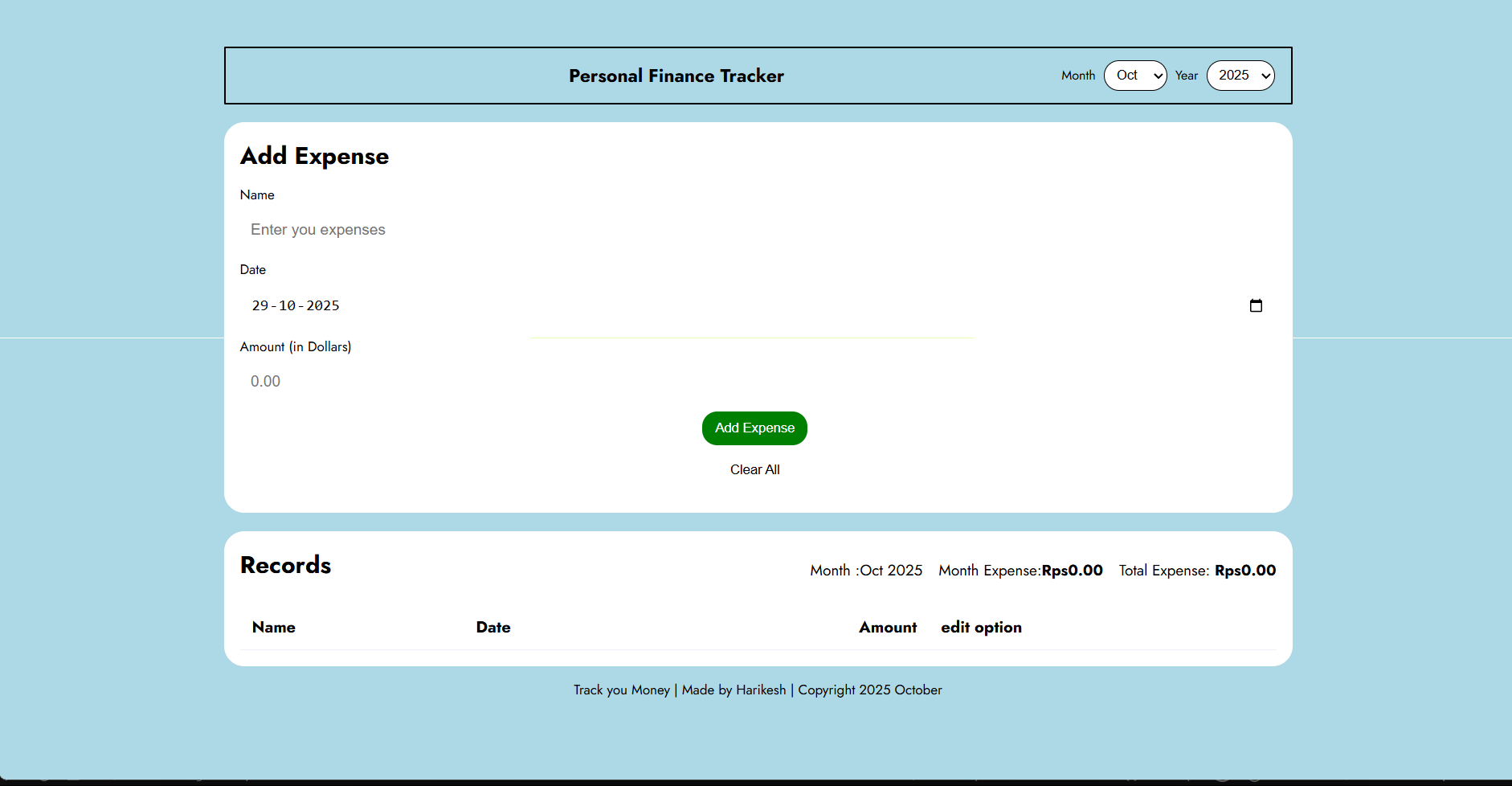
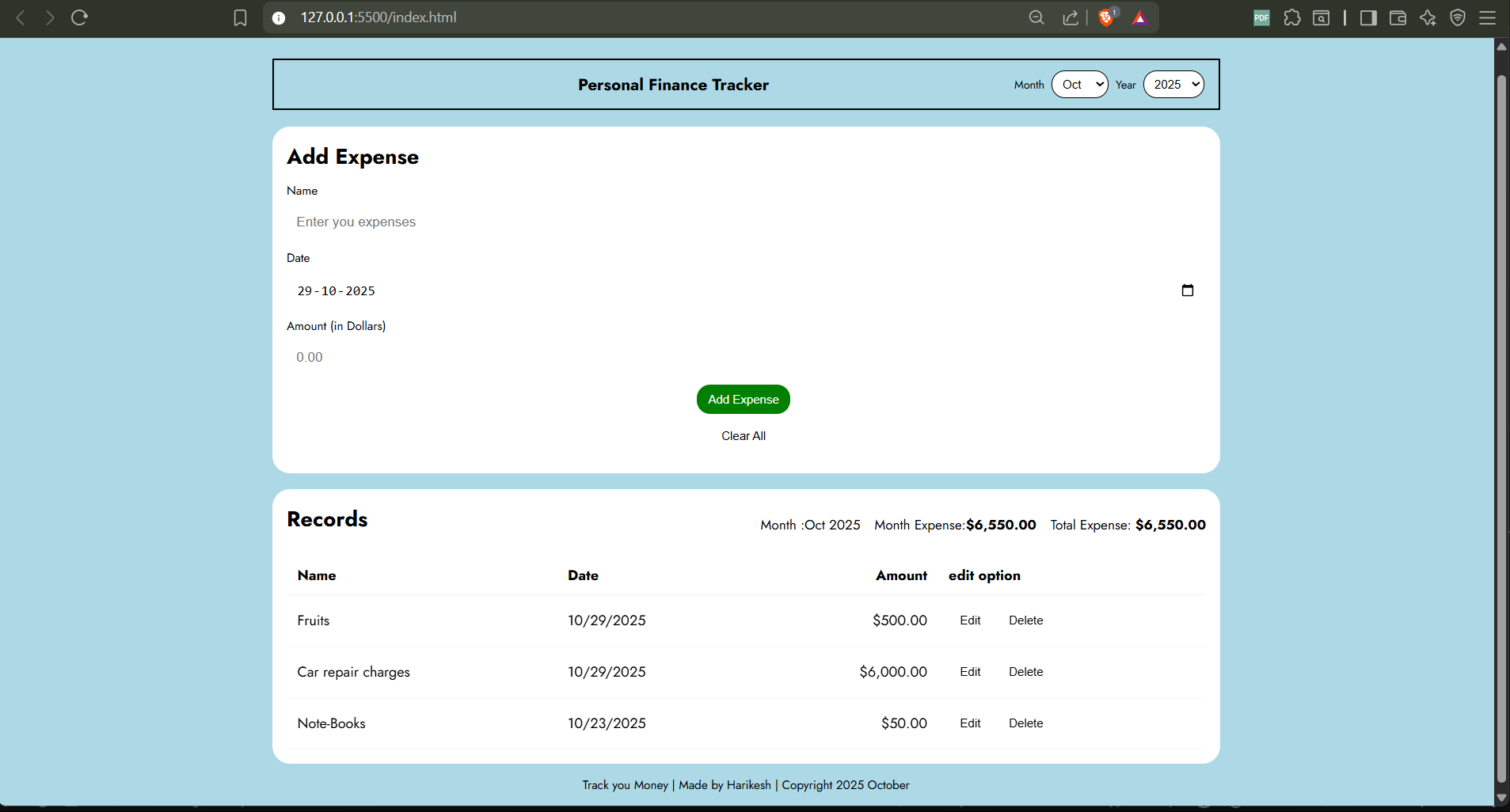
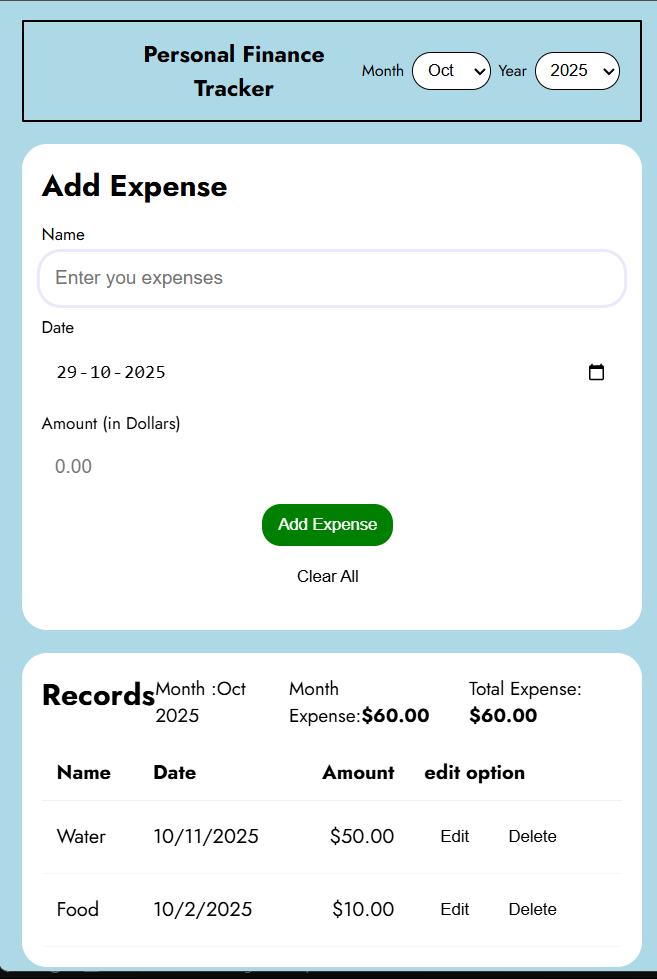
The personal Finance Tracker

Screen Shots of the project





Responsive page



Source code:

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Finance Tracker</title>

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

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<link href="https://fonts.googleapis.com/css2?family=Afacad+Flux:wght@100..1000&family=Ancizar+Serif:ital,wght@0,300..900;1,300..900&family=Barlow+Semi+Condensed:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&family=Bebas+Neue&family=Figtree:ital,wght@0,300..900;1,300..900&family=Fjalla+One&family=Jost:ital,wght@0,100..900;1,100..900&family=Lato:ital,wght@0,100;0,300;0,400;0,700;0,900;1,100;1,300;1,400;1,700;1,900&family=Oswald:wght@200..700&family=Roboto+Condensed:ital,wght@0,100..900;1,100..900&family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&family=Rubik:ital,wght@0,300..900;1,300..900&family=Signika:wght@300..700&family=Sora:wght@100..800&display=swap" rel="stylesheet">

</head>

<body>

<!-- This is Harikesh's personal Finance tracker web app -->

<!-- This is the place to enter the expenses and their names-->

<div class="app">

<header class="topbar">

<div class="brand">Personal Finance Tracker</div>

<div class="date-controls">

<label for="monthSelect">Month</label>

<select id="monthSelect" aria-label="Select month"></select>

<label for="yearSelect">Year</label>

<select id="yearSelect" aria-label="Select year"></select>

</div>

</header>

<main class="container">

<section class="entry-card">

<h2>Add Expense</h2>

<form id="expenseForm">

<div class="row">

<label for="name">Name</label>

<input id="name" name="name" type="text" placeholder="Enter you expenses" required>

</div>

<div class="row">

<label for="date">Date</label>

<input id="date" name="date" type="date" required>

</div>

<div class="row">

<label for="amount">Amount (in Dollars) </label>

<input id="amount" name="amount" type="number" step="0.01" placeholder="0.00" required>

</div>

<div class="row actions">

<button type="submit" id="submitBtn" class="btn primary">Add Expense</button>

<button type="button" id="clearBtn" class="btn">Clear All</button>

</div>

</form>

</section>

<section class="list-card">

<div class="list-header">

<h2>Records</h2>

<div class="totals">

<div>Month :<span id="filterLabel"></span></div>

<div>Month Expense:<strong id="periodTotal">0.00</strong></div>

<div>Total Expense: <strong id="allTotal">0.00</strong></div>

</div>

</div>

<div class="table-wrap">

<table id="expenseTable">

<thead>

<tr>

<th>Name</th>

<th>Date</th>

<th class="amount-col">Amount</th>

<th>edit option</th>

</tr>

</thead>

<tbody>

<!-- rows inserted here -->

</tbody>

</table>

</div>

</section>

</main>

<!--Footer of the page-->

<footer class="footer">Track you Money |

Made by Harikesh |

Copyright 2025 October

</footer>

</div>

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

</body>

</html>

Style.css

:root{

background-color: lightblue;

font-family: "Jost", sans-serif;

}

\*{box-sizing:border-box}

html,body{

height:100%;

padding: 0;

margin:0;

background:linear-gradient(180deg,#eef2ff 0%,100%);

color:black

}

.app{max-width:1100px;

margin:28px auto;

padding:18px

}

.topbar{display:flex;

align-items:center;

justify-content:space-between;

padding:12px 16px;

border-radius:1px;

border: 2px solid black;

}

.brand{font-weight:700;

font-size:1.15rem;

text-align: center;

margin-left: 50px;

flex: 1;

}

.date-controls{display:flex;

gap:8px;

align-items:center

}

.date-controls label{

font-size:0.8rem;

}

.date-controls select{padding:6px 8px;

border-radius:15px;

border:1px solid black;

background:white;

}

.container{display:grid;

grid-template-columns:1fr;

gap:18px;margin-top:18px

}

.entry-card,.list-card{background:white;

padding:16px;

border-radius:20px;

}

.entry-card h2,.list-card h2{

margin:0 0 12px 0

}

.row{display:flex;

flex-direction:column;

margin-bottom:10px

}

.row label{font-size:0.85rem;

margin-bottom:6px;

}

.row input{padding:10px;

border-radius:15px;

border:1px solid white;

font-size:0.95rem}

.row.actions{display:flex;

gap:8px;

align-items:center

}

.btn{padding:8px 12px;

border-radius:15px;

border:1px solid transparent;

background:#fff;cursor:pointer

}

.btn.primary{background:green;

color:#fff;

border-color:transparent;

}

.list-header{display:flex;

justify-content:space-between;

align-items:center;margin-bottom:12px

}

.totals{display:flex;

gap:16px;

align-items:center;

font-size:0.95rem

}

/\* To allow the table to add more responses\*/

.table-wrap{

overflow:auto

}

table{

width:100%;

border-collapse:collapse

}

thead th{text-align:left;

padding:10px 12px;

border-bottom:1px solid #eef2f8;

}

tbody tr{

transition:transform .12s ease,opacity .12s ease

}

tbody td{

padding:12px;

border-bottom:1px solid #f4f6fb

}

.amount-col{

text-align:right

}

.actions button{

margin-right:6px

}

.footer{text-align:center;

margin-top:14px;

font-size:0.85rem

}

/\* Done to make the web responsive \*/

/\* mobile \*/

@media (max-width:900px){

.container{grid-template-columns:1fr;}

.date-controls{gap:6px}

}

/\* small helpers \*/

.fade-in{animation:fadeIn .22s ease both}

@keyframes fadeIn{from{opacity:0;transform:translateY(6px)}to{opacity:1;transform:none}}

/\* accessible focus \*/

select:focus,input:focus,button:focus{outline:3px solid rgba(79,70,229,0.12);outline-offset:2px}

Script.js

// Personal Finance Tracker - reactive, responsive, persists to localStorage

(() => {

const monthSelect = document.getElementById('monthSelect');

const yearSelect = document.getElementById('yearSelect');

const expenseForm = document.getElementById('expenseForm');

const nameInput = document.getElementById('name');

const dateInput = document.getElementById('date');

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

const expenseTableBody = document.querySelector('#expenseTable tbody');

const periodTotalEl = document.getElementById('periodTotal');

const allTotalEl = document.getElementById('allTotal');

const filterLabel = document.getElementById('filterLabel');

const clearBtn = document.getElementById('clearBtn');

const STORAGE\_KEY = 'expenses\_v1';

let expenses = [];

function init(){

populateMonthYear();

dateInput.value = todayISO();

load();

render();

monthSelect.addEventListener('change', render);

yearSelect.addEventListener('change', render);

expenseForm.addEventListener('submit', onSubmit);

clearBtn.addEventListener('click', onClearAll);

}

function populateMonthYear(){

const now = new Date();

const months = ['All','Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'];

months.forEach((m,i)=>{

const opt = document.createElement('option');

opt.value = i; // 0 means All

opt.textContent = m;

monthSelect.appendChild(opt);

});

// years: from now-5 to now+2

const currentYear = now.getFullYear();

const start = currentYear - 5;

const end = currentYear + 2;

const allOpt = document.createElement('option');

allOpt.value = 'all';

allOpt.textContent = 'All';

yearSelect.appendChild(allOpt);

for(let y=end;y>=start;y--){

const o = document.createElement('option');

o.value = String(y);

o.textContent = y;

yearSelect.appendChild(o);

}

// default to current

monthSelect.value = String(now.getMonth()+1); // month index+1 so "All" is 0, Jan=1.. but we used values 0..12, so set to current month number

// adjust because earlier we set option value as index; index 0 is 'All', so month number should be now.getMonth()+1

yearSelect.value = String(currentYear);

}

function todayISO(){

const d = new Date();

return d.toISOString().slice(0,10);

}

function onSubmit(e){

e.preventDefault();

const name = nameInput.value.trim();

const date = dateInput.value;

const amount = parseFloat(amountInput.value);

if(!name || !date || Number.isNaN(amount) || amount <= 0){

alert('Please provide a valid name, date and amount (> 0).');

return;

}

const item = { id: Date.now(), name, date, amount };

expenses.push(item);

save();

render();

expenseForm.reset();

dateInput.value = todayISO();

nameInput.focus();

}

function onClearAll(){

if(!expenses.length) return alert('No records to clear.');

if(confirm('Clear all expense records? This cannot be undone.')){

expenses = [];

save();

render();

}

}

function load(){

try{

const raw = localStorage.getItem(STORAGE\_KEY);

expenses = raw ? JSON.parse(raw) : [];

}catch(err){

console.error('Failed to load expenses', err);

expenses = [];

}

}

function save(){

try{localStorage.setItem(STORAGE\_KEY, JSON.stringify(expenses));}catch(e){console.error('save error',e)}

}

function render(){

// compute filter

const monthVal = Number(monthSelect.value); // 0 = All, 1..12 months

const yearVal = yearSelect.value;

const filtered = expenses.filter(it => {

if(yearVal !== 'all' && String(new Date(it.date).getFullYear()) !== String(yearVal)) return false;

if(monthVal && monthVal !== 0){

// monthVal 1..12

return (new Date(it.date).getMonth()+1) === monthVal;

}

return true;

});

// update labels and totals

filterLabel.textContent = (monthVal && monthVal!==0 ? `${monthName(monthVal)} ` : '') + (yearVal==='all' ? '— All years' : yearVal);

const periodTotal = filtered.reduce((s,i)=>s + Number(i.amount),0);

const allTotal = expenses.reduce((s,i)=>s + Number(i.amount),0);

periodTotalEl.textContent = formatCurrency(periodTotal);

allTotalEl.textContent = formatCurrency(allTotal);

// render table rows

expenseTableBody.innerHTML = '';

// sort descending by date

filtered.sort((a,b)=> new Date(b.date) - new Date(a.date));

for(const it of filtered){

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

tr.className = 'fade-in';

const nameTd = document.createElement('td');

nameTd.textContent = it.name;

const dateTd = document.createElement('td');

dateTd.textContent = formatDateLocal(it.date);

const amtTd = document.createElement('td');

amtTd.className = 'amount-col';

amtTd.textContent = formatCurrency(it.amount);

const actionsTd = document.createElement('td');

actionsTd.className = 'actions';

const delBtn = document.createElement('button');

delBtn.className = 'btn';

delBtn.textContent = 'Delete';

delBtn.addEventListener('click', ()=> onDelete(it.id));

const editBtn = document.createElement('button');

editBtn.className = 'btn';

editBtn.textContent = 'Edit';

editBtn.addEventListener('click', ()=> onEdit(it.id));

actionsTd.appendChild(editBtn);

actionsTd.appendChild(delBtn);

tr.appendChild(nameTd);

tr.appendChild(dateTd);

tr.appendChild(amtTd);

tr.appendChild(actionsTd);

expenseTableBody.appendChild(tr);

}

}

// This will delete in the alert box

function onDelete(id){

if(confirm('Delete this record?')){

expenses = expenses.filter(e => e.id !== id);

save();

render();

}

}

//This will edit in the alert box

function onEdit(id){

const item = expenses.find(e => e.id === id);

if(!item) return;

const newName = prompt('Edit name', item.name);

if(newName === null) return; // cancelled

const newDate = prompt('Edit date (YYYY-MM-DD)', item.date);

if(newDate === null) return;

const newAmtStr = prompt('Edit amount', String(item.amount));

if(newAmtStr === null) return;

const newAmt = parseFloat(newAmtStr);

if(!newName.trim() || !newDate || Number.isNaN(newAmt) || newAmt <= 0){

alert('Invalid values — edit cancelled.');

return;

}

item.name = newName.trim();

item.date = newDate;

item.amount = newAmt;

save();

render();

}

function formatCurrency(v){

return v === 0 ? 'Rps0.00' : v.toLocaleString(undefined,{style:'currency',currency:'USD'});

}

function formatDateLocal(iso){

const d = new Date(iso + 'T00:00:00');

return d.toLocaleDateString();

}

function monthName(n){

const m = ['','Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'];

return m[n] || '';

}

// expose init

init();

})();