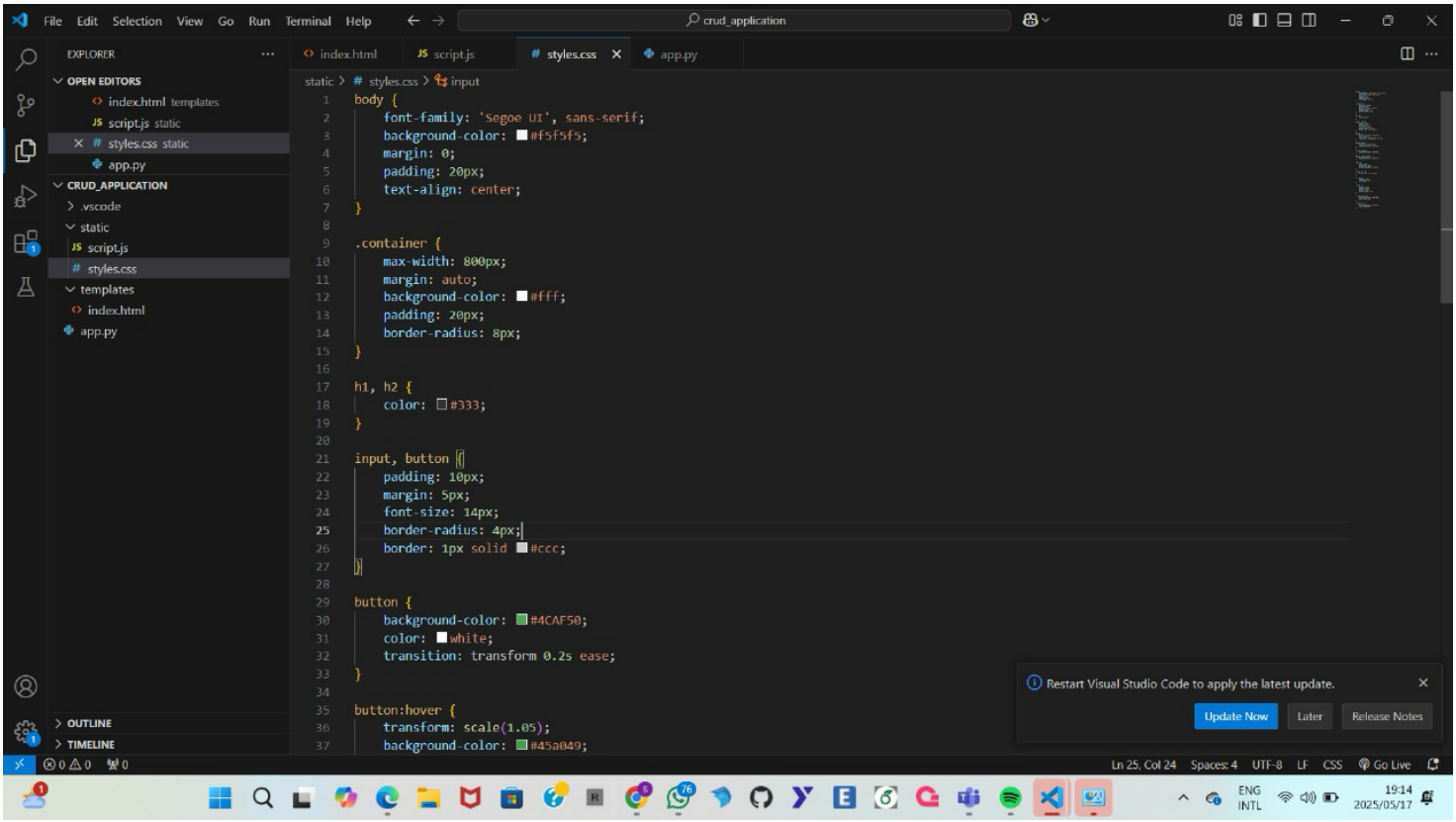
```
.alert-success {
    background-color: #dff0d8;
    color: #3c763d;
}
```

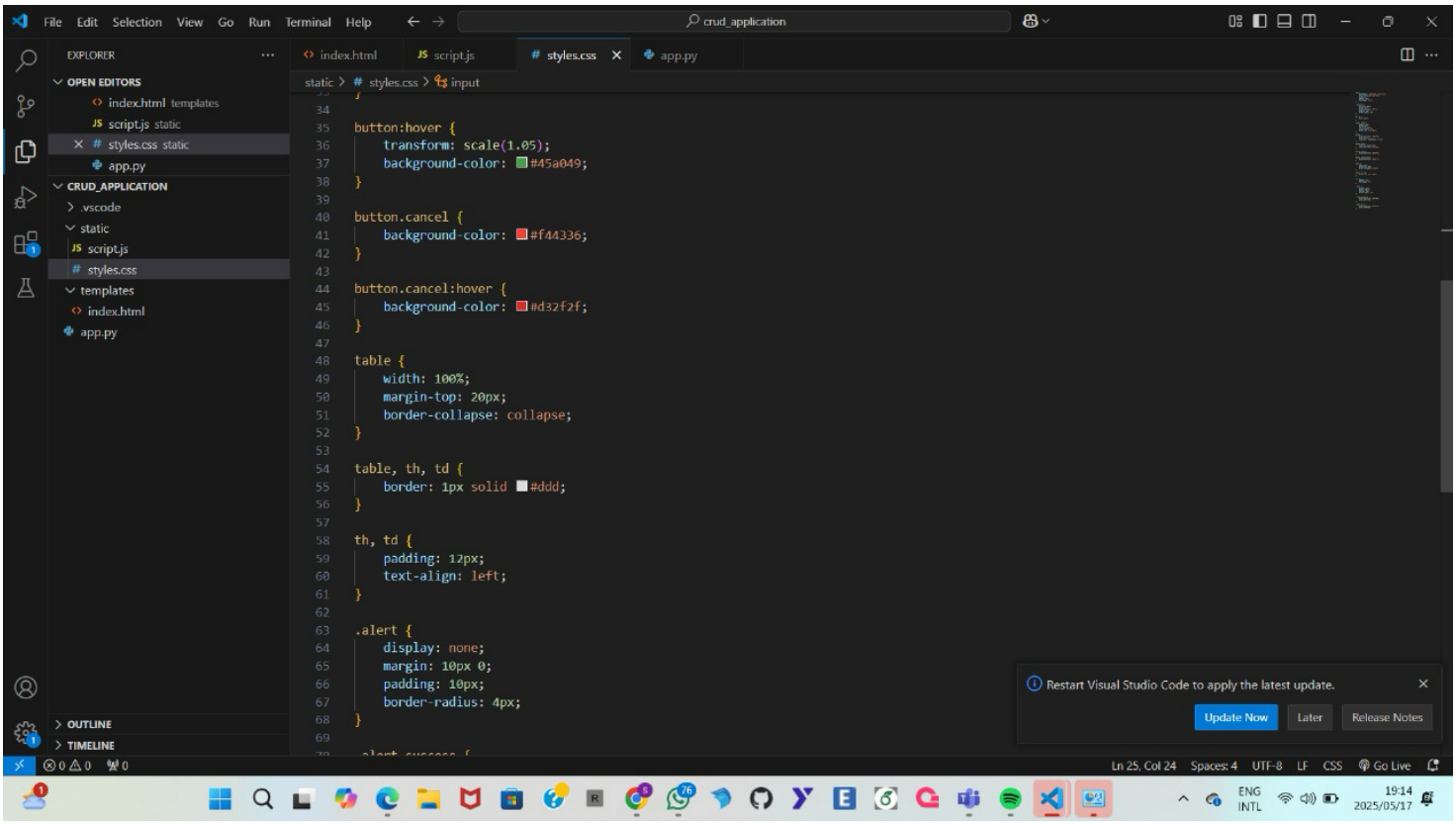
```
.alert-error {
    background-color: #f2dede;
    color: #a94442;
}
```

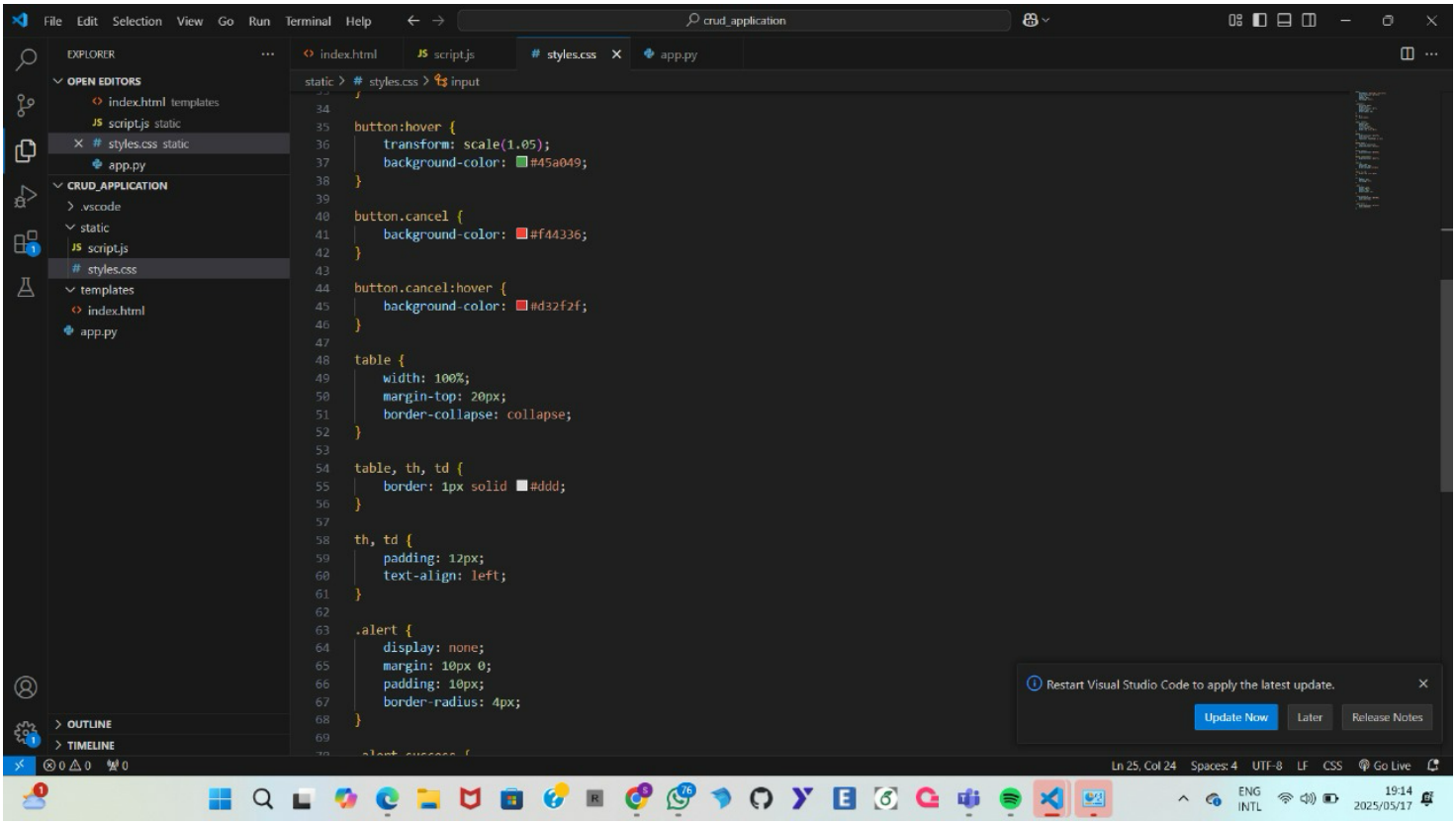
Table Styling for User List:

```
table {
    width: 100%;
    margin-top: 20px;
    border-collapse: collapse;
}

table, th, td {
    border: 1px solid #ddd;
}
```







This JavaScript file powers the interactivity of the CRUD App by:

Fetching users from a backend API and displaying them in a table.

Adding new users via a form.

Editing existing users by filling the form with their data.

Updating user information using a PUT request.

Deleting users from the list using a DELETE request.

Searching users by name and filtering the displayed list.

Showing alert messages for actions like add, update, or delete.

It dynamically updates the user list without reloading the page, making the app responsive and smooth.

Few Core/Important Code Sections
Initial Fetch of All Users:

```
function fetchUsers() {
    fetch(API_URL + '/users')
        .then(res => res.json())
        .then(data => {
            allUsers = data;
            renderUsers(data);
        });
}
```

Render Users into HTML Table:

```
function renderUsers(users) {
    userTable.innerHTML = '';
    users.forEach(user => {
        const row = document.createElement('tr');
        row.innerHTML = `
            <td>${user.id}</td>
            <td>${user.name}</td>
            <td>${user.age}</td>
            <td>
                <button onclick="editUser(${user.id})">Edit</button>
                <button onclick="deleteUser(${user.id})" class="cancel">Delete</button>
            </td>
        `;
        userTable.appendChild(row);
```

```
    });
}
```

Handle Add/Update User Form Submission:

```
userForm.addEventListener('submit', function (e) {
    e.preventDefault();
    const user = { name: nameField.value, age: parseInt(ageField.value) };
    const id = userIdField.value;

    const method = id ? 'PUT' : 'POST';
    const url = id ? `${API_URL}/users/${id}` : `${API_URL}/users`;

    fetch(url, {
        method: method,
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(user)
    })
        .then(res => res.json())
        .then(() => {
            fetchUsers();
            userForm.reset();
            userIdField.value = '';
            submitBtn.textContent = 'Add User';
            cancelBtn.style.display = 'none';
            showAlert(`User ${id ? 'updated' : 'added'} successfully!`, 'success');
        });
});
```

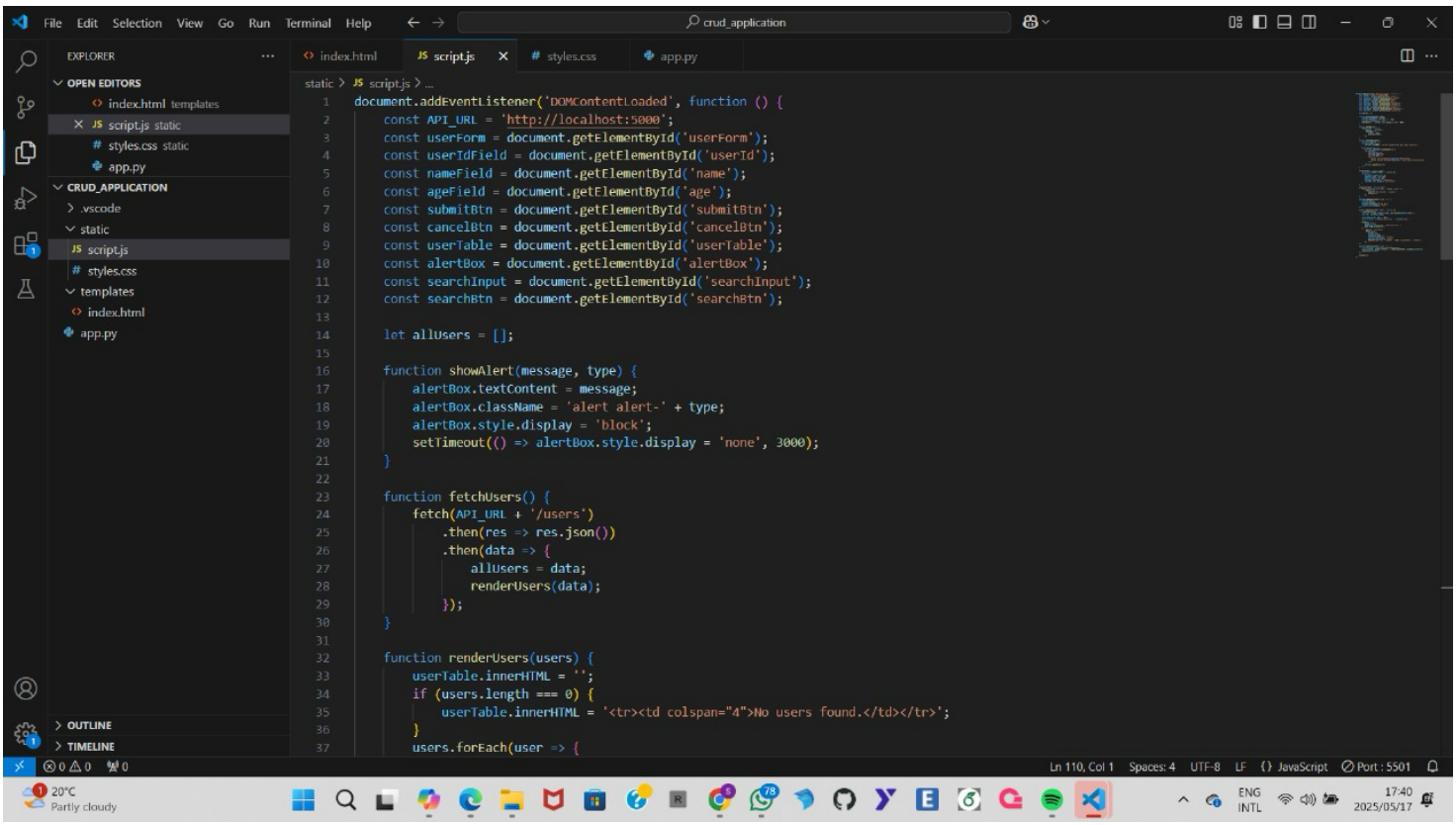
Search by Name:

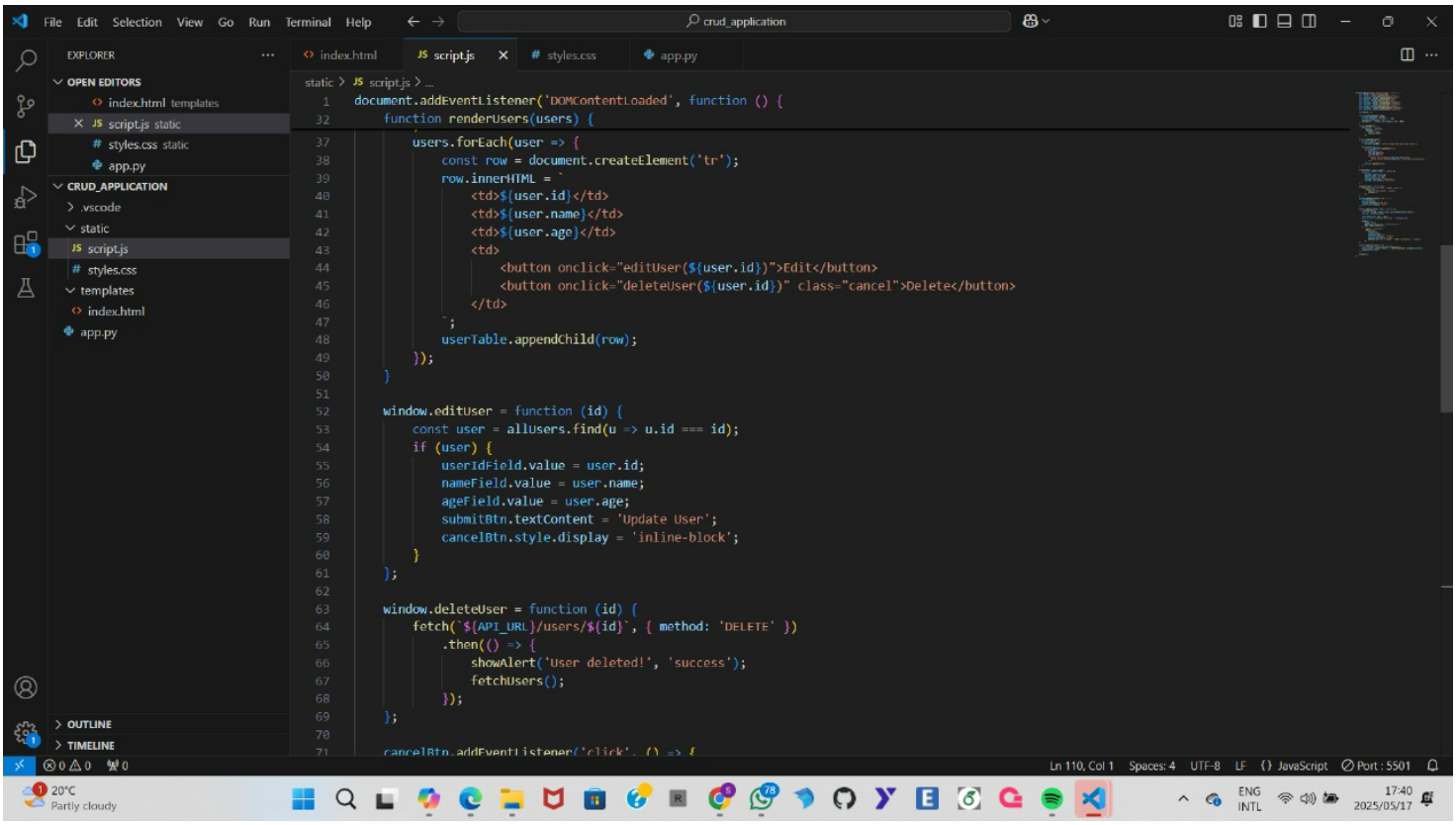
```
searchBtn.addEventListener('click', () => {
    const searchTerm = searchInput.value.toLowerCase();
    const filtered = allUsers.filter(u => u.name.toLowerCase().includes(searchTerm));
    renderUsers(filtered);
});
```

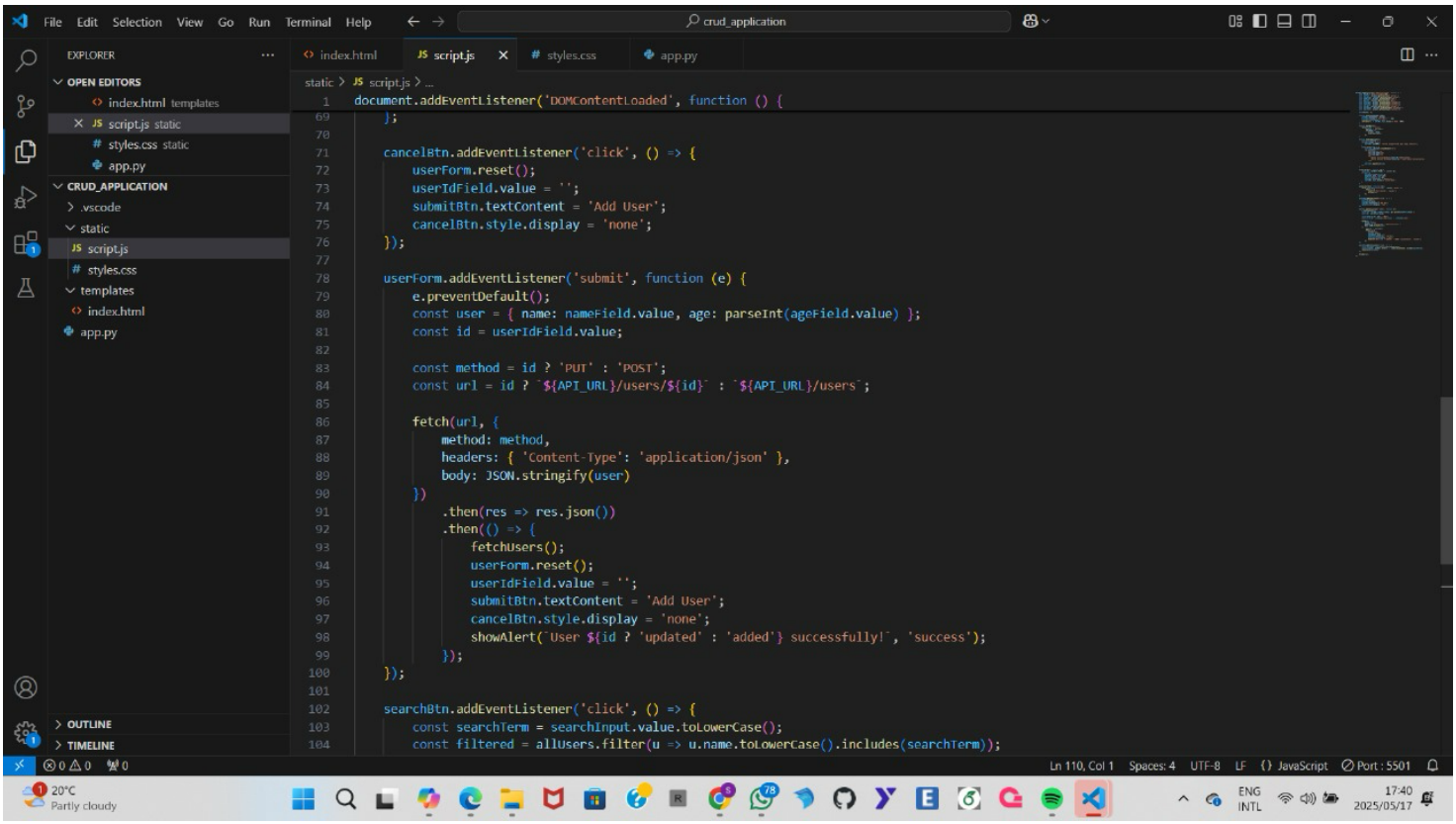
Alert Notification Box:

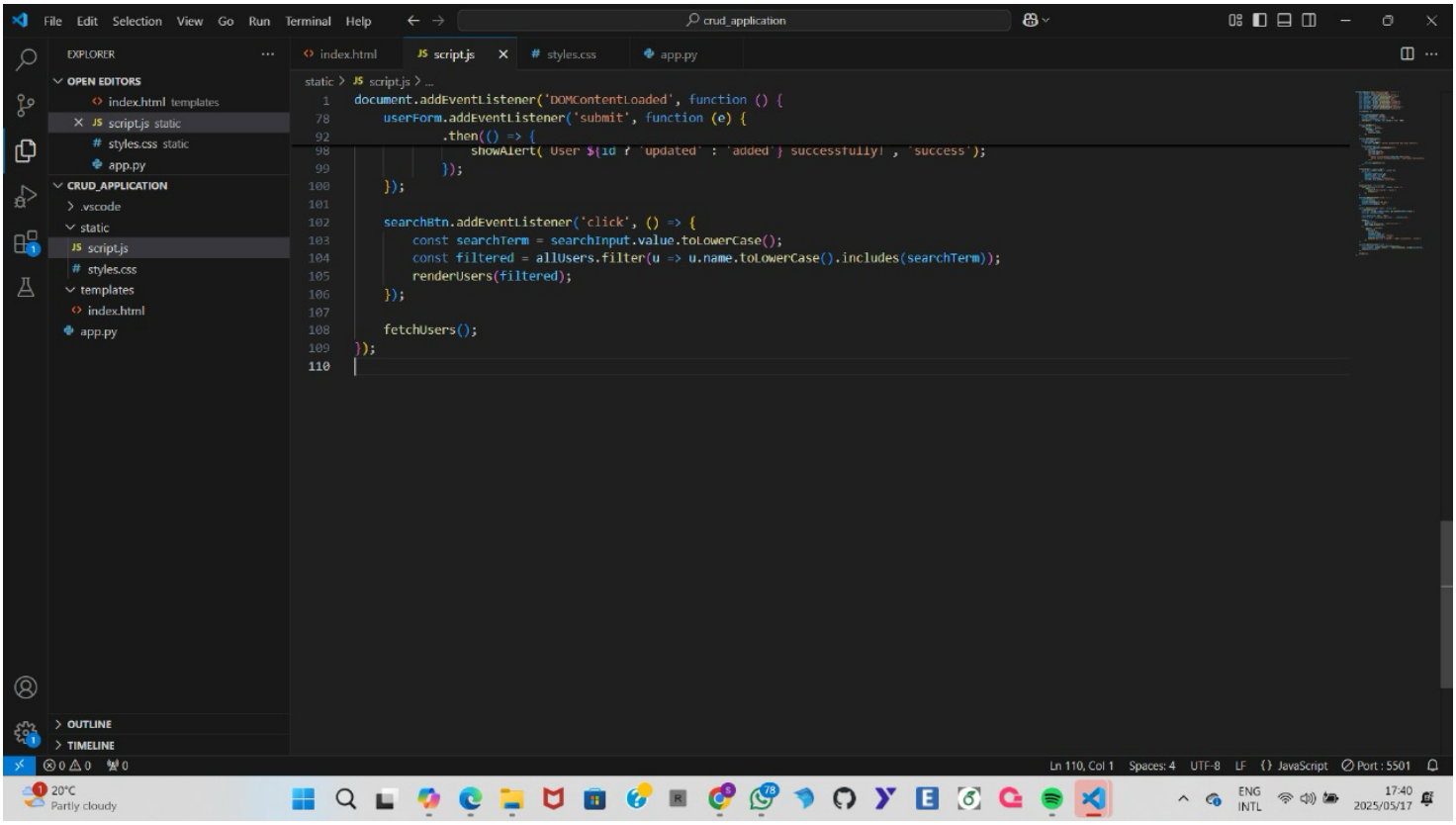
```
function showAlert(message, type) {
    alertBox.textContent = message;
    alertBox.className = 'alert alert-' + type;
    alertBox.style.display = 'block';
    setTimeout(() => alertBox.style.display = 'none', 3000);
```

}


File Edit Selection View Go Run Terminal Help
crud_application
EXPLORER
OPEN EDITORS
index.html templates
script.js static
styles.css static
app.py
CRUD_APPLICATION
.vscode
static
script.js
styles.css
templates
index.html
app.py
OUTLINE
TIMELINE
index.html
script.js
styles.css
app.py
static > script.js > ...
document.addEventListener('DOMContentLoaded', function () {
const API_URL = 'http://localhost:5000';
const userForm = document.getElementById('userForm');
const userIdField = document.getElementById('userId');
const nameField = document.getElementById('name');
const ageField = document.getElementById('age');
const submitBtn = document.getElementById('submitBtn');
const cancelBtn = document.getElementById('cancelBtn');
const userTable = document.getElementById('userTable');
const alertBox = document.getElementById('alertBox');
const searchInput = document.getElementById('searchInput');
const searchBtn = document.getElementById('searchBtn');
let allUsers = [];
function showAlert(message, type) {
alertBox.textContent = message;
alertBox.className = 'alert alert-' + type;
alertBox.style.display = 'block';
setTimeout(() => alertBox.style.display = 'none', 3000);
}
function fetchUsers() {
fetch(API_URL + '/users')
.then(res => res.json())
.then(data => {
allUsers = data;
renderUsers(data);
});
}
function renderUsers(users) {
userTable.innerHTML = '';
if (users.length === 0) {
userTable.innerHTML = '<tr><td colspan="4">No users found.</td></tr>';
}
users.forEach(user => {
Ln 110, Col 1 Spaces: 4 UTF-8 LF {} JavaScript Port : 5501
20°C
Partly cloudy
ENG
INTL
17:40
2025/05/17


File Edit Selection View Go Run Terminal Help
crud_application
EXPLORER
OPEN EDITORS
index.html templates
script.js static
styles.css static
app.py
CRUD_APPLICATION
.vscode
static
script.js
styles.css
templates
index.html
app.py
OUTLINE
TIMELINE
index.html
script.js
styles.css
app.py
static > script.js > ...
document.addEventListener('DOMContentLoaded', function () {
function renderUsers(users) {
users.forEach(user => {
const row = document.createElement('tr');
row.innerHTML = `
<td>${user.id}</td>
<td>${user.name}</td>
<td>${user.age}</td>
<td>
<button onclick="editUser(${user.id})">Edit</button>
<button onclick="deleteUser(${user.id})" class="cancel">Delete</button>
</td>
`;
userTable.appendChild(row);
});
}
window.editUser = function (id) {
const user = allUsers.find(u => u.id === id);
if (user) {
userIdField.value = user.id;
nameField.value = user.name;
ageField.value = user.age;
submitBtn.textContent = 'Update User';
cancelBtn.style.display = 'inline-block';
}
};
window.deleteUser = function (id) {
fetch(`${API_URL}/users/${id}`, { method: 'DELETE' })
.then(() => {
showAlert('User deleted!', 'success');
fetchUsers();
});
};
Ln 110, Col 1 Spaces: 4 UTF-8 LF {} JavaScript Port : 5501
20°C
Partly cloudy
ENG
INTL
17:40
2025/05/17


File Edit Selection View Go Run Terminal Help
crud_application
EXPLORER
OPEN EDITORS
index.html templates
script.js static
styles.css static
app.py
CRUD_APPLICATION
.vscode
static
script.js
styles.css
templates
index.html
app.py
OUTLINE
TIMELINE
index.html
script.js
styles.css
app.py
static > script.js > ...
document.addEventListener('DOMContentLoaded', function () {
};
cancelBtn.addEventListener('click', () => {
userForm.reset();
userIdField.value = '';
submitBtn.textContent = 'Add User';
cancelBtn.style.display = 'none';
});
userForm.addEventListener('submit', function (e) {
e.preventDefault();
const user = { name: nameField.value, age: parseInt(ageField.value) };
const id = userIdField.value;
const method = id ? 'PUT' : 'POST';
const url = id ? `${API_URL}/users/${id}` : `${API_URL}/users`;
fetch(url, {
method: method,
headers: { 'Content-Type': 'application/json' },
body: JSON.stringify(user)
})
.then(res => res.json())
.then(() => {
fetchUsers();
userForm.reset();
userIdField.value = '';
submitBtn.textContent = 'Add User';
cancelBtn.style.display = 'none';
showAlert(`User ${id ? 'updated' : 'added'} successfully!`, 'success');
});
});
searchBtn.addEventListener('click', () => {
const searchTerm = searchInput.value.toLowerCase();
const filtered = allUsers.filter(u => u.name.toLowerCase().includes(searchTerm));
Ln 110, Col 1 Spaces: 4 UTF-8 LF {} JavaScript Port : 5501
20°C
Partly cloudy
ENG
INTL
17:40
2025/05/17


File Edit Selection View Go Run Terminal Help
crud_application
EXPLORER
OPEN EDITORS
index.html templates
script.js static
styles.css static
app.py
CRUD_APPLICATION
.vscode
static
script.js
styles.css
templates
index.html
app.py
OUTLINE
TIMELINE
index.html
script.js
styles.css
app.py
static > script.js > ...
document.addEventListener('DOMContentLoaded', function () {
userForm.addEventListener('submit', function (e) {
.then(() => {
showAlert(`User ${id ? 'updated' : 'added'} successfully!`, 'success');
});
});
searchBtn.addEventListener('click', () => {
const searchTerm = searchInput.value.toLowerCase();
const filtered = allUsers.filter(u => u.name.toLowerCase().includes(searchTerm));
renderUsers(filtered);
});
fetchUsers();
});
Ln 110, Col 1 Spaces: 4 UTF-8 LF {} JavaScript Port : 5501
20°C
Partly cloudy
ENG
INTL
17:40
2025/05/17