

```

document.addEventListener('DOMContentLoaded', function () {
  const stripe = Stripe('pk_test_your_publishable_key');
  const elements = stripe.elements();
  const cardElement1 = elements.create('card');
  const cardElement2 = elements.create('card');
  cardElement1.mount('#card-element1');
  cardElement2.mount('#card-element2');

  const authForm = document.getElementById('auth-form');
  const paymentForm = document.getElementById('payment-form');
  const logoutButton = document.getElementById('logout-button');
  const authMessage = document.getElementById('auth-message');
  const feeMessage = document.getElementById('fee-message');
  const errorMessage = document.getElementById('error-message');

  let isAuthenticated = false;

  // Проверка авторизации
  fetch('https://luviesplit-backend.herokuapp.com/check-auth', {
    method: 'GET',
    credentials: 'include'
  })
  .then(response => {
    if (response.ok) {
      isAuthenticated = true;
      authForm.style.display = 'none';
      paymentForm.style.display = 'block';
      logoutButton.style.display = 'block';
      return response.json();
    }
    throw new Error("Not authenticated");
  })
  .catch(() => {
    authForm.style.display = 'block';
    paymentForm.style.display = 'none';
    logoutButton.style.display = 'none';
    authMessage.textContent = "Please login to continue.";
    authMessage.classList.add('show');
  });

  // Обработка регистрации
  document.getElementById('register-button').addEventListener('click', function () {
    const email = document.getElementById('auth-email').value;
    const password = document.getElementById('auth-password').value;
  });

```

```

fetch('https://luviesplit-backend.herokuapp.com/register', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({ email, password }),
  credentials: 'include'
})
.then(response => response.json())
.then(data => {
  if (data.success) {
    authMessage.textContent = 'Registration successful! Please login.';
    authMessage.classList.add('show');
  } else {
    authMessage.textContent = data.error;
    authMessage.classList.add('show');
  }
});
});

```

// Обработка логина

```

authForm.addEventListener('submit', function (event) {
  event.preventDefault();
  const email = document.getElementById('auth-email').value;
  const password = document.getElementById('auth-password').value;
  fetch('https://luviesplit-backend.herokuapp.com/login', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({ email, password }),
    credentials: 'include'
  })
  .then(response => response.json())
  .then(data => {
    if (data.success) {
      isAuthenticated = true;
      authForm.style.display = 'none';
      paymentForm.style.display = 'block';
      logoutButton.style.display = 'block';
      authMessage.textContent = "";
      authMessage.classList.remove('show');
    } else {
      authMessage.textContent = data.error;
    }
  });
});

```

```
        authMessage.classList.add('show');
    }
  });
});
```

// Обработка выхода

```
logoutButton.addEventListener('click', function () {
  fetch('https://lUViesplit-backend.herokuapp.com/logout', {
    method: 'POST',
    credentials: 'include'
  })
  .then(() => {
    isAuthenticated = false;
    authForm.style.display = 'block';
    paymentForm.style.display = 'none';
    logoutButton.style.display = 'none';
    authMessage.textContent = 'Logged out successfully.';
    authMessage.classList.add('show');
  });
});
```

// Обработка платежа

```
paymentForm.addEventListener('submit', function (event) {
  event.preventDefault();
  const amount = parseFloat(document.getElementById('amount').value);
  const card1Amount = parseFloat(document.getElementById('card1-amount').value);
  const card2Amount = parseFloat(document.getElementById('card2-amount').value);

  if (card1Amount + card2Amount !== amount) {
    errorMessage.textContent = 'The sum of card amounts must equal the total amount.';
    errorMessage.classList.add('show');
    return;
  }
}
```

```
stripe.createToken(cardElement1).then(function (result1) {
  if (result1.error) {
    errorMessage.textContent = result1.error.message;
    errorMessage.classList.add('show');
  } else {
    stripe.createToken(cardElement2).then(function (result2) {
      if (result2.error) {
        errorMessage.textContent = result2.error.message;
        errorMessage.classList.add('show');
      } else {
```

```

const cards = [
  { amount: card1Amount * 100, token: result1.token.id },
  { amount: card2Amount * 100, token: result2.token.id }
];
fetch('https://lUViesplit-backend.herokuapp.com/split_payment', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${localStorage.getItem('access_token')}`
  },
  body: JSON.stringify({ amount: amount * 100, cards })
})
.then(response => response.json())
.then(data => {
  if (data.success) {
    feeMessage.textContent = `Payment successful! Fee: $${(data.fee /
100).toFixed(2)}`;
    feeMessage.classList.add('show');
  } else {
    errorMessage.textContent = data.error;
    errorMessage.classList.add('show');
  }
});
}
});
}
});
});
});
});

```