页面重构：

关于我们页面：



把关于我们页面的bigbanner 部分改为一张一张的轮播图，使用了JavaScript 计时器 setInterval() 实现每五秒自动切换

<div class="bigbanner">

<ul>

<li><img src="../images/小红书0.jpg" alt=""></li>

<li><img src="../images/小红书1.jpg" alt=""></li>

<li><img src="../images/小红书2.jpg" alt=""></li>

<li><img src="../images/小红书3.jpg" alt=""></li>

<li><img src="../images/小红书4.jpg" alt=""></li>

</ul>

</div>

<style>

/\* 轮播图样式 \*/

.bigbanner {

position: relative;

overflow: hidden;

width: 100%;

height: 500px; /\* 根据需要设置高度 \*/

}

.bigbanner ul {

position: relative;

width: 100%;

height: 100%;

display: flex;

transition: transform 0.5s ease;

}

.bigbanner li {

min-width: 100%;

list-style: none;

}

</style>

<script>

$(document).ready(function() {

let currentIndex = 0; // 初始显示的图片索引

const slides = $(".bigbanner ul li"); // 获取所有图片项

const slideCount = slides.length; // 获取图片总数

// 每五秒切换一次图片

setInterval(function() {

currentIndex = (currentIndex + 1) % slideCount; // 更新图片索引

$(".bigbanner ul").css("transform", `translateX(-${currentIndex \* 100}%)`);

}, 5000); // 5000毫秒（5秒）

});

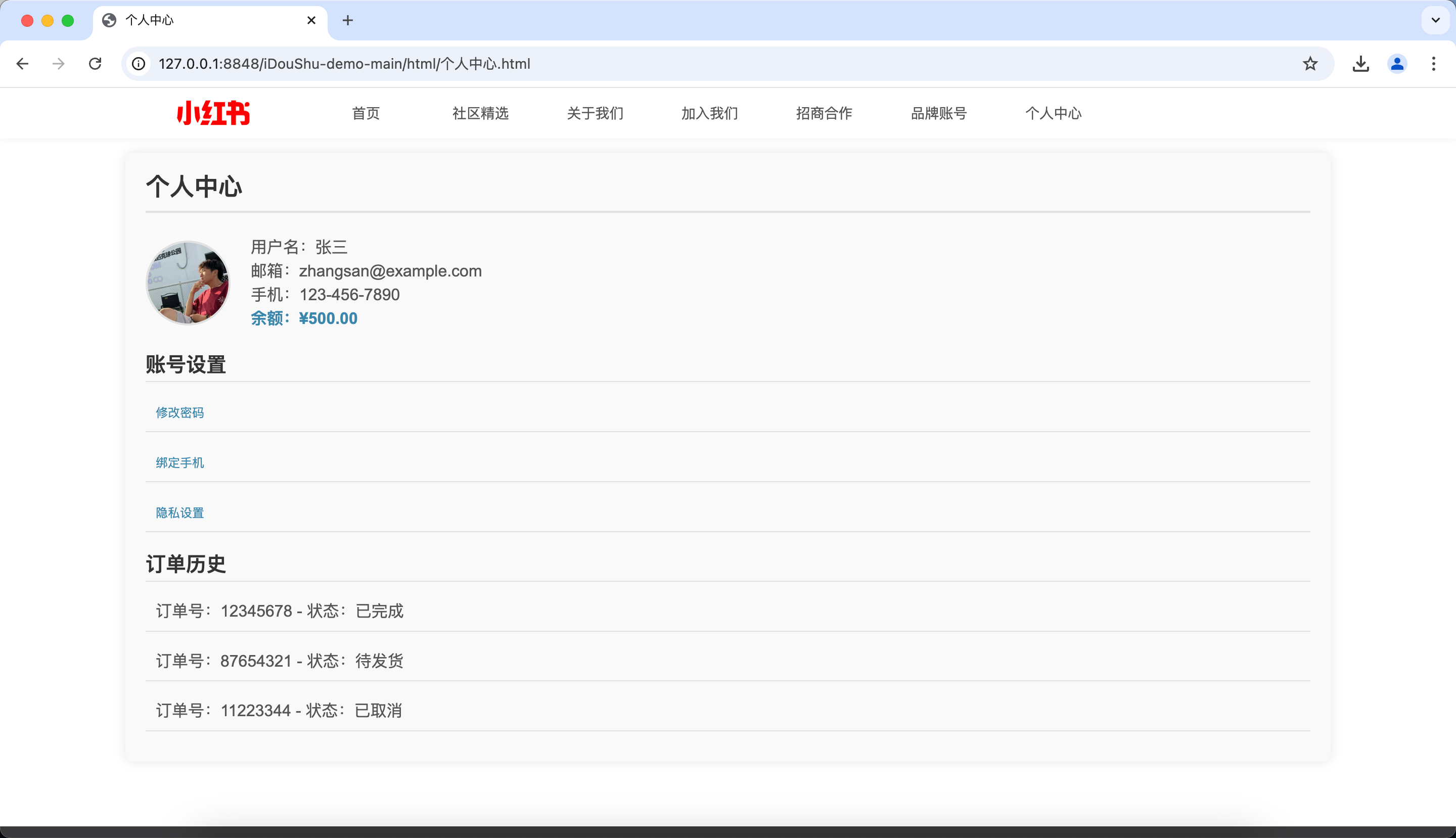
</script>

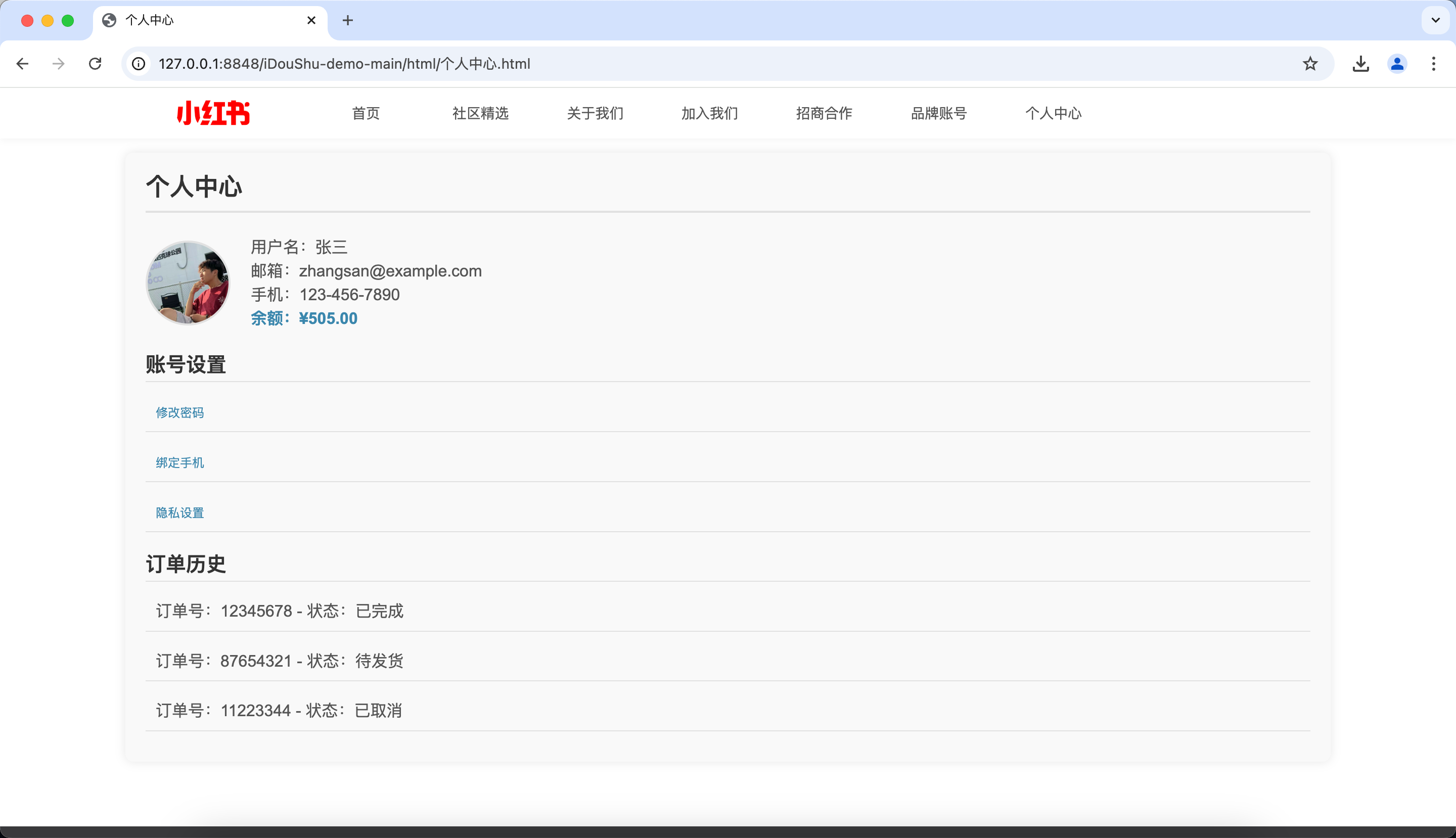
· **样式部分**：设置 bigbanner 的 overflow: hidden 确保只有一张图片可见，ul 设置 display: flex 实现图片横向布局。

· **JavaScript 部分**：通过 setInterval 计时器每五秒更新一次 currentIndex，利用 transform: translateX() 实现平滑切换效果。

新增页面：

个人中心页面：





余额可根据领取红包发生变化

这部分，通过在页面加载时，执行javascript代码，获取本地浏览器缓存的localStorage实现，判断缓存中是否有红包奖励，price参数，如果有，则将参数除以100作为每次动画的增量，使用计时器，完成动画，每次在余额字段后面，增加每次的变化增量，通过变化数值，完成数字跳动变化的效果，每20秒执行一次数值变化，直到增加完成，在增加完成之后，清除缓存localStorage里的price属性。防止红包重复多次叠加。

const balanceElement = document.querySelector('.balance'); // 声明一次

console.log("current",balanceElement.textContent.replace('余额：¥', ''))

let currentBalance = parseFloat(balanceElement.textContent.replace('余额：¥', ''));

console.log("currentBalance",currentBalance)

// 检查 localStorage 中的 price 值

const storedPrice = parseFloat(localStorage.getItem('price'));

if (!isNaN(storedPrice) && storedPrice > 0) {

let displayBalance = currentBalance; // 用于显示累加动画的变量

const increment = storedPrice / 100; // 增量，用于控制动画效果

console.log("displayBalance",displayBalance)

console.log("displayBalance",increment)

const interval = setInterval(() => {

if (displayBalance >= currentBalance + storedPrice) {

clearInterval(interval); // 动画结束

balanceElement.textContent =

`余额：¥${(currentBalance + storedPrice).toFixed(2)}`; // 设置最终值

} else {

displayBalance += increment;

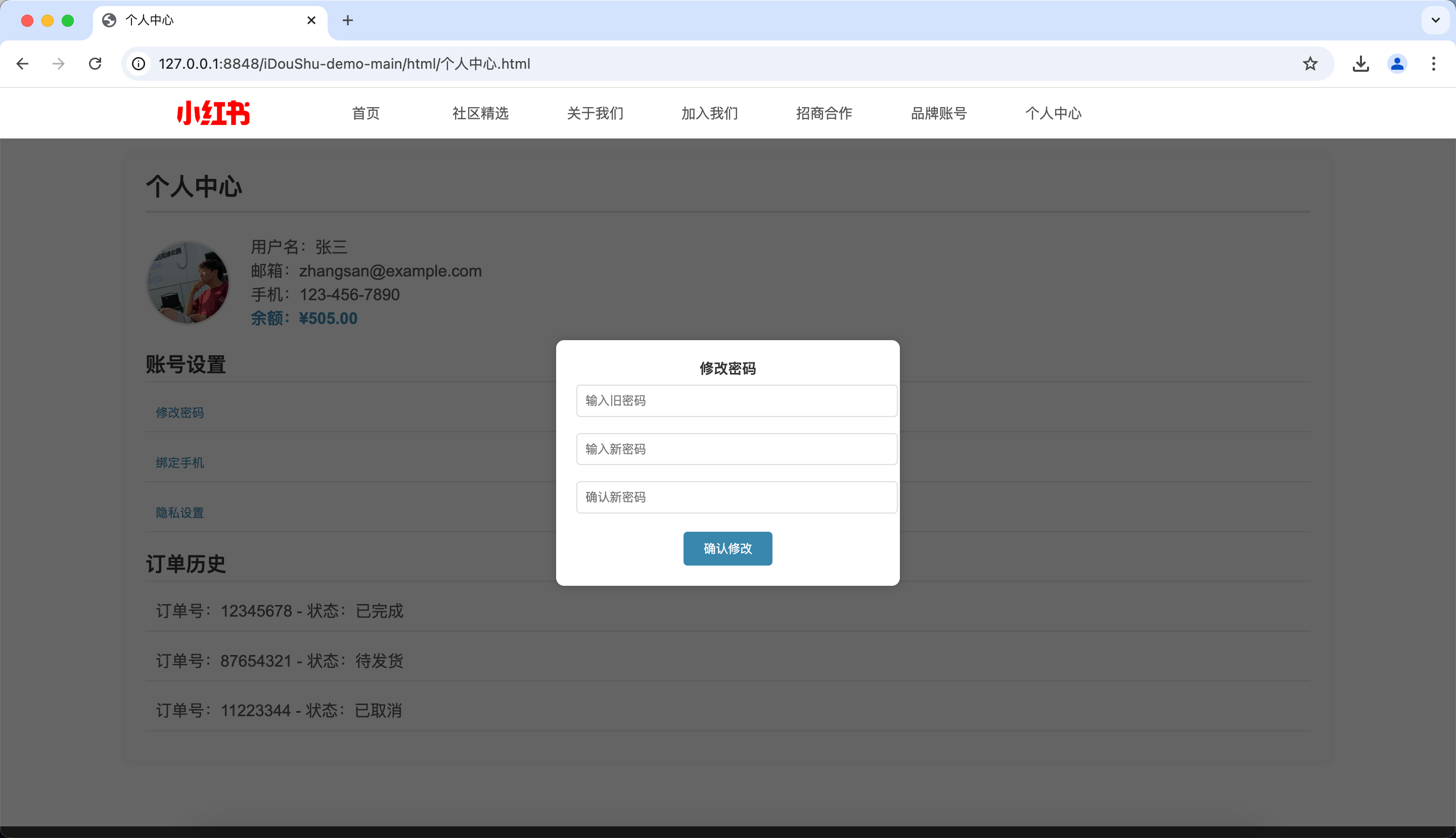
balanceElement.textContent = `余额：¥${displayBalance.toFixed(2)}`;

}

}, 20); // 控制动画速度

// 累加完成后清除 localStorage 中的 price

localStorage.removeItem('price');



点击修改密码会显示弹窗，可以在弹窗里修改新密码，

这部分，通过定义一个一开始隐藏的div元素，把修改密码的部分，放进隐藏区域内，对修改密码按钮 添加如下script代码，在点击时，把modal的隐藏去掉，使用display:flex属性，把元素变为可见，做成点击之后显示弹窗的效果

document.getElementById('changePasswordBtn').addEventListener('click', function() {

document.getElementById('modal').style.display = 'flex';

});

<!-- 弹窗结构 -->

<div id="modal" class="modal">

<div class="modal-content">

<h3>修改密码</h3>

<input type="password" placeholder="输入旧密码" id="oldPassword">

<input type="password" placeholder="输入新密码" id="newPassword">

<input type="password" placeholder="确认新密码" id="confirmPassword">

<button onclick="updatePassword()">确认修改</button>

<p class="success-message" id="successMessage">密码修改成功！</p>

</div>

</div>

.modal {

display: none;

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.6);

justify-content: center;

align-items: center;

animation: fadeIn 0.5s;

}

.modal-content {

background-color: #fff;

padding: 20px;

width: 300px;

border-radius: 8px;

text-align: center;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

animation: scaleUp 0.3s ease-out;

}

.modal-content h3 {

margin-top: 0;

}

.modal-content input {

width: 100%;

padding: 8px;

margin: 8px 0;

border-radius: 4px;

border: 1px solid #ddd;

}

.modal-content button {

padding: 10px 20px;

background-color: #3a87ad;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

margin-top: 10px;

}



文章页面加入有动画的红包图片，放置在页面左上角，通过加入图片元素和css，@keyframe完成有动画的img元素

<img class="small1" id="trigger-button" src="https://resource.tuixb.cn/test/mp\_51dcd190-834c-11e9-8444-018b79c0e985/default/f82aaf97-0fbc-4a7e-afc9-ad9a11277924.png" />

.small1{

position: fixed;

bottom: 80upx;

right: 10upx;

height: 118upx;

width: 120upx;

background: none;

animation:swing .5s linear infinite;

transform:rotate(0deg);

}

@keyframes swing{

0% {transform:rotate(0deg);}

10% {transform:rotate(-10deg);}

20% {transform:rotate(0deg);}

30% {transform:rotate(10deg);}

40% {transform:rotate(0deg);}

50% {transform:rotate(-10deg);}

60% {transform:rotate(0deg);}

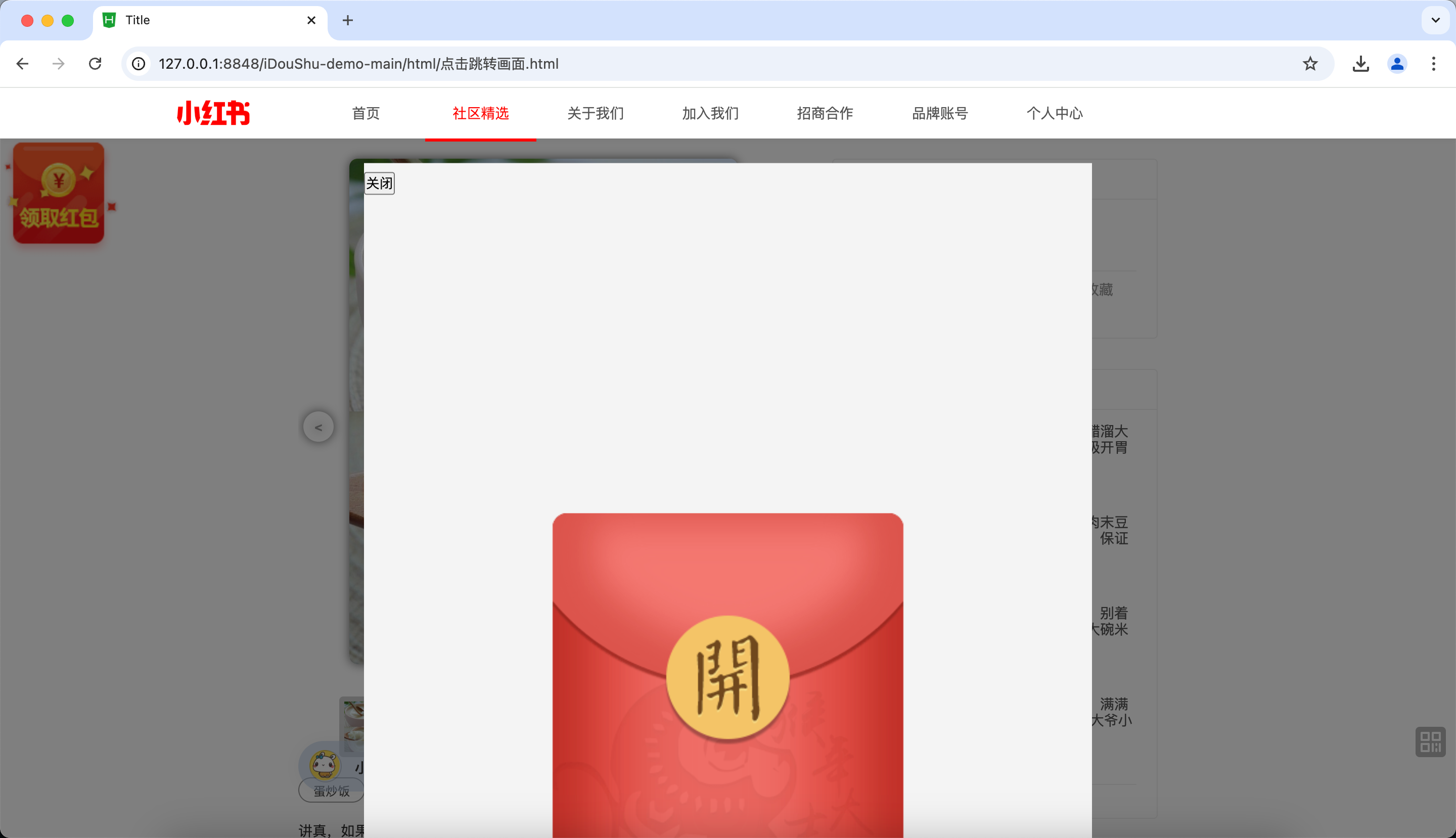
70% {transform:rotate(10deg);}

80% {transform:rotate(0deg);}

90% {transform:rotate(0deg);}

100% {transform:rotate(0deg);}

}



点击领取红包，弹出弹窗，可以实际领取红包，并在缓存中存入price余额，这部分，使用了一个新的思想，通过页面嵌套的方式，在另一个界面中，完成红包部分的渲染，在这个界面中通过iframe标签把页面引入进来，采取上面同样的思想，通过点击事件，控制iframe的显隐，在子界面中，完成红包状态的固定，即，打开就保持已开启的状态，通过对左上角小红包图片绑定点击事件，来完成模拟领取红包的过程

关键部分代码：

<!-- 居中显示的 iframe 容器 -->

<div id="iframe-container">

<iframe src="index.html" id="red-envelope-iframe"></iframe>

</div>

$(document).ready(function() {

// 为 header\_logo 中的图片绑定点击事件

$('.small1').on('click', function() {

// 显示遮罩层和 iframe 容器

$('#overlay').show();

$('#iframe-container').show();

// 发送消息触发动画

document.getElementById('red-envelope-iframe').contentWindow.postMessage('startAnimation', '\*');

});

// 监听 iframe 发来的关闭消息

window.addEventListener('message', function(event) {

if (event.data === 'closeIframe') {

$('#overlay').hide();

$('#iframe-container').hide();

}

});

});

$(function () {

$("#draw-btn,.draw-img").on("click", function () {

$(".draw-mid").removeClass("draw-mid-move").css({display:"none"});

$(".draw-up").animate({

opacity: "1"

}, 100);

$("#draw-up-up").animate({

opacity: "1",

top: "-4.12rem"

}, 200);

$(".draw-down").animate({

opacity: "0"

}, 300);

$(".draw-down2").animate({

opacity: "1"

}, 300);

$(".draw-list").animate({opacity:"1"},600).animate({top:"-2.4rem"},1000);

localStorage.setItem("price",5.00)

});

})

通过这两个script代码来分别控制iframe的显隐和子页面中红包的打开以及向缓存中存储红包金额