<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Money Tracker</title>

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

</head>

<body>

<div class="container">

<h1>Money Tracker</h1>

<form id="transaction-form">

<label for="description">Description:</label>

<input type="text" id="description" name="description" required>

<label for="amount">Amount:</label>

<input type="number" id="amount" name="amount" required>

<select id="type" name="type">

<option value="expense">Expense</option>

<option value="income">Income</option>

</select>

<button type="submit">Add Transaction</button>

</form>

<ul id="transaction-list"></ul>

</div>

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

</body>

</html>

**styles.css:**

body {

font-family: Arial, sans-serif;

}

.container {

width: 80%;

margin: 0 auto;

}

h1 {

text-align: center;

}

form {

margin-bottom: 20px;

}

label {

display: block;

margin-bottom: 5px;

}

input[type="text"],

input[type="number"],

select {

width: 100%;

padding: 10px;

margin-bottom: 10px;

}

button {

padding: 10px 20px;

background-color: #007bff;

color: #fff;

border: none;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

ul {

list-style-type: none;

padding: 0;

}

li {

border: 1px solid #ccc;

padding: 10px;

margin-bottom: 5px;

}

.expense {

border-color: #ff0000;

}

.income {

border-color: #008000;

}

**scripts.js:**

document.getElementById('transaction-form').addEventListener('submit', function(event) {

event.preventDefault();

const description = document.getElementById('description').value;

const amount = document.getElementById('amount').value;

const type = document.getElementById('type').value;

const transaction = {

description: description,

amount: parseFloat(amount),

type: type

};

fetch('/transactions', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(transaction)

})

.then(response => response.json())

.then(data => {

console.log('Transaction added:', data);

displayTransaction(data);

})

.catch(error => console.error('Error adding transaction:', error));

});

function displayTransaction(transaction) {

const transactionList = document.getElementById('transaction-list');

const li = document.createElement('li');

li.textContent = `${transaction.description}: $${transaction.amount} (${transaction.type})`;

li.classList.add(transaction.type);

transactionList.appendChild(li);

}

**server.js:**

const express = require('express');

const bodyParser = require('body-parser');

const mongoose = require('mongoose');

const app = express();

const port = 3000;

// Connect to MongoDB

mongoose.connect('mongodb://localhost:27017/moneyTrackerDB', {

useNewUrlParser: true,

useUnifiedTopology: true

});

const db = mongoose.connection;

db.on('error', console.error.bind(console, 'connection error:'));

db.once('open', () => {

console.log('Connected to MongoDB');

});

// Define transaction schema

const transactionSchema = new mongoose.Schema({

description: String,

amount: Number,

type: String

});

const Transaction = mongoose.model('Transaction', transactionSchema);

// Middleware

app.use(bodyParser.urlencoded({ extended: false }));

app.use(bodyParser.json());

app.use(express.static('public'));

// Route for adding transaction

app.post('/transactions', (req, res) => {

const { description, amount, type } = req.body;

const newTransaction = new Transaction({

description: description,

amount: amount,

type: type

});

newTransaction.save((err, transaction) => {

if (err) {

console.error('Error adding transaction:', err);

res.status(500).json({ error: 'Internal server error' });

} else {

console.log('Transaction added:', transaction);

res.json(transaction);

}

});

});

// Start server

app.listen(port, () => {

console.log(`Server is running on http://localhost:${port}`);

});