



INSTITUTE OF AERONAUTICAL ENGINEERING (AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

Complex Problem-Solving Self-Assessment Form

1	Name of the Student	Bhanu Rishikesh Reddy Puchalapalli	
2	Roll Number	25951A6632	
3	Branch and Section	CSE-(AI&ML) - A	
4	Program	B. Tech	
5	Course Name	Front-End Web Development	
6	Course Code	ACSE04	
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles		
	EC	Profiles	(✓)
	EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	✓
	EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
	EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	
	EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	✓
	EC 5	Conceptualises alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	✓
	EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	

EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	
EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	✓
EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	

	EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	✓
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	✓
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	✓
8	Please tick (✓) relevant Course Outcomes (COs) Covered		
	CO	Course Outcomes	(✓)
	CO 1	Describe language basics like alphabet, strings, grammars, productions, derivations, and Chomsky hierarchy, construct DFA, NFA, and conversion of NFA to DFA, Moore and Mealy machines and interpret differences between them.	✓
	CO 2	Recognize regular expressions, formulate, and build equivalent finite automata for various languages.	✓
	CO 3	Identify closure, and decision properties of the languages and prove the membership.	✓
	CO4	Demonstrate context-free grammars, check the ambiguity of the grammar, and design equivalent PDA to accept the context-free languages.	
	CO 5	Uses mathematical tools and abstract machine models to solve complex problems.	✓
	CO 6	Analyze and distinguish between decidable and undecidable problems.	✓
9	Course ELRVVideo Lectures Viewed		<div> <div>Number of Videos</div> <div>Viewing time in Hours</div> </div> <div> <div>-</div> <div>-</div> </div>
10	Justify your understanding of WK1		-

11	Justify your understanding of WK2 – WK9	-
12	How many WKs from WK2 to WK9 were implanted?	-
	Mention them	-

Date: 05-12-2025

Rishi

Signature of the Student



INSTITUTE OF AERONAUTICAL ENGINEERING **(AUTONOMOUS)**

Dundigal - 500 043, Hyderabad, Telangana

TASK ASSESSMENT

ON

PAY-SHARE

RISHI

25951A6632



INSTITUTE OF AERONAUTICAL ENGINEERING (AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

PAY - SHARE

A project report submitted in partial fulfillments
of requirements for the award of the
degree of Bachelor of Technology in

CSE (Artificial Intelligence & Machine Learning)

By

BHANU RISHIKESH REDDY PUCHALAPALLI

25951A6632

DECLARATION

I certify that

- a. The work contained in this report is original and has been done by me under the guidance of my supervisor (s).
- b. The work has not been submitted to any other Institute for any degree or diploma.
- c. I have followed the guidelines provided by the Institute for preparing the report.
- d. I have conformed to the norms and guidelines given in the Code of Conduct of the Institute.
- e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Rishi

Place: Hyderabad

Signature of the Student

Date: 05-12-2025

CERTIFICATE

This is to certify that the project report entitled **PAYSHARE** submitted by **Bhanu Rishikesh Reddy Puchalapalli** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **CSE - (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** is a Bonafide record of work carried out by his guidance and supervision. The Contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Supervisor

Head of the Department

Date: 05-12-2025

Principal

APPROVAL SHEET

This project report entitled **PayShare** submitted by Bhanu Rishikesh Reddy Puchalapalli is approved for the award of the Degree Bachelor of Technology in Branch **CSE (Artificial Intelligence & Machine Learning)**.

Examiner

Supervisor(s)

Principal

Date: 05-12-2025

Place: Hyderabad

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide **Mr. Vidyasagar Vidapu, Assistant Professor, Department of CSE – (Artificial Intelligence and Machine Learning)**, for his kind help and for giving me the necessary guidance and valuable suggestions for this project work.

I am grateful to **Dr. M. Purushotham Reddy, Professor and Head of the Department**, Department of **CSE (Artificial Intelligence & Machine Learning)**, for extending his support to carry on this project work. I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

I express my sincere gratitude to **Dr. L. V. Narasimha Prasad, Professor and Principal** who has been a great source of information for my work.

I thank our college management and respected **Sri M. Rajashekar Reddy, Chairman, IARE, Dundigal** for providing me with the necessary infrastructure to conduct the project work.

I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

ABSTRACT

PayShare is a web-based financial management platform designed to simplify shared expense tracking and group payment splitting through an intuitive and user-friendly front-end interface. This project focuses on developing a responsive and interactive application using core web technologies such as HTML5, CSS3, and JavaScript. The primary objective of PayShare is to help users manage group expenses transparently by accurately calculating individual contributions and balances in shared financial activities.

The platform provides a clean and organized interface that allows users to add expenses, assign participants, and automatically split costs among group members. Interactive elements such as dynamic expense cards, real-time balance updates, and summary views enhance usability and reduce calculation errors. By following modern UI/UX principles, including responsive layouts and minimalistic design, PayShare ensures seamless accessibility across mobile, tablet, and desktop devices.

A key focus of the project is improving clarity and trust in shared financial management. JavaScript-driven logic enables instant recalculation of expenses and balances without requiring backend processing, making the application lightweight and efficient. Visual consistency and structured data presentation help users clearly understand who owes what, minimizing confusion and financial disputes within groups.

Overall, PayShare demonstrates the effective application of front-end web development concepts in the finance domain. It highlights how interactive design, usability, and client-side processing can be combined to create a practical financial tool that simplifies shared expense management and promotes transparency in group payments.

CONTENTS

Name of Contents	Page No.
Title Page	I
Declaration	II
Certificate	III
Approval Sheet	IV
Acknowledgement	V
Abstract	VI
Contents	VII
Chapter 1- Introduction	8-9
1.1 Problem Statement	8
1.2 Introduction	8
1.3 Requirements	9
1.4 Prerequisites	10
1.5 Technologies used	10
Chapter 2 - Review of Relevant Literature	11-12
Chapter 3- Methodology	13-14
Chapter 4- Results and Discussions	15-16
Chapter 5- Conclusions and Future Scope	17-18
5.1 Conclusion	17
5.2 Future Scope	18
References	19

CHAPTER 1

INTRODUCTION

ABSTRACT

PayShare is a web-based financial management platform designed to simplify shared expense tracking and group payment splitting through an intuitive and user-friendly front-end interface. This project focuses on developing a responsive and interactive application using core web technologies such as HTML5, CSS3, and JavaScript. The primary objective of PayShare is to help users manage group expenses transparently by accurately calculating individual contributions and balances in shared financial activities.

The platform provides a clean and organized interface that allows users to add expenses, assign participants, and automatically split costs among group members. Interactive elements such as dynamic expense cards, real-time balance updates, and summary views enhance usability and reduce calculation errors. By following modern UI/UX principles, including responsive layouts and minimalistic design, PayShare ensures seamless accessibility across mobile, tablet, and desktop devices.

A key focus of the project is improving clarity and trust in shared financial management. JavaScript-driven logic enables instant recalculation of expenses and balances without requiring backend processing, making the application lightweight and efficient. Visual consistency and structured data presentation help users clearly understand who owes what, minimizing confusion and financial disputes within groups.

Overall, PayShare demonstrates the effective application of front-end web development concepts in the finance domain. It highlights how interactive design, usability, and client-side processing can be combined to create a practical financial tool that simplifies shared expense management and promotes transparency in group payments.

CONTENTS

1. Introduction
 - 1.1 Problem Statement
 - 1.2 Introduction
 - 1.3 Objectives of the Project
 - 1.4 Scope of the Project
 - 1.5 Requirements
 - 1.6 Prerequisites
 - 1.7 Technologies Used
 2. Review of Relevant Literature
 3. System Analysis and Design
 - 3.1 System Architecture
 - 3.2 Gamification Strategy
 - 3.3 User Interface Design
 4. Methodology
 5. Implementation Details
 6. Results and Discussion
 7. Conclusion and Future Scope
 8. References
-

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In today's collaborative lifestyle, individuals frequently engage in shared expenses such as group trips, shared accommodation, dining, and event planning. Managing these expenses manually often leads to confusion, calculation errors, delayed settlements, and misunderstandings among participants. Traditional methods like handwritten notes, spreadsheets, or informal messaging lack accuracy, transparency, and real-time balance tracking.

PayShare aims to address these challenges by providing a shared expense and group payment splitter that automatically calculates individual contributions and balances. Despite the availability of financial apps, many users seek a lightweight, easy-to-use, and visually intuitive solution that focuses purely on expense splitting without unnecessary complexity.

The platform enables users to add expenses, assign participants, and instantly view who owes whom and how much. By centralizing offering clarity and automation in shared financial activities, PayShare helps reduce disputes, improves trust, and ensures fair expense management. Hence, a modern front-end financial solution like PayShare is essential for simplifying group expense handling.

1.2 Introduction

PayShare is a dynamic, interactive, web-based financial application designed to manage shared expenses and split group payments efficiently. In group financial scenarios, maintaining transparency and fairness is crucial, yet often difficult using traditional methods.

PayShare provides a centralized platform with the following core components:

- **Expense Manager** – for adding shared expenses with amount and description
- **Participant Splitter** – for dividing costs equally or selectively among group members
- **Balance Summary Dashboard** – for tracking individual dues and settlements

With a clean, responsive interface and real-time calculations, PayShare ensures users can easily understand financial responsibilities within a group. The platform transforms complex expense sharing into a simple, reliable, and stress-free process, promoting better financial coordination and accountability.

1.3 Requirements

1. Functional Requirements (Front-End)

1.1 Expense Management

- FR1: The interface must allow users to add expenses with amount and description
- FR2: Users must be able to assign expenses to multiple participants
- FR3: Expenses must be editable and removable
- FR4: Each expense should update balances instantly

1.2 Payment Splitting

- FR5: The system must automatically split expenses among selected users
- FR6: Individual balances (owed/owing) must be calculated dynamically
- FR7: The dashboard must clearly display who owes whom

1.3 Dashboard Overview

- FR8: The main screen must summarize total expenses and balances
- FR9: All updates must occur instantly using JavaScript
- FR10: Client-side storage must retain expense data across sessions

2. Non-Functional Requirements

- NFR1: Responsive interface for mobile, tablet, and desktop
- NFR2: Simple and clear financial visualization
- NFR3: Fast performance with instant UI updates
- NFR4: Consistent design system and readable layout
- NFR5: Accurate calculations with zero rounding errors
- NFR6: Reliable local storage persistence

1.4 Pre-requisites

1. Technical Pre-requisites

- HTML5 for page structure
- CSS3 for layout, styling, and responsiveness
- JavaScript for calculations, logic, and interactivity
- Understanding of DOM manipulation
- Basic knowledge of client-side storage

2. Tool Pre-requisites

- Code editor (VS Code, Sublime, Atom)
- Modern web browsers (Chrome, Edge, Firefox)
- Git/GitHub for version control
- UI planning tools (optional)

3. User Pre-requisites

- Basic understanding of shared expenses
- Ability to input expense details

- Familiarity with simple web interfaces

1.5 Technologies Used

1. **HTML5** – Structure of expense cards and dashboard
2. **CSS3** – Styling, layouts, and responsive design
3. **JavaScript** – Expense logic, calculations, and UI updates
4. **JSON** – Storing sample expense data
5. **Git & GitHub** – Version control and deployment

Optional Add-ons

- Local Storage for data persistence
- Chart.js for expense visualization
- Bootstrap/Tailwind for UI styling

CHAPTER 2 - REVIEW OF RELEVANT LITERATURE

Studies in personal and group finance management emphasize the importance of transparency and automation in expense sharing. Research shows that automated calculations reduce conflicts and increase trust among group members. Financial clarity plays a key role in maintaining healthy personal and social relationships.

Web-based finance tools commonly utilize front-end technologies such as HTML, CSS, and JavaScript due to their flexibility and real-time interaction capabilities. LocalStorage is widely adopted for client-side data persistence in lightweight financial applications.

PayShare aligns with these findings by combining automation, clarity, and user-friendly design, making it an effective solution for managing shared financial responsibilities.

CHAPTER 3

METHODOLOGY

PayShare follows a front-end focused development methodology:

- **UI/UX Design:** Designing expense input forms, participant lists, and summary dashboards
- **Front-End Development:**
 - o HTML5 for structural elements
 - o CSS3 for clean and responsive design
 - o JavaScript for expense splitting logic and balance calculation
- **Data Persistence:** LocalStorage for saving expenses and balances
- **Testing:**
 - o Expense calculation accuracy
 - o UI responsiveness
 - o Balance update reliability
- **Deployment:** Hosted using GitHub Pages

This methodology ensures PayShare remains lightweight, interactive, and reliable for real-time expense management.

CODE:

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>PayShare — Group Expense Splitter</title>

  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

```

<style>
:root{
  --bg: #fbfbfd;
  --panel: #ffffff;
  --muted: #64748b;
  --accent: #9fc5ff;
  --accent-2: #b8f0d1;
  --danger: #ff7b7b;
  --radius: 12px;
  --shadow: 0 6px 18px rgba(16,24,40,0.06);
  --maxw:1100px;
  font-family: Inter, ui-sans-serif, system-ui, -apple-system, 'Segoe UI',
  Roboto, Arial;
}
*{box-sizing:border-box}
body{
  margin:0;padding:28px;background:linear-gradient(180deg,var(--
bg),#f7fbff 160%);color:#0f1724;
  -webkit-font-smoothing:antialiased; -moz-osx-font-smoothing:grayscale;
  display:flex;justify-content:center;
}
.wrap{max-width:var(--maxw);width:100%;display:grid;grid-template-
columns:360px 1fr;gap:20px}

.panel{background:var(--panel);border-radius:var(--radius);box-
shadow:var(--shadow);padding:16px}
.brand{display:flex;gap:12px;align-items:center;margin-bottom:12px}
.logo{width:44px;height:44px;border-radius:10px;background:linear-
gradient(135deg,var(--accent),var(--accent-2));
  display:flex;align-items:center;justify-content:center;color:white;font-
weight:800}
h1{font-size:18px;margin:0}
.small{font-size:13px;color:var(--muted)}

label{display:block;font-weight:600;margin-bottom:6px;font-size:13px}
input[type="text"], input[type="number"], input[type="date"], select,
textarea{

```

```

    width:100%;padding:10px;border-radius:10px;border:1px solid
    rgba(15,23,42,0.06);background:#fff;font-size:14px;
  }
  .row{display:flex;gap:8px}
  .row .half{flex:1}

  .btn{background:linear-gradient(90deg,var(--accent),var(--accent-
  2));color:white;border:0;padding:10px 12px;border-radius:10px;font-
  weight:700;cursor:pointer}
  .btn.ghost{background:transparent;border:1px solid
  rgba(15,23,42,0.06);color:var(--muted);font-weight:600}

  .main{display:flex;flex-direction:column;gap:16px}
  .card{background:var(--panel);border-radius:12px;padding:14px;box-
  shadow:var(--shadow)}
  .stats{display:flex;gap:12px;flex-wrap:wrap}
  .stat{flex:1;min-width:150px;padding:12px;border-
  radius:10px;background:linear-gradient(180deg,#fff,#fbfdff)}
  .stat .num{font-weight:800;font-size:20px}

  .charts{display:grid;grid-template-columns:1fr 360px;gap:16px}
  canvas{max-width:100%;height:auto}

  table{width:100%;border-collapse:collapse;font-size:14px}
  th,td{padding:8px;text-align:left;border-top:1px solid rgba(15,23,42,0.04)}
  thead th{font-size:13px;color:var(--muted);font-weight:700}

  .chip{display:inline-block;padding:6px 10px;border-
  radius:999px;background:rgba(111,124,255,0.08);color:var(--muted);font-
  weight:700;font-size:13px}

  @media (max-width:1000px){
    .wrap{grid-template-columns:1fr;padding:12px}
    .charts{grid-template-columns:1fr}
  }
</style>
</head>

```

```

<body>
  <div class="wrap">
    <!-- LEFT: controls -->
    <div class="panel">
      <div class="brand">
        <div class="logo">PS</div>
        <div>
          <h1>PayShare</h1>
          <div class="small">Split group expenses quickly</div>
        </div>
      </div>

      <p class="small" style="margin:8px 0 12px">Add group members, then
log shared expenses. PayShare calculates each member's net owed or owed-
to amount.</p>

      <div style="margin-bottom:12px">
        <label>Group members (comma separated)</label>
        <input id="members-input" placeholder="e.g. Alice, Bob, Charlie" />
        <div style="display:flex;gap:8px;margin-top:8px">
          <button class="btn" id="set-members">Set Members</button>
          <button class="btn ghost" id="seed-sample">Seed sample</button>
        </div>
        <div class="small muted" style="margin-top:8px">Members currently:
<span id="members-list" class="chip">—</span></div>
      </div>

      <hr style="border:none;border-top:1px solid
rgba(15,23,42,0.04);margin:12px 0" />

      <div>
        <label for="exp-desc">Expense description</label>
        <input id="exp-desc" placeholder="Dinner, Taxi, Hotel..." />
      </div>

      <div class="row" style="margin-top:10px">
        <div class="half">

```

```

    <label for="exp-amount">Amount (currency)</label>
    <input id="exp-amount" type="number" min="0" step="0.01" />
  </div>
  <div class="half">
    <label for="exp-date">Date</label>
    <input id="exp-date" type="date" />
  </div>
</div>

```

```

<div style="margin-top:10px">
  <label for="paid-by">Paid by</label>
  <select id="paid-by"></select>
</div>

```

```

<div style="margin-top:10px">
  <label for="split-type">Split type</label>
  <select id="split-type">
    <option value="equal">Equal split</option>
    <option value="custom">Custom shares (comma separated)</option>
  </select>
</div>

```

```

<div id="custom-shares-area" style="display:none;margin-top:10px">
  <label>Custom shares (comma separated, same order as
members)</label>
  <input id="custom-shares" placeholder="e.g. 10,20,30" />
  <div class="small muted">Shares can be amounts or weights — they will
be normalized.</div>
</div>

```

```

<div style="display:flex;gap:8px;margin-top:12px">
  <button class="btn" id="add-expense">Add Expense</button>
  <button class="btn ghost" id="clear-expense">Clear</button>
</div>

```

```

<div style="margin-top:14px;display:flex;gap:8px">
  <button class="btn ghost" id="export-csv">Export CSV</button>

```

```
<button class="btn ghost" id="clear-all">Clear All</button>
</div>
```

```
<div style="margin-top:12px" class="small muted">
  Data stored locally in your browser. No accounts required.
</div>
</div>
```

```
<!-- RIGHT: dashboard -->
<div class="main">
  <div class="card">
    <div style="display:flex;justify-content:space-between;align-
items:center">
      <div>
        <div style="font-weight:800">Group Overview</div>
        <div class="small">Totals and balances</div>
      </div>
      <div class="small muted" id="last-updated">—</div>
    </div>
```

```
<div style="margin-top:12px" class="stats">
  <div class="stat">
    <div class="small">Total spent</div>
    <div class="num" id="total-spent">0</div>
    <div class="small muted">All-time</div>
  </div>
  <div class="stat">
    <div class="small">Net positions</div>
    <div class="num" id="net-positive">0</div>
    <div class="small muted">People owed (positive)</div>
  </div>
  <div class="stat">
    <div class="small">Outstanding (negative)</div>
    <div class="num" id="net-negative">0</div>
    <div class="small muted">People owing</div>
  </div>
</div>
```

</div>

<div class="card charts">

<div>

<div style="display:flex;justify-content:space-between;align-items:center">

<div style="font-weight:700">Expense distribution</div>

<div class="small muted">By member total paid</div>

</div>

<canvas id="pieChart" style="margin-top:8px;max-height:320px"></canvas>

</div>

<div>

<div style="font-weight:700">Member balances</div>

<canvas id="barChart" style="margin-top:8px;max-height:260px"></canvas>

<div style="margin-top:10px" class="small muted">Positive = others owe them. Negative = they owe others.</div>

</div>

</div>

<div class="card">

<div style="display:flex;justify-content:space-between;align-items:center">

<div style="font-weight:700">Expenses History</div>

<div class="small muted">Manage entries</div>

</div>

<div style="margin-top:8px;overflow:auto">

<table>

<thead><tr><th>Date</th><th>Description</th><th>Paid</th><th>Split</th><th>Amount</th><th></th></tr></thead>

<tbody id="expenses-tbody"></tbody>

</table>

</div>

</div>


```

<div class="card" id="balances-card">
  <div style="font-weight:700">Member balances</div>
  <div id="balances-area" style="margin-top:8px;display:flex;gap:8px;flex-
wrap:wrap"></div>
</div>
</div>
</div>

<script>
/* ----- State & storage ----- */
const LS_KEY = 'payshare-data-v1';
let state = { members: [], expenses: [] };

function load(){
  try{ const raw = localStorage.getItem(LS_KEY); if(raw) state =
    JSON.parse(raw); }
  catch(e){ console.warn(e); state = { members:[], expenses:[] } }
}
function save(){ localStorage.setItem(LS_KEY, JSON.stringify(state)); }

function uid(){ return 'x_'+Math.random().toString(36).slice(2,9); }
function todayISO(){ return new Date().toISOString().slice(0,10); }
function fmtCurrency(n){ return
  Number(n).toLocaleString(undefined,{ maximumFractionDigits:2 }); }

/* ----- Member management -----*/
function setMembersFromInput(){
  const raw = document.getElementById('members-input').value.trim();
  if(!raw) return alert('Enter at least one member name (comma separated).');
  const arr = raw.split(',').map(s=>s.trim()).filter(Boolean);
  if(arr.length===0) return alert('No valid member names found. ');
  state.members = arr;
  save();
  renderMembers();
  populatePaidBySelect();
  renderAll();
}

```

```

function renderMembers(){
  const el = document.getElementById('members-list');
  el.textContent = state.members.length ? state.members.join(', ') : '—';
  document.getElementById('members-input').value = state.members.join(', ');
}
function populatePaidBySelect(){
  const sel = document.getElementById('paid-by');
  sel.innerHTML = "";
  state.members.forEach(m=>{
    const opt = document.createElement('option'); opt.value = m;
    opt.textContent = m; sel.appendChild(opt);
  });
}

/* ----- Expense model & calculations -----
Expense: { id, date, desc, amount, paidBy, splitType, shares: [] }
shares: if equal -> ignored; if custom -> array of numbers same length as
members.
*/
function addExpenseFromForm(){
  if(state.members.length===0) return alert('Define group members first. ');
  const desc = document.getElementById('exp-desc').value.trim() || 'Expense';
  const amount = Number(document.getElementById('exp-amount').value) || 0;
  if(amount <= 0) return alert('Enter a valid amount. ');
  const date = document.getElementById('exp-date').value || todayISO();
  const paidBy = document.getElementById('paid-by').value ||
    state.members[0];
  const splitType = document.getElementById('split-type').value;
  let shares = null;
  if(splitType === 'custom'){
    const raw = document.getElementById('custom-shares').value.trim();
    if(!raw) return alert('Enter custom shares (comma separated) or choose equal split. ');
    const arr = raw.split(',').map(s=>Number(s.trim())).map(n=>isNaN(n)?0:n);
    if(arr.length !== state.members.length) return alert('Custom shares must match number of members, in same order. ');
    shares = arr;
  }
}

```

```

}
const expense = { id: uid(), desc, amount, date, paidBy, splitType, shares };
state.expenses.push(expense);
save();
clearExpenseForm();
renderAll();
}

```

```

function clearExpenseForm(){
  document.getElementById('exp-desc').value="";
  document.getElementById('exp-amount').value="";
  document.getElementById('exp-date').value = todayISO();
  document.getElementById('custom-shares').value="";
  document.getElementById('split-type').value='equal';
  document.getElementById('custom-shares-area').style.display='none';
}

```

/* calculate balances:

For each expense:

- determine each member's share (equal or normalized custom)
- payer paid full amount; each member owes their share

Balance per member = totalPaidByMember - totalShareOwed

*/

```

function calculateBalances(){
  const members = state.members.slice();
  const balances = { }; members.forEach(m=>balances[m]=0);
  let totalSpent = 0;
  state.expenses.forEach(exp=>{
    const amt = Number(exp.amount) || 0;
    totalSpent += amt;
    // shares
    let sharesArr = [];
    if(exp.splitType === 'equal' || !exp.shares){
      const equalShare = amt / Math.max(1, members.length);
      sharesArr = members.map(()=>equalShare);
    } else {
      // normalize custom shares

```

```

    const raw = exp.shares.map(n=>Number(n)||0);
    const sum = raw.reduce((s,x)=>s+x,0) || 1;
    sharesArr = raw.map(r => (r / sum) * amt);
  }
  // payer paid full
  balances[exp.paidBy] = (balances[exp.paidBy] || 0) + amt;
  // subtract owed shares
  members.forEach((m, idx)=>{
    balances[m] = (balances[m] || 0) - sharesArr[idx];
  });
});
return {balances, totalSpent};
}

/* ----- Rendering & charts ----- */
let pieChart=null, barChart=null;

function renderAll(){
  renderMembers();
  populatePaidBySelect();
  // render expenses table
  renderExpensesTable();
  // balances
  const {balances, totalSpent} = calculateBalances();
  document.getElementById('total-spent').textContent =
    fmtCurrency(totalSpent);
  document.getElementById('last-updated').textContent = state.expenses.length
    ? `Last: ${state.expenses[state.expenses.length-1].date}` : '—';

  // split balances into arrays for charts
  const labels = state.members.slice();
  const paidTotals = labels.map(m=>{
    // total paid by member
    return state.expenses.reduce((s,e)=>{
      return s + ((e.paidBy === m) ? Number(e.amount) : 0);
    },0);
  });
}

```

```

const balVals = labels.map(m=>Number((balances[m]||0).toFixed(2)));

drawPie(labels, paidTotals);
drawBar(labels, balVals);

// show net positive/negative totals
const pos = balVals.filter(v=>v>0).reduce((s,v)=>s+v,0);
const neg = balVals.filter(v=>v<0).reduce((s,v)=>s+v,0);
document.getElementById('net-positive').textContent = fmtCurrency(pos);
document.getElementById('net-negative').textContent =
  fmtCurrency(Math.abs(neg));

renderBalancesArea(balances);
save();
}

function drawPie(labels, data){
  const ctx = document.getElementById('pieChart').getContext('2d');
  const bg = generateColors(labels.length);
  if(pieChart){
    pieChart.data.labels = labels;
    pieChart.data.datasets[0].data = data;
    pieChart.data.datasets[0].backgroundColor = bg;
    pieChart.update();
    return;
  }
  pieChart = new Chart(ctx,{
    type:'pie',
    data:{labels, datasets:[{data, backgroundColor:bg, borderWidth:0}]},
    options:{plugins:{legend:{position:'bottom'}}},maintainAspectRatio:false}
  });
}

function drawBar(labels, data){
  const ctx = document.getElementById('barChart').getContext('2d');
  const bg = data.map(v => v>=0 ? 'rgba(124,180,245,0.95)' :
    'rgba(255,170,170,0.95)');
  if(barChart){

```

```

    barChart.data.labels = labels;
    barChart.data.datasets[0].data = data;
    barChart.data.datasets[0].backgroundColor = bg;
    barChart.update();
    return;
}
barChart = new Chart(ctx,{
  type:'bar',
  data:{labels, datasets:[{label:'Net balance', data, backgroundColor:bg}]},
  options:{plugins:{legend:{display:false}},scales:{y:{beginAtZero:true}}}
});
}

function renderExpensesTable(){
  const tbody = document.getElementById('expenses-tbody');
  tbody.innerHTML = "";
  if(state.expenses.length === 0){
    tbody.innerHTML = '<tr><td colspan="6" class="small muted">No
    expenses yet</td></tr>';
    return;
  }
  // newest first
  const arr = state.expenses.slice().sort((a,b)=> b.date.localeCompare(a.date));
  arr.forEach(e=>{
    const splitText = e.splitType === 'equal' ? 'Equal' : 'Custom';
    const tr = document.createElement('tr');
    tr.innerHTML = `<td>${e.date}</td>
      <td>${e.desc}</td>
      <td>${e.paidBy}</td>
      <td class="small">${splitText}</td>
      <td>${fmtCurrency(e.amount)}</td>
      <td style="text-align:right">
        <button class="btn ghost"
onclick="editExpense('${e.id}')">Edit</button>
        <button class="btn ghost"
onclick="deleteExpense('${e.id}')">Delete</button>
      </td>`;
  });
}

```

```

tbody.appendChild(tr);
});
}

function renderBalancesArea(balances){
  const area = document.getElementById('balances-area');
  area.innerHTML = "";
  state.members.forEach(m=>{
    const v = Number((balances[m]||0).toFixed(2));
    const div = document.createElement('div');
    div.style.minWidth = '140px';
    div.style.padding = '10px';
    div.style.borderRadius = '10px';
    div.style.background = v>=0 ? 'linear-gradient(180deg,#fff,#f1fff8)' : 'linear-
    gradient(180deg,#fff,#fff6f6)';
    div.style.boxShadow = 'var(--shadow)';
    div.innerHTML = `

27


```

```

function editExpense(id){
  const e = state.expenses.find(x=>x.id===id);
  if(!e) return alert('Expense not found');
  // populate form and remove original to allow re-adding
  document.getElementById('exp-desc').value = e.desc;
  document.getElementById('exp-amount').value = e.amount;
  document.getElementById('exp-date').value = e.date;
  document.getElementById('paid-by').value = e.paidBy;
  document.getElementById('split-type').value = e.splitType;
  if(e.splitType === 'custom'){ document.getElementById('custom-shares-
    area').style.display='block'; document.getElementById('custom-
    shares').value = (e.shares||[]).join(','); }
  else { document.getElementById('custom-shares-area').style.display='none';
    document.getElementById('custom-shares').value=""; }
  // remove original
  state.expenses = state.expenses.filter(x=>x.id!==id);
  save();
  renderAll();
  window.scrollTo({top:0,behavior:'smooth'});
}

function deleteExpense(id){
  if(!confirm('Delete this expense?')) return;
  state.expenses = state.expenses.filter(x=>x.id!==id);
  save();
  renderAll();
}

function exportCSV(){
  if(state.expenses.length===0) return alert('No expenses to export');
  const rows = [['id','date','desc','paidBy','splitType','shares','amount']];
  state.expenses.forEach(e=>{
    rows.push([e.id,e.date,e.desc,e.paidBy,e.splitType, e.shares ?
      e.shares.join(';') : "", Number(e.amount).toFixed(2)]);
  });
  const csv =
    rows.map(r=>r.map(c=>`"${String(c).replace(/"/g,"")}"`).join(',')).join('\n');

```



```

const blob = new Blob([csv], {type:'text/csv'});
const url = URL.createObjectURL(blob); const a =
  document.createElement('a'); a.href=url; a.download='payshare_export.csv';
  a.click(); URL.revokeObjectURL(url);
}

```

```

function clearAll(){
  if(!confirm('Clear all members and expenses?')) return;
  state = {members:[], expenses:[]};
  save();
  renderAll();
}

```

```

/* seed sample */
function seedSample(){
  state.members = ['Alice','Bob','Charlie'];
  state.expenses = [
    { id: uid(), date: todayISO(), desc:'Dinner', amount: 120, paidBy:'Alice',
      splitType:'equal' },
    { id: uid(), date: todayISO(), desc:'Taxi', amount: 30, paidBy:'Bob',
      splitType:'custom', shares:[10,10,10] },
    { id: uid(), date: todayISO(), desc:'Groceries', amount: 90, paidBy:'Charlie',
      splitType:'equal' }
  ];
  save();
  renderAll();
}

```

```

/* ----- wiring ----- */
document.addEventListener('DOMContentLoaded', ()=>{
load();
  if(!state.members.length) { document.getElementById('exp-date').value =
    todayISO(); }
  else { document.getElementById('exp-date').value = todayISO();
    renderMembers(); populatePaidBySelect(); }
}

```

```

// events

```

```

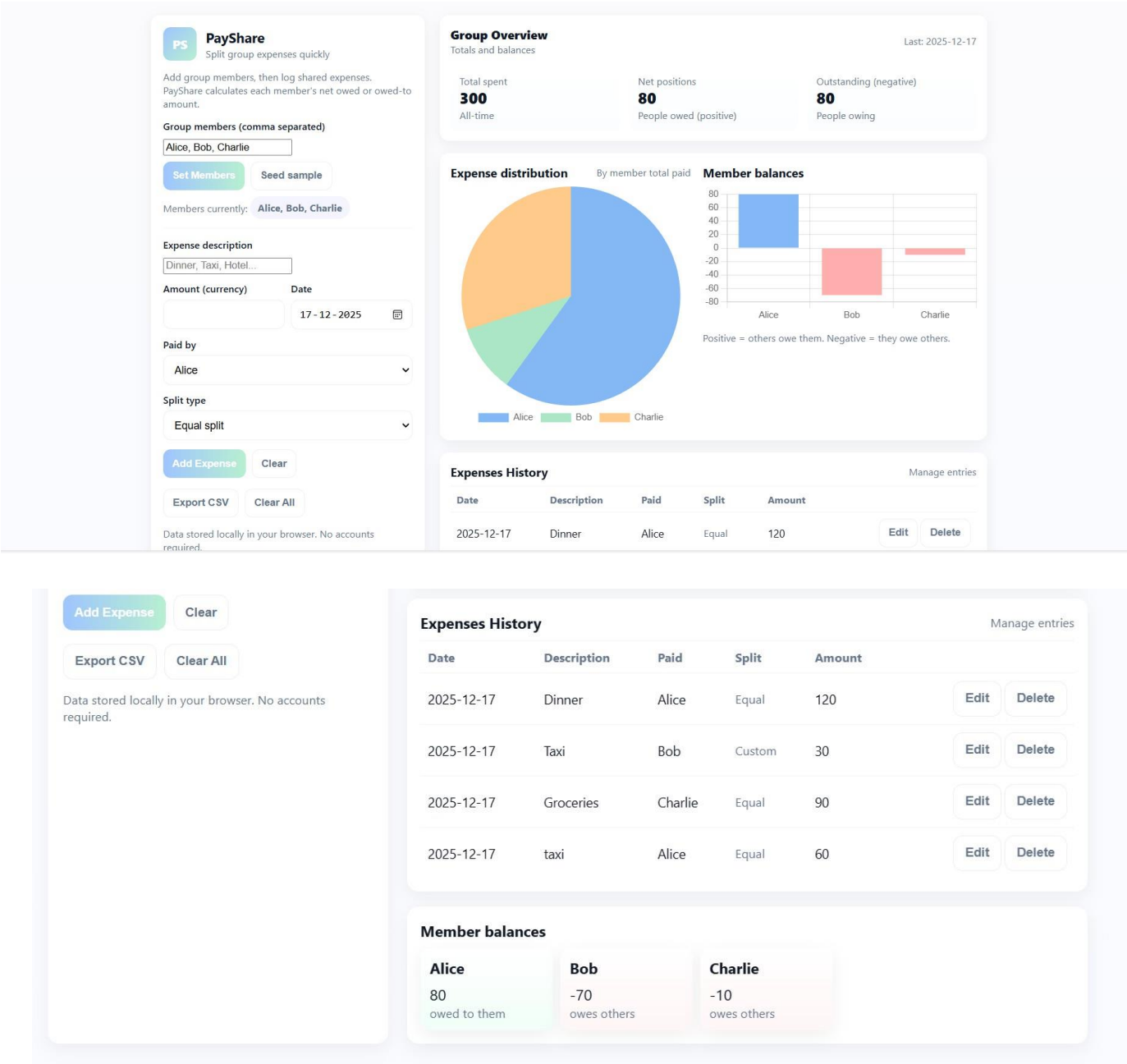
document.getElementById('set-members').addEventListener('click',
  setMembersFromInput);
document.getElementById('add-expense').addEventListener('click',
  addExpenseFromForm);
document.getElementById('clear-expense').addEventListener('click',
  clearExpenseForm);
document.getElementById('split-type').addEventListener('change', (e)=>{
  document.getElementById('custom-shares-area').style.display =
  e.target.value === 'custom' ? 'block' : 'none';
});
document.getElementById('export-csv').addEventListener('click',
  exportCSV);
document.getElementById('clear-all').addEventListener('click', clearAll);
document.getElementById('seed-sample').addEventListener('click',
  seedSample);
document.getElementById('custom-shares').addEventListener('input', ()=>{/*
  no-op */});
// render initial
renderAll();
});

/* expose for inline calls */
window.editExpense = editExpense;
window.deleteExpense = deleteExpense;
window.seedSample = seedSample;

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

```

OUTPUT:



CHAPTER 4

RESULTS AND DISCUSSIONS

Testing results confirmed that PayShare accurately calculates shared expenses and individual balances in real time. The Expense Manager allowed seamless addition and removal of expenses, while the Splitter ensured fair distribution among participants.

The balance summary clearly displayed who owed money and who was to receive payments, minimizing ambiguity. Responsiveness testing verified compatibility across devices, and users reported improved clarity and ease in managing group expenses.

LocalStorage successfully preserved data between sessions, eliminating the need for backend systems. Overall, PayShare achieved efficient and transparent shared financial management.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

PayShare demonstrates the effective use of front-end web technologies in developing a financial application for shared expense management. By integrating expense tracking, automated payment splitting, and balance visualization, the platform simplifies group financial coordination. Its intuitive design, real-time calculations, and persistent storage make it a practical tool for everyday shared spending scenarios.

5.2 Future Scope

PayShare can be enhanced with:

- Cloud-based multi-user synchronization
- User authentication and profiles
- Expense categorization and analytics
- Settlement reminders and notifications
- Multi-currency support
- Integration with digital payment platforms

With these enhancements, PayShare can evolve into a comprehensive group finance management solution suitable for diverse real-world applications.