

OUTPUT:

Digital Wallet & Subscription Tracker

My Wallet

Current Balance

₹ 538

Withdraw

Add Money

Subscription Tracker

Your subscriptions

Add Subscription

Netflix

Standard

₹ 500 / Monthly

EntertainmentDebit Card

Spotify

Pro

₹ 3000 / Yearly

MusicWallet

Amazon

Pro+

₹ 300 / Weekly

Amazon PrimeUPI

Digital Wallet & Subscription Tracker

My Wallet

Current Balance

₹ 538

Withdraw

Add Money

Name

Enter Name

Subscription Plan

Billing Cycle

Amount

Enter Amount

Category

Payment Method

Add Subscription

Add Subscription

Amazon

Pro+

₹ 300 / Weekly

Amazon PrimeUPI

## Local storage:

http://localhost:4200	
Origin	http://localhost:4200
Key	http://localhost:4200
subscriptions	[{"name":"Netflix","subplan":"Standard","billcycle":"Monthly","su...]
▼ [{"name": "Netflix", subplan: "Standard", billcycle: "Monthly", subamount: "500",...},...] ▶ 0: {name: "Netflix", subplan: "Standard", billcycle: "Monthly", subamount: "500",...} ▶ 1: {name: "Spotify", subplan: "Pro", billcycle: "Yearly", subamount: "3000", category: "Music",...} ▶ 2: {name: "Amazon", subplan: "Pro+", billcycle: "Weekly", subamount: "300", category: "Amazon Prime",...}	

## CODE:

### Index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <meta charset="utf-8">
```

```
  <title>AssessmentApp</title>
```

```
  <meta name="description" content="">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
  {{content-for "head"}}
```

```
  <link integrity="" rel="stylesheet" href="{{rootURL}}assets/vendor.css">
```

```
  <link integrity="" rel="stylesheet" href="{{rootURL}}assets/assessment-app.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=Poppins: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&display=swap"
```

```
  rel="stylesheet">
```

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">
```

```
{{content-for "head-footer"}}
```

```
</head>
```

```
<body>
```

```
  {{content-for "body"}}
```

```
<script src="{{rootURL}}assets/vendor.js"></script>
```

```
<script src="{{rootURL}}assets/assessment-app.js"></script>
```

```
  {{content-for "body-footer"}}
```

```
</body>
```

```
</html>
```

## **Templates/application.hbs**

```
{{page-title "AssessmentApp"}}
```

```
{{outlet}}
```

```
<h3>Digital Wallet & Subscription Tracker</h3>
```

```

<div class="container">
  <div class="my-wallet">
    <h2>My Wallet</h2>
    <p class="amount-title">Current Balance</p>
    <span class="amount"><i class="bi bi-currency-rupee"></i> {{this.amount}}</span>
    <div class="wallet-btn-container">
      <button class="wallet-btn withdraw" type="button">Withdraw</button>
      <button class="wallet-btn add-money" type="button" {{on "click"
this.toggleModal}}>Add Money</button>
    </div>

    {{#if this.isModelOpen}}
    <div class="overlay"></div>
    <div class="add-modal">
      <i class="bi bi-x-lg close-icon" {{on "click" this.toggleModal}}></i>
      <label for="amount-value">Amount</label>
      <input type="number" placeholder="Enter Amount" id="amount-value">
      <button class="wallet-btn" type="button" {{on "click" this.addToWallet}}>Add to
Wallet</button>
    </div>

    {{/if}}
  </div>

  <div class="subscription-tracker">
    <div class="subscription-tracker-header">
      <div class="subscription-tracker-head">
        <span class="heading">Subscription Tracker</span>

```

```

        <span class="description">Your subscriptions</span>

    </div>

    <button class="add-sub-btn" type="button" {{on "click" this.toggleSubModal}}>Add
Subscription</button>

</div>

<div class="subscriptions">

    {{#each this.subscriptions as |subscription|}}

        <div class="subscription-card">

            <span class="name">{{subscription.name}}</span>

            <span class="sub-plan">{{subscription.subplan}}</span>

            <div><span class="sub-amount"><i class="bi bi-currency-rupee"></i>
{{subscription.subamount}}</span>/

            <span class="billing-cycle">{{subscription.billcycle}}</span></div>

            <div><span class="category">{{subscription.category}}</span>

            <span class="payment-method">{{subscription.paymethod}}</span></div>

        </div>

    {{/each}}

</div>

{{#if this.isSubModelOpen}}

<div class="overlay"></div>

<div class="add-sub-modal">

    <i class="bi bi-x-lg close-icon" {{on "click" this.toggleSubModal}}></i>

    <label for="amount-value">Name</label>

    <input type="text" placeholder="Enter Name" id="name" {{on "input"
this.updateName}}>

```

```

        <PowerSelect @selected={{this.subplan}} @options={{this.plans}}
@onChange={{this.updatePlan}}

        @labelText="Subscription Plan" as |plan|>

        {{plan}}

    </PowerSelect>

    <PowerSelect @selected={{this.billcycle}} @options={{this.billings}}
@onChange={{this.updateBilling}}

        @labelText="Billing Cycle" as |billing|>

        {{billing}}

    </PowerSelect>

    <label for="sub-amount">Amount</label>

    <input type="number" placeholder="Enter Amount" id="sub-amount" {{on "input"
this.updateAmount}}>

    <PowerSelect @selected={{this.category}} @options={{this.categories}}
@onChange={{this.updateCategory}}

        @labelText="Category" as |category|>

        {{category}}

    </PowerSelect>

    <PowerSelect @selected={{this.paymethod}} @options={{this.paymentMethod}}
@onChange={{this.updatePayment}}

        @labelText="Payment Method" as |pay|>

        {{pay}}

    </PowerSelect>

    <button class="wallet-btn" type="button" {{on "click" this.addSubscription}}>Add
Subscription</button>

    </div>

    {{/if}}

</div>

</div>

<BasicDropdownWormhole />

```

## Controllers/application.js

```
import Controller from '@ember/controller';
import { tracked } from '@glimmer/tracking';
import { action } from '@ember/object'

export default class ApplicationController extends Controller {
  constructor() {
    super(...arguments)

    this.loadAmount();

    this.loadSubscription();
  }

  @tracked amount=0;

  @tracked isModelOpen = false;

  @tracked isSubModelOpen = false;

  @tracked subscriptions = [];

  @tracked name = "";
  @tracked subplan="";
  @tracked billcycle="";
  @tracked subamount="";
  @tracked category="";
  @tracked paymethod="";

  plans=['Standard', 'Pro', 'Pro+'];
  billings=['Weekly', 'Monthly', 'Yearly'];
  categories=['Entertainment', 'Music', 'Amazon Prime', 'Jio Star'];
```

```
paymentMethod = ['UPI', 'Debit Card', 'Net Banking', 'Wallet']
```

```
@action toggleModal() {  
  this.isModelOpen = !this.isModelOpen;  
}
```

```
@action toggleSubModal() {  
  this.isSubModelOpen = !this.isSubModelOpen;  
}
```

```
@action addToWallet() {  
  const value = document.getElementById('amount-value').value;  
  this.amount=parseInt(this.amount)+ parseInt(value);  
  localStorage.setItem("amount",this.amount);  
  this.toggleModal();  
}
```

```
@action loadAmount() {  
  const resAmount= localStorage.getItem("amount");  
  this.amount=resAmount;  
}
```

```
@action updateName(event) {  
  this.name= event.target.value;  
}
```

```
@action updatePlan(plan) {  
  this.subplan=plan;
```



```
}
```

```
@action updateBilling(bill) {
```

```
    this.billcycle=bill;
```

```
}
```

```
@action updateAmount(event) {
```

```
    this.subamount=event.target.value;
```

```
}
```

```
@action updateCategory(cat) {
```

```
    this.category=cat;
```

```
}
```

```
@action updatePayment(pay) {
```

```
    this.paymethod=pay;
```

```
}
```

```
@action addSubscription() {
```

```
    if(this.subscriptions!=null){
```

```
        this.subscriptions=[...this.subscriptions,
```

```
        {
```

```
            name:this.name,
```

```
            subplan:this.subplan,
```

```
            billcycle:this.billcycle,
```

```
            subamount:this.subamount,
```

```
            category:this.category,
```

```
            paymethod:this.paymethod
```

```

        }
    ];
    localStorage.setItem("subscriptions", JSON.stringify(this.subscriptions));
    this.loadSubscription();
    this.toggleSubModal();
} else {
    this.subscriptions= [
        {
            name:this.name,
            subplan:this.subplan,
            billcycle:this.billcycle,
            subamount:this.subamount,
            category:this.category,
            paymethod:this.paymethod
        }
    ]
    localStorage.setItem("subscriptions", JSON.stringify(this.subscriptions))
    this.loadSubscription();
    this.toggleSubModal();
}
}

@action loadSubscription() {
    const resSubscription= JSON.parse(localStorage.getItem("subscriptions"));
    this.subscriptions=resSubscription;
    console.log(this.subscriptions)
}
}

```

## Styles/app.css

/\* Ember supports plain CSS out of the box. More info:

<https://cli.emberjs.com/release/advanced-use/stylesheets/> \*/

```
body {
```

```
  font-family: Poppins;
```

```
}
```

```
.overlay {
```

```
  width: 100%;
```

```
  height: 100%;
```

```
  background-color: #2a2a2a;
```

```
  opacity: 0.7;
```

```
  position: fixed;
```

```
  top: 0;
```

```
  left: 0;
```

```
}
```

```
.container {
```

```
  display: flex;
```

```
  gap: 2rem;
```

```
  padding: 1rem;
```

```
}
```

```
.my-wallet {
```

```
  background: linear-gradient(90deg, rgb(42 123 155 / 100%) 0%, rgb(87 199 133 / 100%) 50%, rgb(237 221 83 / 100%) 100%);
```

```
  color: #fff;
```

```
border-radius: 10px;
padding: 2rem;
width: 500px;
height: 350px;
}
```

```
.amount-title {
  color: #3e5058;
}
```

```
.amount {
  font-size: 48px;
}
```

```
.wallet-btn-container {
  margin: 1rem;
  background-color: #fff;
  border-radius: 20px;
  padding: 2rem;
  display: flex;
  gap: 2rem;
  justify-content: center;
  align-items: center;
}
```

```
.wallet-btn {
  border: none;
  padding: 1rem 3rem;
```

```
border-radius: 10px;
background-color: #2a2a2a;
color: #fff;
cursor: pointer;
}
```

```
.add-modal {
  background-color: #fff;
  color: black;
  border-radius: 10px;
  padding: 2rem 4rem;
  z-index: 5;
  position: absolute;
  top: 30%;
  left: 40%;
  display: flex;
  flex-direction: column;
  gap: 1rem;
}
```

```
input {
  padding: 0.5rem 1rem;
  border-radius: 5px;
  border: 1px solid #ccc;
  width: auto;
}
```

```
.close-icon {  
  position: absolute;  
  top: 1rem;  
  right: 1rem;  
  cursor: pointer;  
}
```

```
.subscription-tracker {  
  background-color: #fff;  
  box-shadow: 1px 2px 4px rgb(0 0 0 / 20%);  
  padding: 2rem;  
  display: flex;  
  flex-direction: column;  
  border: 1px solid #ddd;  
}
```

```
.add-sub-btn {  
  border: none;  
  padding: 0.6rem 1rem;  
  background-color: #2a2a2a;  
  color: white;  
  border-radius: 5px;  
  cursor: pointer;  
}
```

```
.subscription-tracker-header {  
  display: flex;  
  align-items: start;
```

```
    justify-content: space-between;
    gap: 2rem;
}
```

```
.subscription-tracker-head {
    display: flex;
    flex-direction: column;
}
```

```
.heading {
    font-size: 20px;
    font-weight: bold;
}
```

```
.subscriptions {
    display: flex;
    gap: 0.5rem;
    flex-wrap: wrap;
    width: 100%;
}
```

```
.subscription-card {
    margin: 1rem auto;
    padding: 1.5rem;
    width: 200px;
    border-radius: 20px;
    display: flex;
    flex-direction: column;
```

```
    justify-content: center;
    align-items: start;
    gap: 0.5rem;
    box-shadow: 1px 2px 4px rgba(0,0,0,0.3);
}
```

```
.add-sub-modal {
    background-color: #fff;
    color: black;
    border-radius: 10px;
    padding: 2rem;
    z-index: 5;
    position: absolute;
    top: 10%;
    display: flex;
    flex-direction: column;
    gap: 1rem;
    width: 400px;
}
```

```
.name {
    font-size: 24px;
}
```

```
.sub-plan {
    font-size: 14px;
    color: #969696;
}
```



```
.sub-amount {  
  font-size: 30px;  
}
```

```
.category, .payment-method {  
  border-radius: 20px;  
  padding: 0.3rem 0.6rem;  
  font-size: 12px;  
}
```

```
.category {  
  background-color: #ccc;  
  
}
```

```
.payment-method {  
  background-color: #4cad6e;  
  color: white;  
}
```