

Codice Front-End – Progetto Revolut Web App

index.html

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <title>Dashboard Revolut-style</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Dashboard multivaluta – Prototipo Revolut</h1>
  </header>

  <main>
    <section class="panel">
      <h2>1. Carica conti utente</h2>
      <label for="userId">ID utente:</label>
      <input type="number" id="userId" value="1">
      <button id="loadAccountsBtn">Carica conti</button>

      <table id="accountsTable">
        <thead>
          <tr>
            <th>ID conto</th>
            <th>Valuta</th>
            <th>Saldo</th>
          </tr>
        </thead>
        <tbody></tbody>
      </table>
    </section>

    <section class="panel">
      <h2>2. Transazioni conto</h2>
      <label for="accountSelect">Seleziona conto:</label>
      <select id="accountSelect">
        <option value="">-- nessun conto --</option>
```

```

</select>
<button id="loadTransactionsBtn">Carica transazioni</button>

<table id="transactionsTable">
  <thead>
    <tr>
      <th>ID</th>
      <th>Tipo</th>
      <th>Importo</th>
      <th>Valuta</th>
      <th>Descrizione</th>
      <th>Data</th>
    </tr>
  </thead>
  <tbody></tbody>
</table>
</section>

<section class="panel">
  <h2>3. Trasferimento interno</h2>
  <form id="transferForm">
    <div class="form-row">
      <label for="sourceAccountId">Conto sorgente (ID):</label>
      <input type="number" id="sourceAccountId" required>
    </div>
    <div class="form-row">
      <label for="targetAccountId">Conto destinazione (ID):</label>
      <input type="number" id="targetAccountId" required>
    </div>
    <div class="form-row">
      <label for="amount">Importo:</label>
      <input type="number" step="0.01" id="amount" required>
    </div>
    <button type="submit">Esegui trasferimento</button>
  </form>

  <div id="transferResult" class="feedback"></div>
</section>
</main>

```

```
<footer>
  <p>Progetto didattico – Web app full-stack API-based ispirata a Revolut</p>
</footer>

<script>
  const API_BASE = "http://127.0.0.1:8000";
</script>
<script src="app.js"></script>
</body>
</html>
```

styles.css

```
body {
  font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI", sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f5f7fb;
  color: #222;
}

header {
  background-color: #1f41bb;
  color: #fff;
  padding: 1.5rem;
  text-align: center;
}

main {
  max-width: 1000px;
  margin: 2rem auto;
  padding: 0 1rem;
}

.panel {
  background-color: #fff;
  border-radius: 8px;
  padding: 1.5rem;
  margin-bottom: 1.5rem;
  box-shadow: 0 2px 6px rgba(0,0,0,0.05);
}
```

app.js

```
async function loadAccounts() {
  const userId = document.getElementById("userId").value;
  if (!userId) return;

  const res = await fetch(`${API_BASE}/api/users/${userId}/accounts`);
  const tableBody = document.querySelector("#accountsTable tbody");
  const accountSelect = document.getElementById("accountSelect");

  tableBody.innerHTML = "";
  accountSelect.innerHTML = '<option value="">-- nessun conto --</option>';

  if (!res.ok) {
    alert("Errore nel caricamento dei conti (utente non trovato?)");
    return;
  }

  const data = await res.json();
  data.forEach(acc => {
    const tr = document.createElement("tr");
    tr.innerHTML =
      `<td>${acc.id}</td><td>${acc.currency}</td><td>${acc.balance.toFixed(2)}</td>`;
    tableBody.appendChild(tr);

    const opt = document.createElement("option");
    opt.value = acc.id;
    opt.textContent = `${acc.id} (${acc.currency})`;
    accountSelect.appendChild(opt);
  });
}

// altri metodi inclusi nel file reale...
```