

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>USD !” INR Cash Book</title>
<style>
* {
margin: 0;
padding: 0;
box-sizing: border-box;
}

body {
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
min-height: 100vh;
padding: 20px;
}

.container {
max-width: 900px;
margin: 0 auto;
}

.header {
text-align: center;
background: white;
padding: 30px;
border-radius: 20px;
box-shadow: 0 10px 30px rgba(0,0,0,0.1);
margin-bottom: 30px;
}

.header h1 {
color: #333;
font-size: 2.5rem;
margin-bottom: 10px;
}

.header p {
color: #666;
```

```
font-size: 1.2rem;  
}
```

```
.currency-selector {  
margin-top: 20px;  
display: flex;  
justify-content: center;  
align-items: center;  
gap: 15px;  
flex-wrap: wrap;  
}
```

```
.currency-selector label {  
color: #333;  
font-weight: 600;  
}
```

```
.currency-selector select {  
padding: 10px 15px;  
border: 2px solid #e2e8f0;  
border-radius: 8px;  
font-size: 1rem;  
background: white;  
cursor: pointer;  
}
```

```
.exchange-info {  
background: #f7fafc;  
padding: 15px;  
border-radius: 10px;  
margin: 15px 0;  
text-align: center;  
color: #4a5568;  
font-size: 0.9rem;  
}
```

```
.balance-section {  
background: white;  
padding: 30px;  
border-radius: 20px;  
box-shadow: 0 10px 30px rgba(0,0,0,0.1);  
margin-bottom: 30px;
```

```
text-align: center;
}
```

```
.balance-title {
color: #333;
font-size: 1.3rem;
margin-bottom: 15px;
}
```

```
.current-balance {
font-size: 3rem;
font-weight: bold;
color: #667eea;
margin: 20px 0;
}
```

```
.balance-summary {
display: flex;
justify-content: space-around;
margin-top: 25px;
}
```

```
.summary-item {
text-align: center;
}
```

```
.summary-item h4 {
color: #666;
margin-bottom: 10px;
font-size: 1.1rem;
}
```

```
.cash-in-amount {
color: #48bb78;
font-size: 1.5rem;
font-weight: bold;
}
```

```
.cash-out-amount {
color: #f56565;
font-size: 1.5rem;
font-weight: bold;
}
```

```
}
```

```
.forms-section {  
display: grid;  
grid-template-columns: 1fr 1fr;  
gap: 30px;  
margin-bottom: 30px;  
}
```

```
.form-card {  
background: white;  
padding: 30px;  
border-radius: 20px;  
box-shadow: 0 10px 30px rgba(0,0,0,0.1);  
}
```

```
.form-card h3 {  
color: #333;  
margin-bottom: 25px;  
font-size: 1.5rem;  
text-align: center;  
}
```

```
.input-group {  
margin-bottom: 20px;  
}
```

```
.input-group label {  
display: block;  
margin-bottom: 8px;  
color: #333;  
font-weight: 600;  
font-size: 1.1rem;  
}
```

```
.input-group input,  
.input-group textarea {  
width: 100%;  
padding: 15px;  
border: 2px solid #e2e8f0;  
border-radius: 12px;  
font-size: 1.1rem;
```

```
transition: all 0.3s ease;
}
```

```
.input-group input:focus,
.input-group textarea:focus {
outline: none;
border-color: #667eea;
box-shadow: 0 0 0 3px rgba(102, 126, 234, 0.1);
}
```

```
.input-group textarea {
resize: vertical;
min-height: 80px;
}
```

```
.submit-btn {
width: 100%;
padding: 15px;
border: none;
border-radius: 12px;
font-size: 1.1rem;
font-weight: bold;
cursor: pointer;
transition: all 0.3s ease;
text-transform: uppercase;
letter-spacing: 1px;
}
```

```
.cash-in-btn {
background: linear-gradient(135deg, #48bb78, #38a169);
color: white;
}
```

```
.cash-in-btn:hover {
background: linear-gradient(135deg, #38a169, #2f855a);
transform: translateY(-2px);
box-shadow: 0 10px 25px rgba(72, 187, 120, 0.3);
}
```

```
.cash-out-btn {
background: linear-gradient(135deg, #f56565, #e53e3e);
color: white;
```

```
}
```

```
.cash-out-btn:hover {  
background: linear-gradient(135deg, #e53e3e, #c53030);  
transform: translateY(-2px);  
box-shadow: 0 10px 25px rgba(245, 101, 101, 0.3);  
}
```

```
.transactions-section {  
background: white;  
padding: 30px;  
border-radius: 20px;  
box-shadow: 0 10px 30px rgba(0,0,0,0.1);  
}
```

```
.transactions-title {  
color: #333;  
font-size: 1.5rem;  
margin-bottom: 25px;  
text-align: center;  
}
```

```
.transaction {  
display: flex;  
justify-content: space-between;  
align-items: center;  
padding: 20px;  
border-bottom: 1px solid #f0f0f0;  
transition: all 0.3s ease;  
}
```

```
.transaction:hover {  
background-color: #f8f9ff;  
transform: translateX(5px);  
}
```

```
.transaction:last-child {  
border-bottom: none;  
}
```

```
.transaction-details {  
flex: 1;
```

```
}
```

```
.transaction-desc {  
font-weight: bold;  
color: #333;  
font-size: 1.1rem;  
margin-bottom: 5px;  
}
```

```
.transaction-time {  
color: #666;  
font-size: 0.95rem;  
}
```

```
.transaction-right {  
display: flex;  
align-items: center;  
gap: 15px;  
}
```

```
.transaction-value {  
font-weight: bold;  
font-size: 1.3rem;  
padding: 8px 16px;  
border-radius: 25px;  
}
```

```
.cash-in-value {  
background: #e6fffa;  
color: #48bb78;  
}
```

```
.cash-out-value {  
background: #fed7d7;  
color: #f56565;  
}
```

```
.delete-button {  
background: #ff9800;  
color: white;  
border: none;  
padding: 8px 15px;
```

```
border-radius: 8px;
cursor: pointer;
font-size: 0.9rem;
transition: all 0.3s ease;
}
```

```
.delete-button:hover {
background: #f57c00;
transform: translateY(-1px);
}
```

```
.no-transactions {
text-align: center;
color: #666;
font-style: italic;
padding: 50px 20px;
font-size: 1.1rem;
}
```

```
@media (max-width: 768px) {
.forms-section {
grid-template-columns: 1fr;
}
```

```
.balance-summary {
flex-direction: column;
gap: 20px;
}
```

```
.header h1 {
font-size: 2rem;
}
```

```
.current-balance {
font-size: 2.5rem;
}
```

```
.transaction {
flex-direction: column;
align-items: flex-start;
gap: 15px;
}
```



```

.transaction-right {
align-self: stretch;
justify-content: space-between;
}
}
</style>
</head>
<body>
<div class="container">
<!-- Header -->
<div class="header">
< h 1 >USD !" INR Cash Book</h1>
<p>Track your US Dollar and Indian Rupee transactions with live exchange rates</p>
<div class="currency-selector">
<label for="baseCurrency">Base Currency:</label>
<select id="baseCurrency" onchange="updateCurrency()">
<option value="USD">USD ($) - US Dollar</option>
<option value="INR">INR ( ₹ ) Indian Rupee</option>
</select>
</div>
<div class="exchange-info" id="exchangeInfo">
USD !" INR exchange rate updated every 30 seconds. Current rate loading...
</div>
</div>

<!-- Balance Section -->
<div class="balance-section">
<h3 class="balance-title">Current Balance</h3>
<div class="current-balance" id="currentBalance">$0.00</div>
<div class="balance-summary">
<div class="summary-item">
<h4>Total Cash In</h4>
<div class="cash-in-amount" id="totalCashIn">$0.00</div>
</div>
<div class="summary-item">
<h4>Total Cash Out</h4>
<div class="cash-out-amount" id="totalCashOut">$0.00</div>
</div>
</div>
</div>

```

```

<!-- Forms Section -->
<div class="forms-section">
<!-- Cash In Form -->
<div class="form-card">
  < h 3 >Add Cash In</h3>
  <div class="input-group">
    <label>Amount</label>
    <div style="display: flex; gap: 10px;">
      <input type="number" id="cashInAmount" step="0.01" min="0.01" placeholder="Enter amount"
        style="flex: 1;">
      <select id="cashInCurrency" style="width: 120px; padding: 15px; border: 2px solid #e2e8f0;
        border-radius: 12px;">
        <option value="USD">USD ($)</option>
        <option value="INR">INR ( ₹ ) < / o p t i o n >
      </select>
    </div>
    <div id="cashInConverted" style="margin-top: 8px; color: #666; font-size: 0.9rem;"></div>
  </div>
  <div class="input-group">
    <label>Description</label>
    <textarea id="cashInDescription" placeholder="What is this money for?"></textarea>
  </div>
  <button class="submit-btn cash-in-btn" onclick="addCashIn()">Add Cash In</button>
</div>

<!-- Cash Out Form -->
<div class="form-card">
  < h 3 >Add Cash Out</h3>
  <div class="input-group">
    <label>Amount</label>
    <div style="display: flex; gap: 10px;">
      <input type="number" id="cashOutAmount" step="0.01" min="0.01" placeholder="Enter
        amount" style="flex: 1;">
      <select id="cashOutCurrency" style="width: 120px; padding: 15px; border: 2px solid #e2e8f0;
        border-radius: 12px;">
        <option value="USD">USD ($)</option>
        <option value="INR">INR ( ₹ ) < / o p t i o n >
      </select>
    </div>
    <div id="cashOutConverted" style="margin-top: 8px; color: #666; font-size: 0.9rem;"></div>
  </div>
  <div class="input-group">

```

```
<label>Description</label>
<textarea id="cashOutDescription" placeholder="What did you spend on?"></textarea>
</div>
<button class="submit-btn cash-out-btn" onclick="addCashOut()">Add Cash Out</button>
</div>
</div>
```

```
<!-- Transactions Section -->
<div class="transactions-section">
  <h3 class="transaction-section">Transaction History</h3>
  <div id="transactionsList">
    <div class="no-transactions">No transactions yet. Add your first transaction above!</div>
  </div>
</div>
</div>
```

```
<script>
// Initialize data
let transactions = [];
let exchangeRates = {
  USD: 1,
  INR: 83.25 // Current approximate USD to INR rate
};
let baseCurrency = 'USD';
```

```
// Currency symbols
const currencySymbols = {
  USD: '$',
  INR: '₹'
};
```

```
// Currency names
const currencyNames = {
  USD: 'US Dollar',
  INR: 'Indian Rupee'
};
```

```
// Load data when page starts
function loadData() {
  const saved = localStorage.getItem('usdinrcashbook');
  if (saved) {
    const data = JSON.parse(saved);
```

```

transactions = data.transactions || [];
baseCurrency = data.baseCurrency || 'USD';
document.getElementById('baseCurrency').value = baseCurrency;
}
updateExchangeRates();
updateDisplay();
}

// Save data to browser storage
function saveData() {
const data = {
transactions: transactions,
baseCurrency: baseCurrency,
lastUpdated: new Date().toISOString()
};
localStorage.setItem('usdinrcashbook', JSON.stringify(data));
}

// Update exchange rates with realistic USD-INR fluctuation
function updateExchangeRates() {
// Simulate realistic INR fluctuation (typically ±0.5% daily movement)
const baseRate = 83.25;
const fluctuation = (Math.random() - 0.5) * 1.0; // ±0.5 rupees
exchangeRates.INR = Math.max(80, Math.min(86, baseRate + fluctuation)); // Keep within realistic range

const info = document.getElementById('exchangeInfo');
const rate = exchangeRates.INR.toFixed(2);
info.innerHTML = `
<strong>Current Rate:</strong> 1 USD = 1 $ { rate } ₹ INR =
${(1/exchangeRates.INR).toFixed(4)}
<br><small>Last updated: ${new Date().toLocaleTimeString()}</small>
`;
}

// Convert currency
function convertCurrency(amount, fromCurrency, toCurrency) {
if (fromCurrency === toCurrency) return amount;

if (fromCurrency === 'USD' && toCurrency === 'INR') {
return amount * exchangeRates.INR;
} else if (fromCurrency === 'INR' && toCurrency === 'USD') {

```

```

return amount / exchangeRates.INR;
}
return amount;
}

```

// Format currency display

```

function formatCurrency(amount, currency) {
const symbol = currencySymbols[currency];
if (currency === 'INR') {
// Format INR with Indian number system (lakhs, crores)
if (amount >= 10000000) { // 1 crore
return `${symbol}${(amount/10000000).toFixed(2)}Cr`;
} else if (amount >= 100000) { // 1 lakh
return `${symbol}${(amount/100000).toFixed(2)}L`;
} else {
return `${symbol}${amount.toFixed(2)}`;
}
} else {
return `${symbol}${amount.toLocaleString('en-US', {minimumFractionDigits: 2,
maximumFractionDigits: 2})}`;
}
}

```

// Update currency display

```

function updateCurrency() {
baseCurrency = document.getElementById('baseCurrency').value;
saveData();
updateDisplay();
updateConversionDisplays();
}

```

// Update conversion displays with enhanced formatting

```

function updateConversionDisplays() {
// Cash In conversion
const cashInAmount = parseFloat(document.getElementById('cashInAmount').value)|| 0;
const cashInCurrency = document.getElementById('cashInCurrency').value;
if (cashInAmount > 0 && cashInCurrency !== baseCurrency) {
const converted = convertCurrency(cashInAmount, cashInCurrency, baseCurrency);
const rate = cashInCurrency === 'USD' ? exchangeRates.INR : (1/exchangeRates.INR);
document.getElementById('cashInConverted').innerHTML =
`<strong>${formatCurrency(converted, baseCurrency)}</strong> in ${baseCurrency}<br>
<small style="opacity: 0.7;">Rate: 1 ${cashInCurrency} = ${formatCurrency(rate,

```

```

baseCurrency))</small>`;
} else {
document.getElementById('cashInConverted').textContent = "";
}

// Cash Out conversion
const cashOutAmount = parseFloat(document.getElementById('cashOutAmount').value) || 0;
const cashOutCurrency = document.getElementById('cashOutCurrency').value;
if (cashOutAmount > 0 && cashOutCurrency !== baseCurrency) {
const converted = convertCurrency(cashOutAmount, cashOutCurrency, baseCurrency);
const rate = cashOutCurrency === 'USD' ? exchangeRates.INR : (1/exchangeRates.INR);
document.getElementById('cashOutConverted').innerHTML =
`<strong>${formatCurrency(converted, baseCurrency)}</strong> in ${baseCurrency}<br>
<small style="opacity: 0.7;">Rate: 1 ${cashOutCurrency} = ${formatCurrency(rate,
baseCurrency)}</small>`;
} else {
document.getElementById('cashOutConverted').textContent = "";
}
}

// Add event listeners for real-time conversion
function setupConversionListeners() {
document.getElementById('cashInAmount').addEventListener('input',
updateConversionDisplays);
document.getElementById('cashInCurrency').addEventListener('change',
updateConversionDisplays);
document.getElementById('cashOutAmount').addEventListener('input',
updateConversionDisplays);
document.getElementById('cashOutCurrency').addEventListener('change',
updateConversionDisplays);
}

// Add cash in
function addCashIn() {
const amount = document.getElementById('cashInAmount').value;
const currency = document.getElementById('cashInCurrency').value;
const description = document.getElementById('cashInDescription').value;

if (!amount || !description) {
alert('Please fill in both amount and description');
return;
}

```

```
if (parseFloat(amount) <= 0) {  
  alert('Please enter a valid amount greater than 0');  
  return;  
}  
  
// Convert to base currency for storage  
const baseAmount = convertCurrency(parseFloat(amount), currency, baseCurrency);  
  
const transaction = {  
  id: Date.now(),  
  type: 'in',  
  amount: parseFloat(amount),  
  currency: currency,  
  baseAmount: baseAmount,  
  baseCurrency: baseCurrency,  
  description: description.trim(),  
  date: new Date()  
};  
  
transactions.push(transaction);  
  
// Clear inputs  
document.getElementById('cashInAmount').value = '';  
document.getElementById('cashInDescription').value = '';  
document.getElementById('cashInConverted').textContent = '';  
  
saveData();  
updateDisplay();
```