



گزارش پروژه کارآموزی

Crypto Payment Gateway

محسن پاکزاد

استاد کارآموزی: دکتر صدرالدینی

تعریف پروژه

هدف از انجام این پروژه، نوشتن درگاه پرداختی ای است که با استفاده از رمز ارز ([Crypto Currency](#)) ها کار کند.

به طور کلی در یک سناریوی پرداخت ما با موجودیت های زیر طرف هستیم:

- **فروشنده:** که تامین کننده کالا یا خدماتی است
- **مشتري:** که قصد خرید یا بهره برداری از خدمات فروشنده را دارد
- **درگاه پرداخت:** که به عنوان واسط بین فروشنده و خریدار عمل می کند و هزینه ای را از مشتري گرفته و به فروشنده می دهد

بدین صورت که فروشنده اجناس یا خدماتی را که ارائه می دهد با ارز های صادره دولتی (که به آن ها [Fiat Currency](#) نیز گویند) قیمت گذاری کرده و مشتري برای پرداخت این هزینه ها، از سایت فروشنده به یک درگاه پرداخت هدایت شده و در آن جا نیز با استفاده از همین ارز های صادره دولتی پرداخت خود را انجام می دهد.

اما در اینجا تفاوتی که یک درگاه پرداخت معمولی با یک درگاه پرداخت کریپتویی دارد این است که در مرحله پرداخت وجه در درگاه پرداخت، ارز هایی که می توانیم با آن ها پرداخت مان را انجام دهیم، از نوع رمز ارز (Crypto Currency) هستند.

لازم به ذکر است که در این روش همچنان فروشنده برای قیمت گذاری کالا و یا خدمات خود از ارز های صادره دولتی استفاده می کند و پولی که نیز درگاه پرداخت به او می دهد نیز از همین نوع است.

فایده استفاده از درگاه پرداخت کریپتویی هم این است که از آن جایی که رمز ارز ها محدود به دولت و یا کشور خاصی نیستند، می توان از آن ها در شرایطی که دسترسی به ارز های معمول دشوار یا نشدنی هستند استفاده کرد.

نحوه کارکرد پروژه

بصورت کلی، کاربران یک درگاه پرداخت **فروشنندگان** هستند که قصد دارند در ازاء چیزی از مشتری خود پولی دریافت کنند.

در قدم اول فروشنندگان وارد سایت های درگاه پرداخت شده و به عنوان کاربر در آن جا ثبت نام می کنند و بعد از ثبت نام، به عنوان یک کاربر درگاه پرداخت می توانند در سیستم پرداخت درست کنند و کسی را برای انجام آن به درگاه پرداخت هدایت کنند.

در ادامه وقتی پرداختی ایجاد شد، فروشنده مشتری را از سایت خودش به درگاه پرداخت هدایت می کند و مشتری در درگاه پرداخت از رمز ارز های موجود یکی را انتخاب می کند که پرداختش را با استفاده از آن دهد.

سپس درگاه پرداخت معادل رمز ارز انتخاب شده با مقدار هزینه قابل پرداخت که توسط فروشنده مشخص شده را در همان لحظه محاسبه کرده و از مشتری می خواهد این مقدار را به آدرس یکی از کیف پول های درگاه پرداخت واریز کند.

بعد از اتمام پرداخت توسط مشتری، دوباره در همان لحظه، درگاه پرداخت مقدار رمز ارز واریزی را در ازاء ارز دولتی مورد نظر فروشنده می فروشد و مبلغ فروخته شده را به حساب فروشنده واریز می کند.

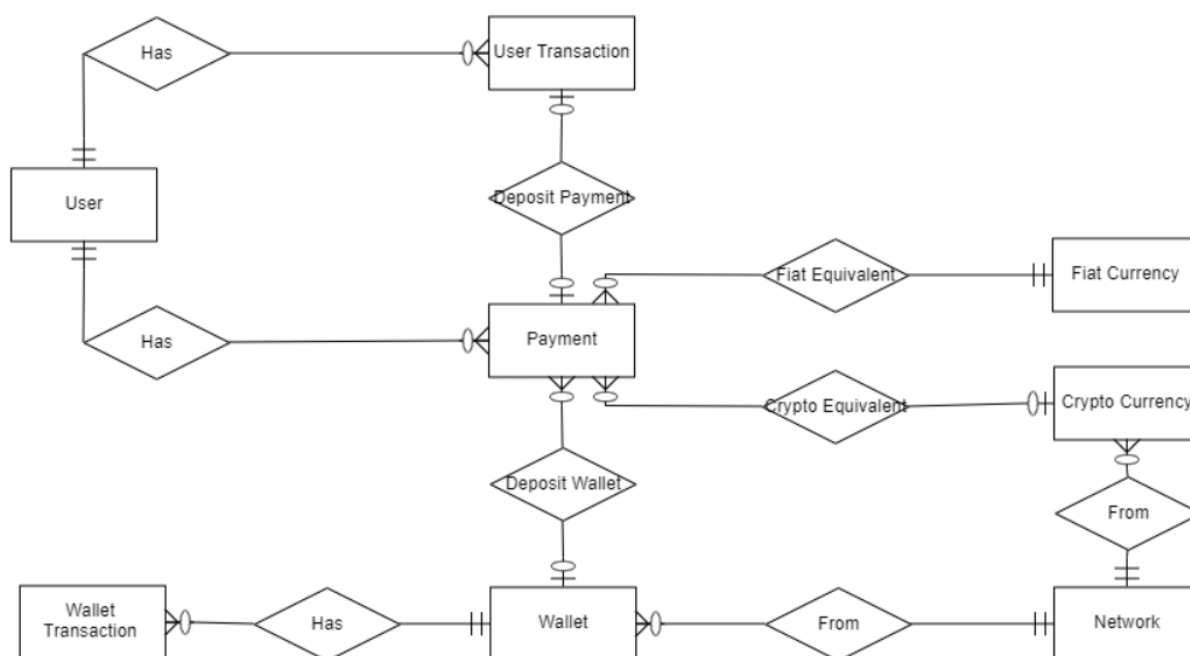
و در قدم آخر فروشنده به حساب کاربری خود در درگاه پرداخت رفته و در آن جا می تواند مقدار پول های واریزی به حساب خود را مشاهده و از آن برداشت کند.

آشنایی با موجودیت های سیستم

موجودیت هایی که در سیستم با آن ها سرو کار خواهیم داشت به شرح زیر هستند:

- کاربر (User): که نمایانگر کاربران سیستم ما هستند
- تراکنش کاربر (User Transaction): که نمایانگر واریز یا برداشت پول از حساب کاربران سیستم است
- پرداخت (Payment): که نمایانگر یک پرداخت است که باید توسط مشتری انجام شود
- ارز رایج دولتی (Fiat Currency): که نمایانگر ارز های رایج دولتی است
- رمز ارز (Crypto Currency): که نمایانگر ارز دیجیتال در یک شبکه بلاک چین است
- شبکه (Network): که نمایانگر شبکه بلاک چین رمز ارز ها است
- کیف پول (Wallet): که نمایانگر کیف پول در یک شبکه بلاک چین است
- تراکنش کیف پول (Wallet Transaction): که نمایانگر تراکنش هایی هست که بر کیف پول های رمز ارز ها زده شده است

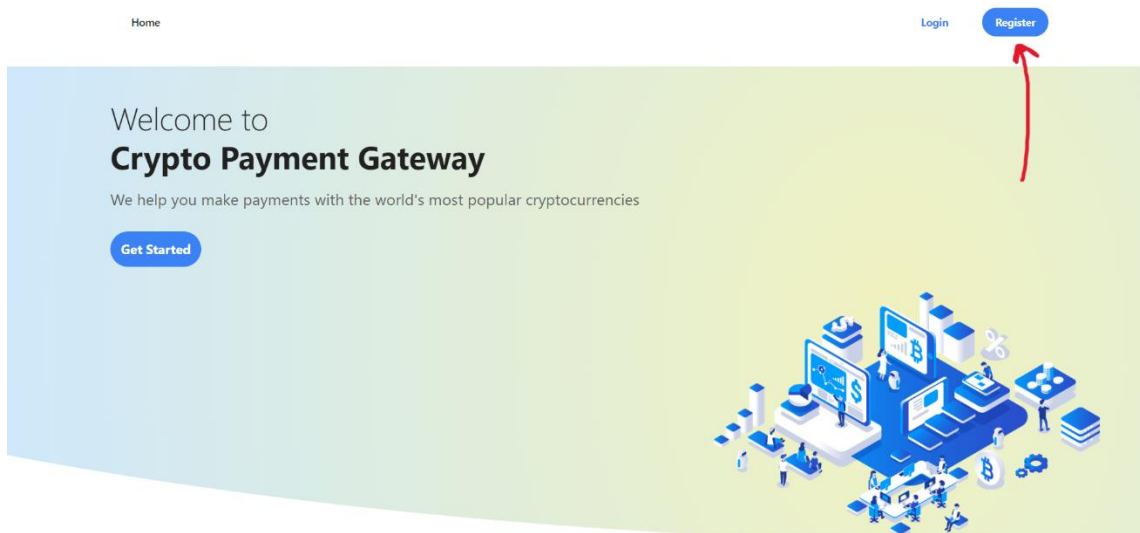
در ادامه نمودار ER موجودیت های سیستم و نوع روابط شان آورده شده است:



ERD

ثبت نام و ورود به سیستم

قبل از آن که بتوانیم اولین پرداخت مان را آغاز کنیم، لازم است در سیستم ثبت نام کنیم. برای این کار ابتدا باید وارد صفحه اصلی وب سایت درگاه پرداخت مان شویم.

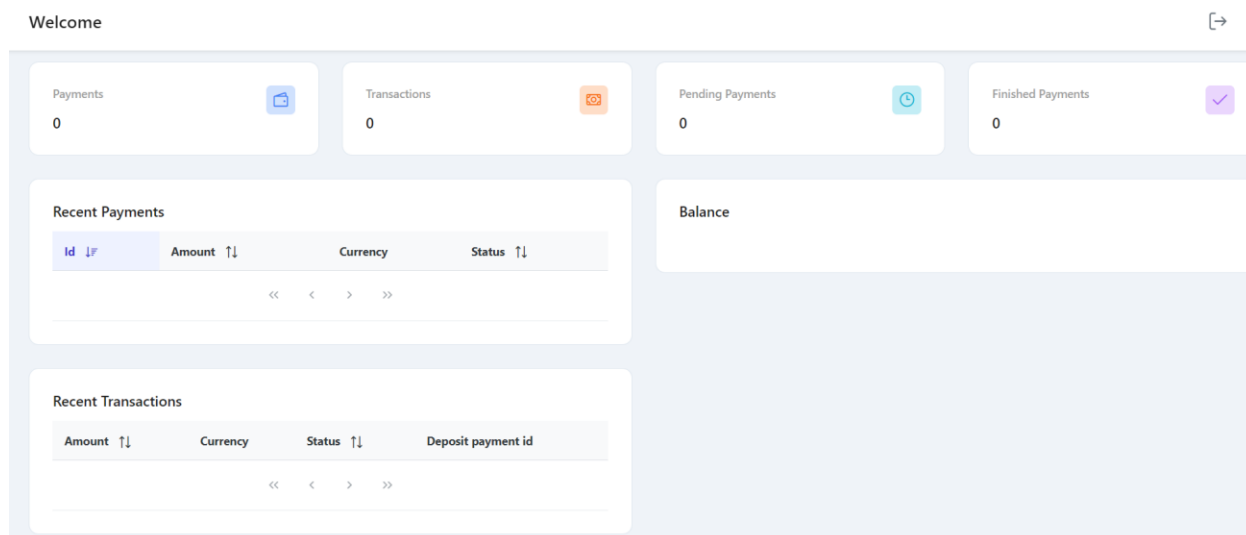


صفحه اصلی وب سایت

برای شروع باید در سیستم ثبت نام کنیم، در اینجا بر روی دکمه ثبت نام (Register) کلیک کرده و فرم ثبت نام را پر می‌کنیم.

فرم ثبت نام

بعد از انجام دادن عمل ثبت نام، به داشبورد پروفایل کاربری خود هدایت می‌شویم که در آن جا می‌توانیم اطلاعات مربوط به پرداخت های انجام شده، تراکنش ها و مقدار موجودی حساب مان را ببینیم.



داشبورد پروفایل کاربری

بعد از ثبت نام اولیه نیز برای ورود مجدد به حساب کاربری، در صفحه اولیه روی دکمه ورود (Login) کلیک کرده و با پر کردن فرمی مشابه با فرم ثبت نام به صفحه داشبورد پروفایل کاربری خود هدایت خواهیم شد.

ساخت و شروع اولین پرداخت

حال که در سیستم ثبت نام کردیم، میتوانیم ساخت اولین پرداخت خود را آغاز کنیم. در ادامه، قدم به قدم از نحوه ساخت پرداخت تا برداشت پول واریزی از حساب و اینکه هر یک از این قدم ها توسط چه کسی انجام می شود را خواهیم دید.

لازم به ذکر است از آن جایی که مراحل که توسط فروشنده انجام می شوند در وب سایت او و به صورت کد نوشته می شود، برای اینکه بتوانیم بدون نیاز به وجود سایت فروشنده این مراحل را ببینیم از یک ابزار فرستنده درخواست های Http استفاده می کنیم که بتواند با یک رابط کاربری، کاری که در وب سایت فروشنده انجام می شود را برای ما شبیه سازی کند.

قدم اول (انجام شونده توسط فروشنده)

گرفتن لیست واحد های ارزی دولتی پشتیبانی شده:

در ابتدا با فرستادن یک درخواست با نوع GET به سیستم، لیست واحد های ارزی دولتی پشتیبانی شده را دریافت می‌کنیم:

The screenshot shows a REST client interface. The request is a GET to `http://localhost:8080/api/fiat-currencies` with a status of 200 OK, size of 82 Bytes, and time of 12 ms. The response is a JSON array of two currency objects: USD and EUR.

Query	Headers 2	Auth 1	Body 1	Tests	Pre Run New	Response	Headers 7	Cookies	Results				
<p>Query Parameters</p> <table border="1"><thead><tr><th>parameter</th><th>value</th></tr></thead><tbody><tr><td></td><td></td></tr></tbody></table>						parameter	value			<pre>1 [2 { 3 "id": 1, 4 "name": "Us Dollar", 5 "symbol": "USD" 6 }, 7 { 8 "id": 2, 9 "name": "Euro", 10 "symbol": "EUR" 11 } 12]</pre>			
parameter	value												

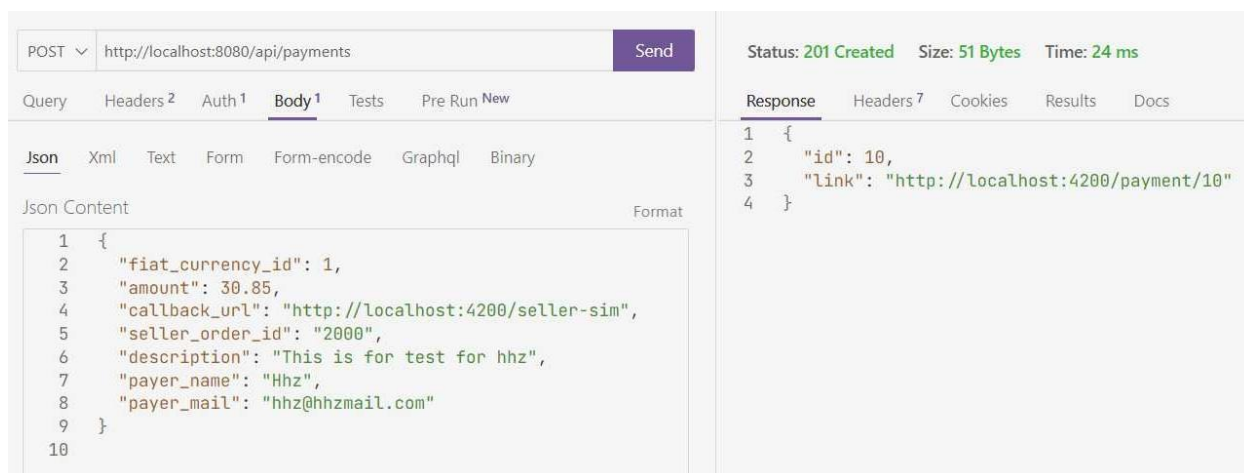
لیست واحد های ارزی دولتی پشتیبانی شده سیستم

با نگاه کردن به لیست می‌توانیم دریابیم که درسیستم ما دو نوع ارز دولتی، دلار آمریکا و یورو تعریف شده که می‌توانیم پرداخت های خود از این واحد ها استفاده کنیم.

قدم سوم (انجام شونده توسط فروشنده)

ساخت پرداخت:

در اینجا با فرستادن درخواستی از نوع POST به سیستم و وارد کردن اطلاعات لازم برای پرداخت آن را می‌سازیم:



ساخت پرداخت

در این مثال، مقدار 30.85 از دلار آمریکا انتخاب شده و سپس آدرسی را که بعد از پرداخت می‌خواهیم مشتری را به آن برگردانیم را مشخص کرده‌ایم. در نهایت هم شناسه سفارش سایت فروشنده و اطلاعات پرداخت و مشتری قرار گرفته است.

در جواب این درخواست نیز، به ما شناسه پرداخت و آدرسی را که باید برای پرداخت در درگاه پرداخت به آن مراجعه کنیم به ما داده شده است.

در ادامه می‌بایستی این آدرس را به مشتری دهیم تا وارد درگاه پرداخت شده و عمل پرداخت را انجام دهد.

قدم چهارم (انجام شونده توسط مشتری)

ورود به درگاه پرداخت و انتخاب رمز ارز:

در این مرحله به عنوان مشتری به آدرس پرداخت داده شده در قدم قبل می‌رویم.

Payment Status

1 Choose Crypto 2 Subscribing Transactions 3 Result

Available Crypto Currencies

	Name ↑↓	Symbol ↑↓	Network
<input type="radio"/>	Ethereum	ETH	ETHEREUM_SEPOLIA
<input type="radio"/>	Ethereum	ETH	ETHEREUM_GOERLI

Select

صفحه انتخاب رمز ارز مورد نظر برای پرداخت

در اینجا در قسمت بالای صفحه می‌توانیم ببینیم هر پرداخت از سه قسمت تشکیل شده و ما الان در این قدم می‌بایستی از لیست پایین صفحه نوع رمز ارزی که مایل هستیم پرداخت مان را با آن انجام دهیم مشخص کنیم.

Payment Status

1 Choose Crypto 2 Subscribing Transactions 3 Result

Available Crypto Currencies

	Name ↑↓	Symbol ↑↓	Network
<input type="radio"/>	Ethereum	ETH	ETHEREUM_SEPOLIA
<input checked="" type="radio"/>	Ethereum	ETH	ETHEREUM_GOERLI

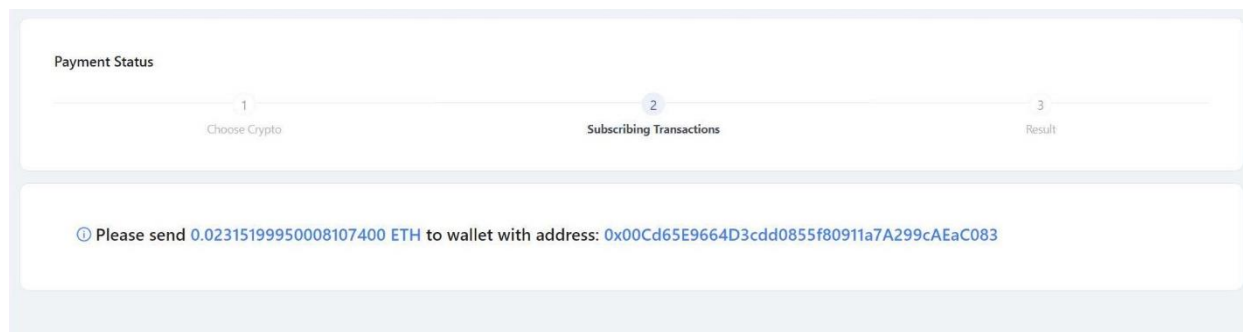
Select

بعد از انتخاب رمز ارز مورد نظر برای پرداخت

قدم پنجم (انجام شونده توسط مشتری)

انجام پرداخت:

در مرحله بعد، وارد این صفحه می‌شویم که به ما نشان می‌دهد چه مقداری از رمز ارز انتخاب شده را باید به آدرس چه کیف پولی بریزیم:



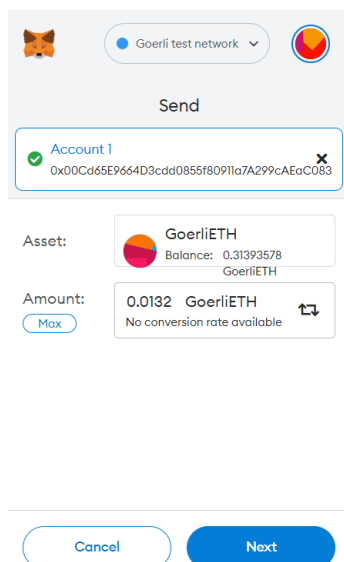
Payment Status

1 Choose Crypto 2 Subscribing Transactions 3 Result

Please send 0.02315199950008107400 ETH to wallet with address: 0x00Cd65E9664D3cdd0855f80911a7A299cAEaC083

در این مرحله می‌توانیم با یک یا تعداد بیشتری از تراکنش، مبلغ واریزی را به این آدرس کیف پول تا قبل از انقضای پرداخت واریز کنیم.

در اینجا ما توسط دو تراکنش مبلغ پرداخت را واریز خواهیم کرد و برای این منظور از MetaMask که یک سیستم مدیریت کیف پول است که به ما اجازه می‌دهد با شبکه های رمز ارز ها تامل داشته و در آن جا تراکنش انجام دهیم استفاده می‌کنیم.



Goerli test network

Send

Account 1
0x00Cd65E9664D3cdd0855f80911a7A299cAEaC083

Asset: GoerliETH
Balance: 0.31393578
GoerliETH

Amount: 0.0132 GoerliETH
No conversion rate available

Cancel Next

نمونه ارسال رمز ارز در MetaMask

هر تراکنش واریزی به کیف کیف پول معین شده نیز به صورت خودکار توسط سیستم از شبکه رمز ارز مربوطه خوانده شده و اطلاعات مربوط به هر تراکنش انجام شده در لیستی در پایین صفحه نمایش داده خواهد شد و تا انجام کامل پرداخت، مبلغ واریزی آن از مبلغ قابل پرداخت کم خواهد شد.

Payment Status

1
Choose Crypto
2
Subscribing Transactions
3
Result

① Please send **0.009951999500081073 ETH** to wallet with address: **0x00Cd65E9664D3cdd0855f80911a7A299cAEaC083**

Received Transactions

Hash	From Address	Value
0xdf6310ae19d95a6dcea32c8d2055a4aae6954b6006692f8239bf2fb496eb4097	0x750fa5ef39a0759d6023b5f65eecb3a185a75f05	0.0132

پس از انجام اولین تراکنش

و در نهایت پس از واریز کامل مبلغ قابل پرداخت، تراکنش به حالت انجام شده تبدیل می‌شود و در این حالت می‌توانیم با کلیک بر روی دکمه اتمام تراکنش به مرحله آخر برویم.

Payment Status

1
Choose Crypto
2
Subscribing Transactions
3
Result

✔ Payment done, please press complete transaction button to continue

Received Transactions

Hash	From Address	Value
0xdf6310ae19d95a6dcea32c8d2055a4aae6954b6006692f8239bf2fb496eb4097	0x750fa5ef39a0759d6023b5f65eecb3a185a75f05	0.0132
0x88018dedc6e7abd825382f7da4315726cc259da561d3474335a2d3c890a0929f	0x750fa5ef39a0759d6023b5f65eecb3a185a75f05	0.01

Complete Transaction

پس از انجام دومین تراکنش و اتمام پرداخت

قدم ششم (انجام شونده توسط مشتری)

اتمام پرداخت و بازگشت به سایت پذیرنده:

پس از کلیک بر روی دکمه اتمام پرداخت در قدم قبل به صفحه نتیجه پرداخت منتقل می‌شویم که در آن جا وضعیت پرداخت را به می‌دهد و سپس بعد از مدتی ما را به آدرسی که فروشنده موقع ساخت پرداخت آن را مشخص کرده بود منتقل می‌کند.



اتمام پرداخت

هنگام منتقل شدن نیز چندین پارامتر به این آدرس پاس داده می‌شود که نمایانگر وضعیت پرداخت، علت ناموفق بودن پرداخت (در صورت ناموفق بودن پرداخت)، شناسه پرداخت و شناسه سفارش سایت فروشنده است.



انتقال به سایت پذیرنده و پارامترهای پرداخت ارسالی

قدم هفتم (انجام شونده توسط فروشنده)

تایید پرداخت:

پس از تکمیل پرداخت توسط مشتری و بازگشت به سایت فروشنده، در اینجا لازم است که پرداخت از سوی فروشنده نیز تایید شود تا پرداخت کامل شده و پول حاصله از پرداخت به حساب او واریز شود.

برای این کار هم لازم است به سادگی، شناسه پرداخت را که در مرحله قبل به سایت فروشنده فرستاده شده را با یک درخواست از نوع POST به سیستم ارسال شود.

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/api/payments/verify`. The request body is a JSON object with `"id": 10`. The response is a 200 OK status with a JSON body containing payment verification details.

Request:

```
POST http://localhost:8080/api/payments/verify
```

Response:

```
{
  "id": 10,
  "user_id": 1,
  "fiat_currency_id": 1,
  "amount": "30.8500",
  "callback_url": "http://localhost:4200/seller-sim",
  "seller_order_id": "2000",
  "description": "This is for test for hhz",
  "payer_name": "Hhz",
  "payer_phone": null,
  "payer_mail": "hhz@hhzmail.com",
  "status": "Verified",
  "crypto_currency_id": 2,
  "crypto_amount": "0.02315199950008107400",
  "dest_wallet_id": 2,
  "created_at": "2023-01-09T19:48:56.083827",
  "expired_at": "2023-01-09T20:18:56.083828",
  "done_at": "2023-01-09T20:03:42.127334",
  "verified_at": "2023-01-09T20:09:55.479764"
}
```

تایید پرداخت

قدم هشتم (انجام شونده توسط فروشنده)

مشاهده موجودی و برداشت از حساب:

در قدم آخر برای مشاهده موجودی و برداشت از حساب مان می‌بایستی همان گونه که در قبل گفته شد، به حساب کاربری خود وارد شویم.

The screenshot shows a user dashboard with a 'Welcome' header and a navigation bar. The dashboard is divided into four main sections: Payments (8), Transactions (10), Pending Payments (4), and Finished Payments (4). Below these are two main content areas. The left area, titled 'Recent Payments', contains a table with columns: Id, Amount, Currency, and Status. A red arrow points to the first row of the table. The right area, titled 'Balance', shows the account balance in US Dollar (48,1800 USD) and Euro (61,7500 EUR), with input fields for withdrawal amounts and 'Withdrawal' buttons.

Id	Amount	Currency	Status
10	30.8500	Us Dollar	Finished
9	30.8500	Us Dollar	Expired
8	30.8500	Euro	Finished
7	82.8500	Euro	Done
6	82.8500	Euro	Waiting

پرداخت های اخیر و موجودی کاربر

در اینجا می‌توانیم ببینیم که در قسمت تراکنش های اخیر، تراکنش ما اضافه شده و در قسمت راست صفحه نیز موجودی ما برای هر رمز ارز نشان داده شده است.

در قسمت پایین صفحه هم می‌توانیم تراکنش های اخیر حساب خود را ببینیم که در اینجا می‌توانیم ببینیم که مبلغ آخرین پرداخت به حساب ما واریز شده است.

The screenshot shows a 'Recent Transactions' table with columns: Amount, Currency, Status, and Deposit payment id. A red arrow points to the first row of the table.

Amount	Currency	Status	Deposit payment id
30.6800	Us Dollar	Deposit	10
30	Euro	Withdrawal	
30.9200	Euro	Deposit	8
2	Euro	Withdrawal	
1	Us Dollar	Withdrawal	

تراکنش های اخیر کاربر

در ادامه نیز می‌توانیم با توجه به موجودی هر کدام از ارز هایی که داریم به سیستم درخواست برداشت وجه بدهیم. در اینجا به طور مثال درخواست برداشت مقدار 26.5 از موجودی دلار حساب خود را می‌دهیم.

Balance

Us Dollar

\$

48.1800 USD

↓

26.5

Withdrawal

Euro

\$

61.7500 EUR

↓

Amount

Withdrawal

درخواست برداشت

در نهایت پس از مدتی برداشت انجام شده و می‌توانیم تراکنش برداشت را در قسمت تراکنش های اخیر حساب کاربری خود ببینیم.

Recent Transactions			
Amount ↑↓	Currency	Status ↑↓	Deposit payment id
26.5000	Us Dollar	Withdrawal	
30.6800	Us Dollar	Deposit	10
30	Euro	Withdrawal	
30.9200	Euro	Deposit	8
2	Euro	Withdrawal	
<div><< < 1 2 3 > >></div>			

تراکنش برداشت

جزئیات پیاده سازی

پیاده سازی این پروژه نیز به دو قسمت **هسته** (Backend) و **رابط کاربری** (Frontend) تقسیم گردیده است.

برای پیاده سازی قسمت هسته از فریمورک [Actix-web](#) استفاده شده که از زبان برنامه نویسی Rust استفاده می‌کند که با ساختار غیر همگام (Asynchronous) خود یکی از سریع ترین و کاراترین فریم ورک ها در این حوزه است.

برنامه نویسی [Rust](#) نیز یک زبان مدرن و سطح پایین و با کارایی بالاست که شعارش این است که: "زبانی که همه را توانمند می کند که نرم افزار قابل اعتماد و کارآمد بنویسند" و در سال های اخیر رقیبی برای زبان هایی مثل C و C++ شناخته می‌شود.

برای قسمت رابط کاربری نیز از فریمورک [Angular](#) استفاده شده که توسط شرکت Google پشتیبانی و توسعه داده می‌شود.

در کنار Angular نیز از کتابخانه [PrimeNg](#) استفاده شده است که با داشتن اجزاء (component) از پیش آماده برای رابط کاربری، کار ما را آسان می‌کند و به ما کمک می‌کند که لازم نباشد تمامی اجزاء را خودمان از پایه بنویسیم.

ویژگی های کلی برنامه

- ✓ استفاده بهینه از پردازنده و سرعت بالای اجرا، به علت اینکه هسته اجرایی برنامه با زبان Rust نوشته شده و فایل اجرایی این برنامه مستقیماً بدون واسطه توسط سیستم عامل اجرا می‌شوند
- ✓ استفاده بهینه از حافظه، به دلیل آن که هسته اجرایی برنامه از گاریج کالکتور (GC) استفاده نمی‌کند
- ✓ استفاده از ساختار غیر همگام (Asynchronous) که ما اجازه می‌دهد بدون معطل کردن برنامه، به بهترین نحو ممکن از منابع سیستم استفاده کنیم
- ✓ رابط کاربری ساده، زیبا و پویا (Responsive)
- و در آخر پروژه به صورت متن باز (open source) توسعه داده شده است و تمامی کدهای منبع آن در مخزن های (repository) زیر قابل دسترس هستند:
- لینک مخزن هسته (Backend):

<https://github.com/mohsenpakzad/crypto-payment-gateway>

لینک مخزن رابط کاربری (Frontend):

<https://github.com/mohsenpakzad/crypto-payment-gateway-front>