

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Informacijske tehnologije

PROJEKTNI ZADATAK

Kriptovalute

Student: Roko Grubić

Datum: 20.02.2021

Sadržaj

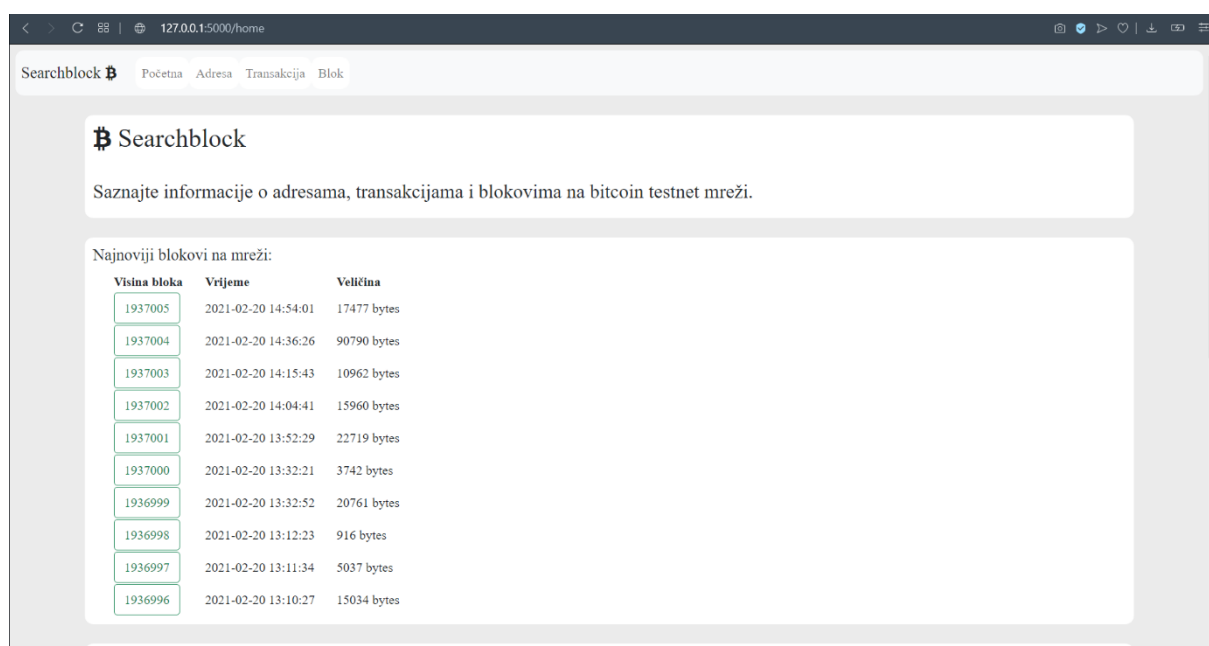
Opis projekta	2
Sučelje	3
Popis korištenih API poziva	4
Izvadci iz programskog koda	10

Opis projekta

Projekt napravljen za kolegij kriptovalute je istraživač blokova (eng. *blockexplorer*) koji daje informacije o blokovima, transakcijama i adresama na temelju API poziva iz Bitcoin Core. Istraživač blokova je realiziran kao web aplikacija (web stranica) na kojoj je moguće vidjeti najnovije blokove i transakcije na bitcoin testnet mreži (za pristup informacijama korišten je poslužitelj koji se nalazi na Sveučilišnom odjelu za stručne studije). Također je moguće upisati određeni blok, transakciju ili adresu te saznati više informacija o njima. Kao podloga je korišten Python programski jezik i Flask internetsko razvojno okruženje (eng. *web framework*). Unutar Pythona, kao i većine drugih programskih jezika, postoje biblioteke (eng. *libraries*) koje se povezuju sa Bitcoin Core API-em i koriste njihove pozive u aplikaciji. Za projekt je korištena biblioteka *python-bitcoinlib* (verzija 0.11.0) čija je referenca i u knjizi „Mastering bitcoin: Programming the open blockchain“ koja je obavezna literatura kolegija. Flask je, po definiciji sa njihove službene stranice, „microframework“ koji pomaže razvojnim inženjerima u što lakšem razvoju internetskih aplikacija preko Pythona. Flask predstavlja laganiju verziju Django (web razvojno okruženje sa svim potrebnim alatima) u kojoj razvojni inženjer sam bira alate koje želi implementirati (u sebi također donosi i internetski poslužitelj koji služi u razvojne svrhe). Razlog više za korištenje ovog razvojnog okruženja je i mogućnost integracije već napisanog programskog kôda iz konzolne aplikacije koji dobavlja informacije sa poslužitelja. Što se tiče internetskog dijela aplikacije korišteni su HTML i CSS koji prikazuju dobivene podatke. Slike sučelja i kôda uz detaljnije informacije bit će prikazane u nastavku dokumenta.

Sučelje

Na Slika 1 i Slika 2 je prikazano sučelje početne stranice. Na stranici se mogu vidjeti posljednji blokovi na mreži kao i transakcije koje se nalaze u mempoolu (zadnjih deset). Klikom bilo na blok ili na transakciju mogu se dobiti dodatne informacije o bloku, odnosno transakciji. U navigacijskoj traci, kao i na dnu stranice postavljene su poveznice na forme za unos određenog bloka, transakcije ili adrese.



Slika 1: Početna stranica

Transakcije u mempoolu:

ID transakcije bloka	Vrijeme	Veličina
7ebfca0e621c1091365e9b3e24b2b6628d43cb0b820b3d61d46f453fc9377	2021-02-20 14:54:08	144 bytes
a3d1d934ef03835d1512f9589b3efb8f01b4114a878c540d388661699624a231	2021-02-20 14:54:31	142 bytes
6dcdd9986680809624ed79ca156712c82bd2b2a17eb1fe37f0f852262d0afa2	2021-02-20 14:55:29	227 bytes
d55a80bf20631839c300de4f4fb07d95d0c22a7180ba8352fa7e74b393988f982	2021-02-20 14:55:59	141 bytes
8e20f6bc5e30ee956b981c18f92e6c3095779f45061304ae365b6f5841aec93a	2021-02-20 14:56:27	141 bytes
f4a21c9f2154739fe6a60792291ef35cdabe3ef04ed1233c00e4e73a97234959	2021-02-20 14:56:27	351 bytes
a9f0fc2aa9def4c44f0bebe71fbf91e5611cd8a07e83435b5db023b097eb7f62	2021-02-20 14:56:33	145 bytes
7d3ffc6d5b3f108fb9b6bd4822b1922c3388aabe9ae3803dcd8c0f49e18945	2021-02-20 14:57:31	142 bytes

Saznajte više o:

- [adresi](#)
- [transakciji](#)
- [bloku](#)

Slika 2: početna stranica

Na Sliku 3 je prikazana stranica na kojoj se nalazi forma za unos određene adrese. Klikom na Pretraži! dobivaju se informacije o adresi. Valja napomenuti kako isto ovako izgledaju i forme za unos bloka ili transakcije (osim drukčijeg natpisa unutar forme).

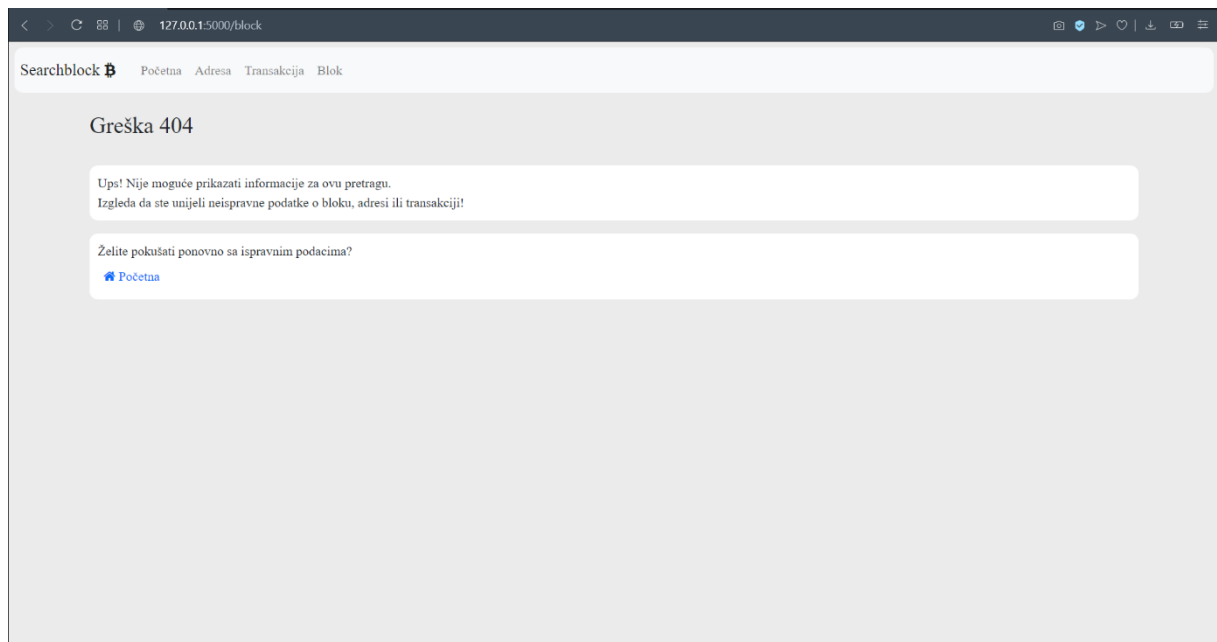
Searchblock **B** Početna Adresa Transakcija Blok

Upišite adresu

Zelite istražiti nešto drugo? [Početna](#)

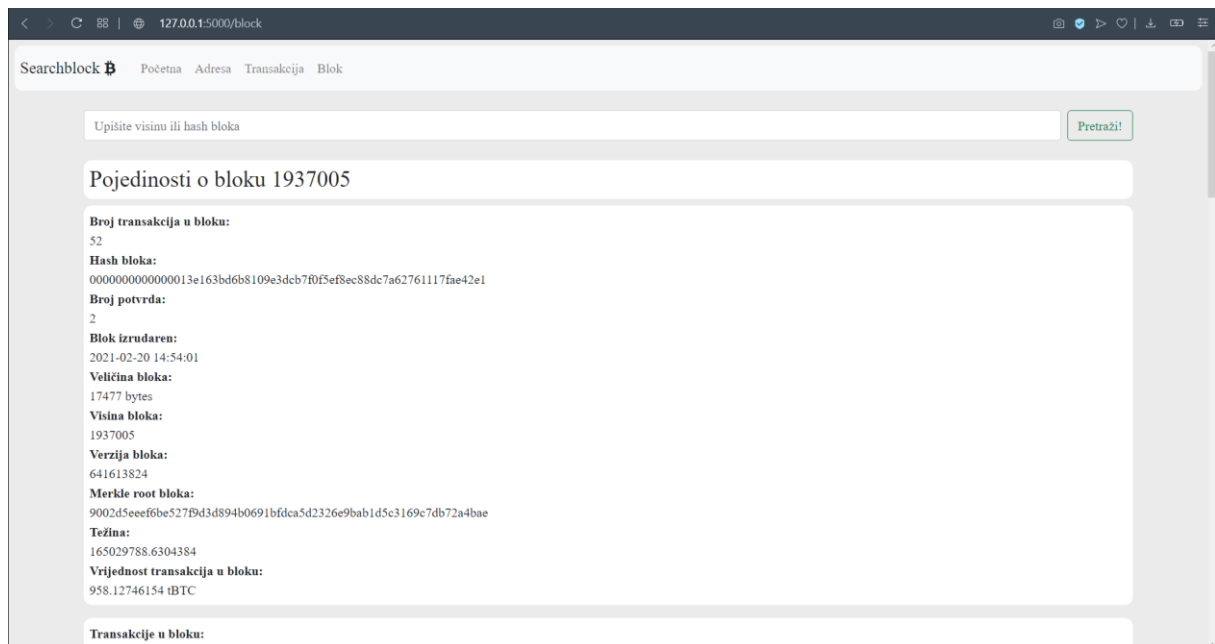
Slika 3: Forma za unos adrese

Dodana je i stranica koja prikazuje grešku 404 (podatci nisu pronađeni) ukoliko se unese krivi podatak, a može se vidjeti na Slika 4.

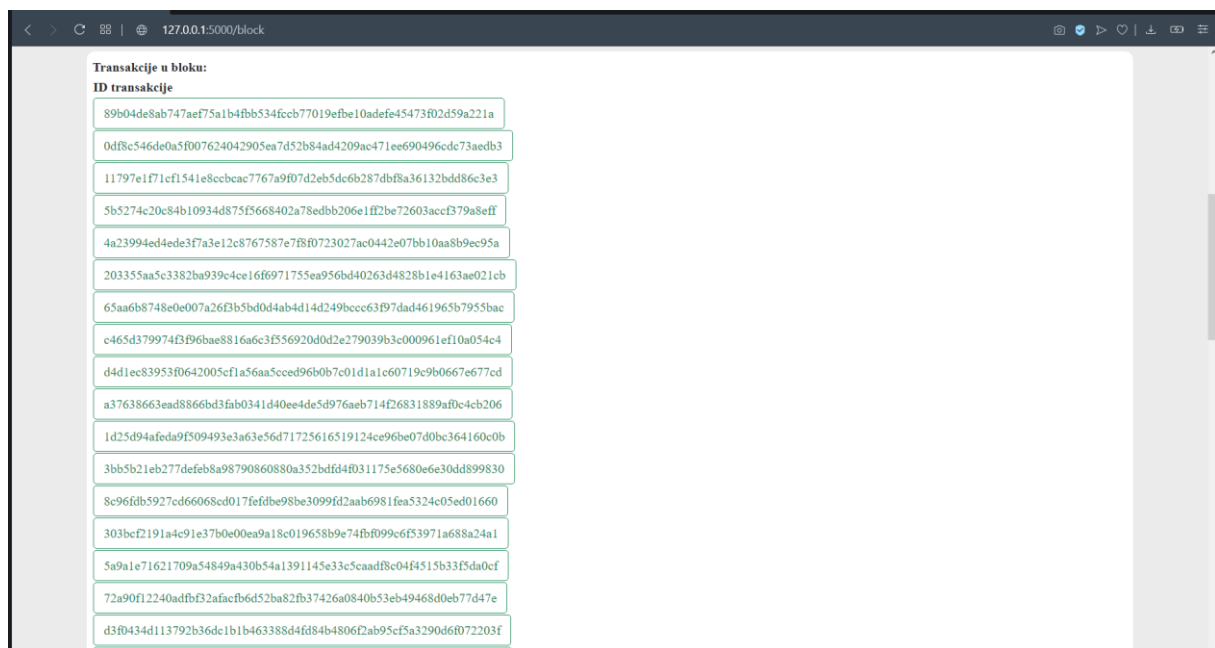


Slika 4: Greška 404

Slijedeće slike prikazuju informacije o određenom bloku. U vrhu stranice je ponovno prikazana forma za unos hasha ili broja bloka koji se želi pretražiti i vide se tehnički detalji o bloku kao i transakcije koje su ušle u njega. Sve to je vidljivo na Slika 5 i Slika 6.



Slika 5: Informacije o bloku



Slika 6: informacije o bloku

Na Sliku 7 su prikazane informacije o određenoj transakciji (ulazi i izlazi transakcija i njihov broj, ukupna vrijednost ulaza i izlaza kao i nagrada za rudara koji je blok uključio u blockchain). Također, na vrhu je vidljiva forma za unos druge transakcije.

The screenshot shows a web browser at the URL 127.0.0.1:5000/transaction. The page has a navigation bar with 'Searchblock' and links for 'Početna', 'Adresa', 'Transakcija', and 'Blok'. Below the navigation bar is a search input field labeled 'Upišite transakciju' with a 'Pretraži!' button. The main content area displays the transaction ID: 'd4d1ec83953f0642005cf1a56aa5cced96b0b7c01d1a1c60719c9b0667e677cd'. It then lists the transaction inputs and outputs. The inputs section shows one input with address 'tb1q49ghnawh3s9theahhu6rkad7a5ghb677xcz458x' and a value of '33.11016444 tBTC'. The outputs section shows two outputs: one with address 'tb1qhm40a3755k3uvfj5ltut5ay3ddk2wx2sew2p4x' and a value of '33.09449735 tBTC', and another with address 'tb1qmg8Sec87a24yf05a8dz0at430jcpn80avnd8' and a value of '0.01552344 tBTC'. At the bottom, a 'Dodatno:' section provides summary statistics: 'Ukupna vrijednost ulaza: 33.11016444 tBTC', 'Ukupna vrijednost izlaza: 33.11002079 tBTC', and 'Nagrada za rudara: 0.00014365 tBTC'.

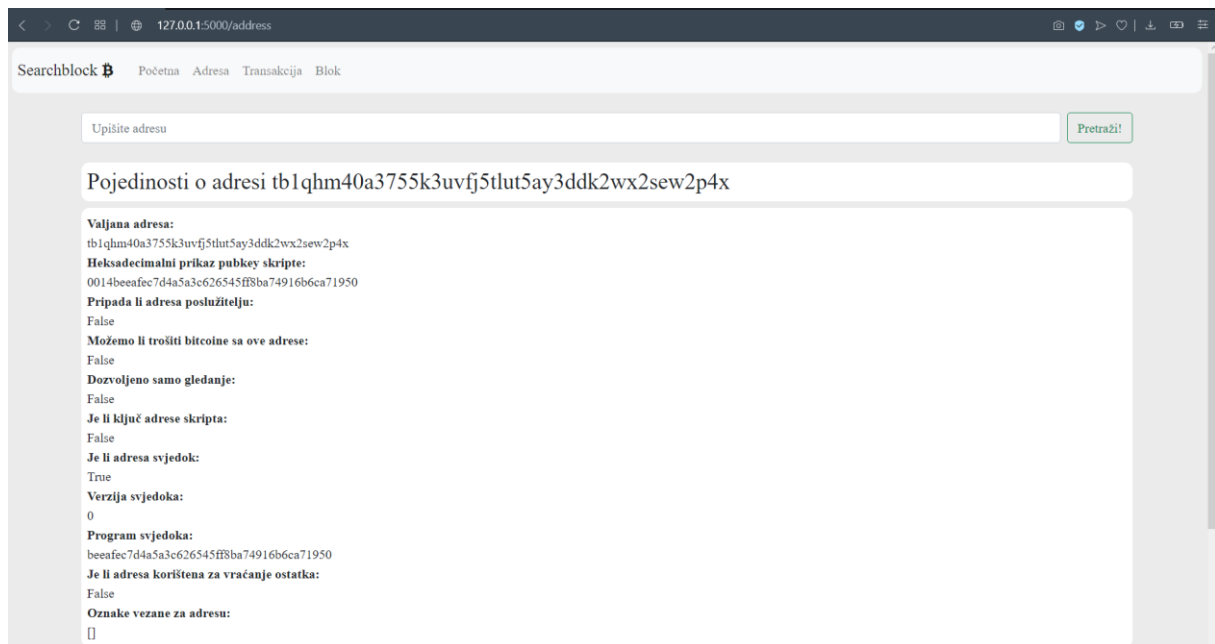
Ulazi transakcije (ukupno 1) i njihove vrijednosti:	
Adresa	Vrijednost ulaza
tb1q49ghnawh3s9theahhu6rkad7a5ghb677xcz458x	33.11016444 tBTC

Izlazi transakcije (ukupno 2) i njihove vrijednosti:	
Adresa	Vrijednost izlaza
tb1qhm40a3755k3uvfj5ltut5ay3ddk2wx2sew2p4x	33.09449735 tBTC
tb1qmg8Sec87a24yf05a8dz0at430jcpn80avnd8	0.01552344 tBTC

Dodatno:	
Ukupna vrijednost ulaza:	33.11016444 tBTC
Ukupna vrijednost izlaza:	33.11002079 tBTC
Nagrada za rudara:	0.00014365 tBTC

Slika 7: Podaci o transakciji

Posljednja slika sučelja za ovu projektnu dokumentaciju donosi dodatne informacije o određenoj adresi. Mogućnost daljnje pretrage je ista kao i na informacijama o bloku, odnosno transakciji. Prikaz na Slika 8.



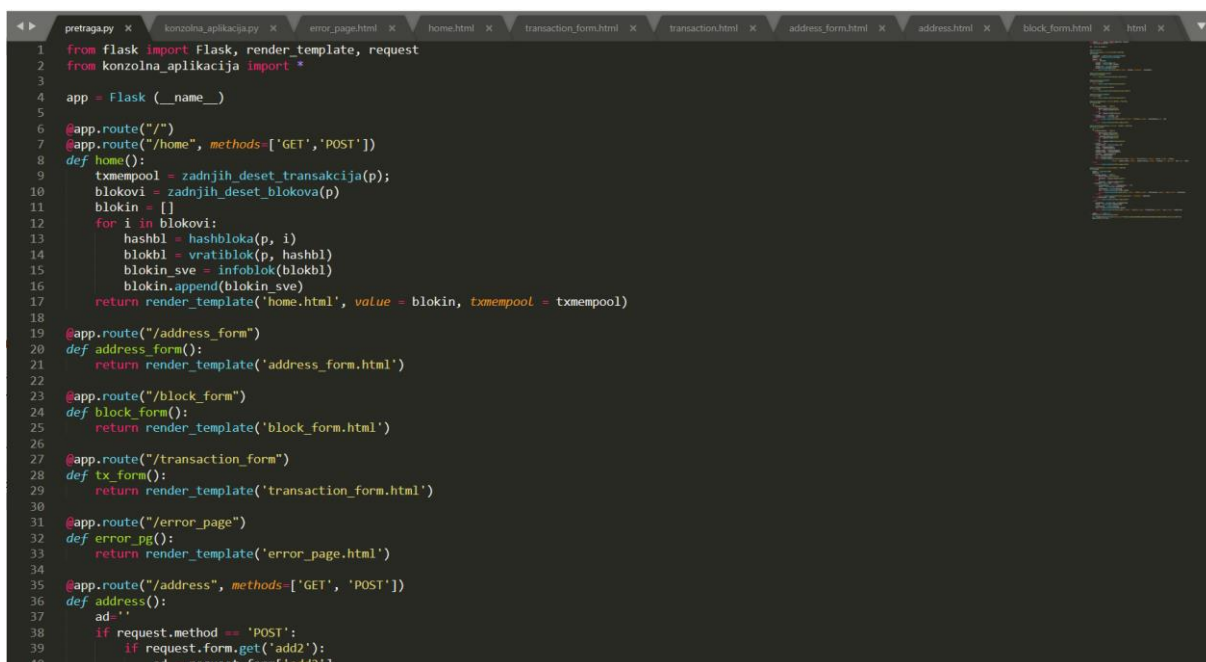
Slika 8: Informacije o adresi

Popis korištenih API poziva

Poziv	Funkcija
<code>getblockcount()</code>	Vraća broj blokova u najdužem (najboljem) lancu na mreži
<code>getrawmempool()</code>	Vraća sve transakcije u mempoolu i informacije o njima
<code>getblockhash()</code>	Vraća hash određenog bloka iz najdužeg (najboljeg) lanca
<code>getblock()</code>	Vraća sve važne informacije o određenom bloku
<code>getrawtransaction()</code>	Vraća sve važne informacije o određenoj transakciji u sirovom (eng. <i>raw</i>) obliku (hex zapis)
<code>decoderawtransaction()</code>	Dekodira „sirovu“ transakciju u čitljiviji oblik
<code>getaddressinfo()</code>	Vraća informacije o određenoj bitcoin adresi

Izvadci iz programskog koda

Na Slika 9, Slika 10 i Slika 11 je prikazana polovica kôda koji obavlja posao u aplikaciji. Na slikama je vidljiva pretraga.py datoteka koja koristi Flask razvojno okruženje i kreira internetsku aplikaciju koja izvršava zadatke. Definirane su putanje do stranica koje prikazuju informacije kao i funkcije koje se obavljaju na svakoj stranici (sadržavaju logiku dobavljanja podataka i slanja tih podataka .html datotekama koje ih prikazuju na internetu). Na dnu datoteke se vidi kôd koji obavlja spajanje na udaljeni poslužitelj.



```
1 from flask import Flask, render_template, request
2 from konzolna_aplikacija import *
3
4 app = Flask(__name__)
5
6 @app.route("/")
7 @app.route("/home", methods=['GET', 'POST'])
8 def home():
9     txmempool = zadnjih_deset_transakcija(p);
10    blokovi = zadnjih_deset_blokova(p)
11    blokin = []
12    for i in blokovi:
13        hashbl = hashbloka(p, i)
14        blokbl = vratiblok(p, hashbl)
15        blokin_sve = infoblok(blokbl)
16        blokin.append(blokin_sve)
17    return render_template('home.html', value = blokin, txmempool = txmempool)
18
19 @app.route("/address_form")
20 def address_form():
21     return render_template('address_form.html')
22
23 @app.route("/block_form")
24 def block_form():
25     return render_template('block_form.html')
26
27 @app.route("/transaction_form")
28 def tx_form():
29     return render_template('transaction_form.html')
30
31 @app.route("/error_page")
32 def error_pg():
33     return render_template('error_page.html')
34
35 @app.route("/address", methods=['GET', 'POST'])
36 def address():
37     ad = ''
38     if request.method == 'POST':
39         if request.form.get('add2'):
40             ad = request.form['add2']
```

Slika 9: Pretraga.py

```
34
35 @app.route("/address", methods=['GET', 'POST'])
36 def address():
37     ad = ''
38     if request.method == 'POST':
39         if request.form.get('add2'):
40             ad = request.form['add2']
41         else:
42             ad = request.form['adresa']
43     if len(ad) >= 14 and len(ad) <= 74:
44         saznajadresu = adresa(p, ad)
45         return render_template('address.html', title = "address", value = saznajadresu, ad = ad)
46     else:
47         return render_template("error_page.html")
48
49 @app.route("/transaction", methods = ['GET', 'POST'])
50 def transaction():
51     tx = ''
52     if request.method == 'POST':
53         if request.form.get('tx2'):
54             tx = request.form['tx2']
55         elif request.form.get('tx3'):
56             tx = request.form['tx3']
57         else:
58             tx = request.form['transakcija']
59     if len(tx) == 64:
60         transakcijaa = transakcija(p, tx)
61         ulazi = transakcijaa[0]
62         izlazi = transakcijaa[1]
63         ukupno_ulaza = transakcijaa[2]
64         ukupno_izlaza = transakcijaa[3]
65         razlika = transakcijaa[4]
66         inn = transakcijaa[5]
67         out = transakcijaa[6]
68         return render_template('transaction.html', title = "transaction", value1 = ulazi, value2 = izlazi,
69                               value3 = ukupno_ulaza, value4 = ukupno_izlaza, value5 = razlika, tx = tx, inn = inn, out = out);
70     else:
71         return render_template("error_page.html")
72
73 @app.route("/block", methods=['GET', 'POST'])
```

Slika 10: Pretraga.py

```
76 blokinfo = ''
77 if request.method == 'POST':
78     if request.form.get('blok2'):
79         blokinfo = request.form['blok2']
80     else:
81         blokinfo = request.form['blok']
82 if blokinfo.isnumeric() == False:
83     if len(blokinfo) == 64 and blokinfo[0] == '0':
84         blok = vratiblok(p, blokinfo);
85         informacije = infoblok(blok)
86         sve = vrijednosttransakcija(p, blok)
87         return render_template('block.html', title = "block", value = informacije, ukupno = sve, visina = blokinfo);
88     else:
89         return render_template("error_page.html", blokinfo = blokinfo)
90 elif int(blokinfo) > height:
91     return render_template("error_page.html")
92 else:
93     hashblokaa = hashbloka(p, int(blokinfo))
94     blok = vratiblok(p, hashblokaa);
95     informacije = infoblok(blok)
96     sve = vrijednosttransakcija(p, blok)
97     return render_template('block.html', title = "block", value = informacije, ukupno = sve, visina = blokinfo);
98
99 if __name__ == "__main__":
100     bitcoin.SelectParams('testnet');
101     p = bitcoin.rpc.RawProxy("Podaci za poslužitelja");
102     app.run(debug=True);
```

Slika 11: Pretraga.py

Druga polovica kôda je prikazana na Slika 12, Slika 13 i Slika 14. Na njima se vidi prilagođena konzolna aplikacija koja koristi biblioteku python-bitcoinlib koja dohvaća podatke sa udaljenog poslužitelja koristeći API pozive koji su nabrojani u poglavlju „Popis korištenih API poziva“. Te informacije se dalje prilagode,

spreme u spremnike podataka (integer, string, lista itd.) i šalju dalje do .html datoteka.

```
1 import time
2 import bitcoin
3 import bitcoin.rpc
4 import sys
5
6 def brojblokova(p):
7     visina = p.getblockcount()
8     return visina
9
10 def zadnjih_deset_blokova(p):
11     blokovi = []
12     num = 1
13     zadnji = brojblokova(p)
14     blokovi.append(zadnji)
15     while(num < 10):
16         blokovi.append(zadnji-num)
17         num+=1
18     return blokovi
19
20 def zadnjih_deset_transakcija(p):
21     transakcije=[]
22     memorijski_bazen = p.getrawmempool(True);
23     keys=memorijski_bazen.keys()
24     vel = len(keys)
25     granica=0
26     if vel > 10:
27         granica = 10
28     else:
29         granica = vel
30     for i in keys:
31         if (granica > 0):
32             vrijeme_tx = memorijski_bazen[i]['time'];
33             datum = time.strftime("%Y-%m-%d %H:%M:%S", time.gmtime(vrijeme_tx));
34             transakcije.append((i, memorijski_bazen[i]['size'], datum));
35             granica -= 1
36     return transakcije
37
38 def hashbloka(p, visina):
39     blockhash = p.getblockhash(visina)
40     return blockhash
```

Slika 12: Konzolna_aplikacija.py

```
42 def vratiblok(p, blockhash):
43     blok = p.getblock(blockhash)
44     blok_brojtx = blok['nTx']
45     return blok
46
47 def vrijednosttransakcija(p, blok):
48     transactions = blok['tx'];
49     value = 0
50     for txid in transactions:
51         tx_value = 0
52         raw_tx = p.getrawtransaction(txid)
53         decoded_tx = p.decoderawtransaction(raw_tx)
54         for output in decoded_tx['vout']:
55             tx_value = tx_value + output['value']
56         value = value + tx_value
57     return value
58
59 def infoblok(blok):
60     info = []
61     block_numbertx = blok['nTx']
62     info.append(block_numbertx)
63
64     hash_block = blok['hash']
65     info.append(hash_block)
66
67     confirmations = blok['confirmations']
68     info.append(confirmations)
69
70     time_block = blok['time']
71     date=time.strftime("%Y-%m-%d %H:%M:%S", time.gmtime(time_block))
72     info.append(date)
73
74     size_block = blok['size']
75     info.append(size_block)
76
77     height_block = blok['height']
78     info.append(height_block)
79
80     version_block = blok['version']
81     info.append(version_block)
```

Slika 13: Konzolna_aplikacija.py

```
93
94 def transakcija(p, adresa):
95     ulazi_transakcija = []
96     izlazi_transakcija = []
97     ukupno_ulaza = 0
98     ukupno_izlaza = 0
99     inn = 0
100    out = 0
101    sirova_transakcija = p.getrawtransaction(adresa)
102    dekodirana_transakcija = p.decoderawtransaction(sirova_transakcija)
103    ulazi = dekodirana_transakcija['vin']
104    for i in ulazi:
105        identifikator = i['vout']
106        raw_tx = p.getrawtransaction(i['txid'])
107        decoded_tx = p.decoderawtransaction(raw_tx)
108        for j in decoded_tx['vout']:
109            if j['n'] == identifikator:
110                adresa_za_printanje_u = str(j['scriptPubKey']['addresses'])
111                adresa_za_printanje_u = adresa_za_printanje_u.strip("[,,'"]
112                ulazi_transakcija.append((adresa_za_printanje_u, j['value']))
113                ukupno_ulaza += j['value']
114                inn += 1
115
116    izlazi = dekodirana_transakcija['vout']
117    for j in izlazi:
118        adresa_za_printanje_i = str(j['scriptPubKey']['addresses'])
119        adresa_za_printanje_i = adresa_za_printanje_i.strip("[,,'"]
120        izlazi_transakcija.append((adresa_za_printanje_i, j['value']))
121        ukupno_izlaza += j['value']
122        out += 1
123
124    razlika = ukupno_ulaza - ukupno_izlaza
125    return (ulazi_transakcija, izlazi_transakcija, ukupno_ulaza, ukupno_izlaza, razlika, inn, out)
126
127 def adresa(p, broj):
128    informacija = p.getaddressinfo(broj)
129    addressinfo = []
130    addressinfo.append(informacija['address'])
131    addressinfo.append(informacija['scriptPubKey'])
132    addressinfo.append(informacija['ismine'])
133    return addressinfo
```

Slika 14: Konzolna_aplikacija.py

Nakon dobavljanja podataka, potrebno je te iste podatke prikazati na internetu. Na Sliku 15 i Sliku 16 je prikazana layout.html datoteka koja se koristi kao predložak za sve ostale stranice koje samo dodaju svoje podatke na predložak. Vidljiva je referenca na *bootstrap* temu koja ljepše prikazuje dugmad, navigacijsku traku i forme za unos podataka, referenca na *fontawesome* koji daje mogućnost korištenja raznih ikona na stranici i referenca na JavaScript skripte koje su također dio bootstrapa. Kako na ovoj, tako i na ostalim stranicama ukomponiran je CSS koji doprinosi boljem vizualnom izgledu.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="
sha384-g1j66koqN00vuy+HMDDP7az0uL0xtbfIcaT9wJKHr8BdVdVhYtFAAsrekWkMP1" crossorigin="anonymous">
8 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.15.1/css/fontawesome.min.css"
9 {% if title %}
10 <title> Search Block - {{title}} </title>
11 {% else %}
12 <title> Search Block </title>
13 {% endif %}
14
15 <style type="text/css">
16 body {
17 background-color: rgb(235, 235, 235);
18 background-size: auto;
19 text-align: justify;
20 font-size: 1em;
21 font-family: 'Architects Daughter', serif;
22 margin: 7px;
23 }
24 </style>
25
26 </head>
27
28 <body>
29 <style type="text/css">
30 .navbar {
31 border-radius: 10px;
32 }
33 a {
34 padding: 7px;
35 }
36 </style>
37 <nav class="navbar navbar-expand-lg navbar-light bg-light">
38 <a class="navbar-brand" href="/home">Searchblock <i class="fa fa-btc w3-large"></i> </a>
39 <div class="collapse navbar-collapse" id="navbarSupportedContent">

```

Slika 15: Layout.html

```

28 <body>
29 <style type="text/css">
30 .navbar {
31 border-radius: 10px;
32 }
33 a {
34 padding: 7px;
35 }
36 </style>
37 <nav class="navbar navbar-expand-lg navbar-light bg-light">
38 <a class="navbar-brand" href="/home">Searchblock <i class="fa fa-btc w3-large"></i> </a>
39 <div class="collapse navbar-collapse" id="navbarSupportedContent">
40 <ul class="navbar-nav mr-auto">
41 <li class="nav-item active">
42 <a class="nav-link" href="/home">Početna <span class="sr-only"></span></a>
43 </li>
44 <li class="nav-item">
45 <a class="nav-link" href="/address_form">Adresa</a>
46 </li>
47 <li class="nav-item">
48 <a class="nav-link" href="/transaction_form">Transakcija</a>
49 </li>
50 <li class="nav-item">
51 <a class="nav-link" href="/block_form">Blok</a>
52 </li>
53 </ul>
54 </div>
55 </nav>
56 <div class="container">
57 {% block content %}
58 {% endblock %}
59 </div>
60 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js" integrity="sha384-q2kxQ16AaE6UbzuKqyBE9/u/
KzioAlnx2maXQhIDx9d4/zp80k3f+M7DPm+Ib6IU" crossorigin="anonymous"></script>
61 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.min.js" integrity="
sha384-pQQkAEnwaBkjpqZ8RU1fF1AKtTcHJwF13pb1pTlHXybj4HpMv79HV3hI4NKxyj" crossorigin="anonymous"></script>
62 </body>
63
64 </html>

```

Slika 16: Layout.html

Sljedeće slike prikazuju kôd home.html datoteke koji predstavlja početnu stranicu istraživača blokova. Vidljivo je kako se uz korištenje HTML-a i CSS-a prikazuju podaci koji su poslani iz .py datoteka prikazanih u dokumentu. Na ovakav način se prikazuju i dodatne informacije o transakcijama, blokovima i

adresa (datoteke transaction.html, address.html i block.html) Prikaz na Slika 17 i Slika 18.

```
1 {% extends "layout.html" %}
2
3 {% block content %}
4
5 <style type="text/css">
6   span {
7     display: grid;
8     background-color: white;
9     border-radius: 10px;
10    padding: 10px;
11  }
12  li {
13    display: grid;
14    background-color: white;
15    border-radius: 10px;
16  }
17  h5 {
18    display: grid;
19    background-color: white;
20    border-radius: 10px;
21  }
22  table {
23    display: grid;
24    border-collapse: separate;
25    border-spacing: 25px 0;
26  }
27 </style>
28
29 <br/>
30 <span>
31   <h2><i class="fa fa-btc w3-large"></i> Searchblock </h2> <br/>
32   <h4>Saznajite informacije o adresama, transakcijama i blokovima na bitcoin testnet mreži.</h4>
33 </span>
34 <br/>
35 <span>
36   <h5>Najnoviji blokovi na mreži: <br/></h5>
37   <table>
38     <tr>
39       <th>Visina bloka </th>
40       <th>Vrijeme </th>
```

Slika 17: Home.html

```
46   <button class="btn btn-outline-success" type="submit" name = "blok2" value="{{i[5]}}"> {{i[5]}} </button>
47 </td>
48 <td> {{i[3]}} </td>
49 <td> {{i[4]}} bytes </td>
50 </tr>
51 {%endfor%}
52 </table>
53 </span>
54 <br/>
55 <span>
56   <h5>Transakcije u mempoolu: <br/></h5>
57   <table>
58     <tr>
59       <th> ID transakcije bloka </th>
60       <th> Vrijeme </th>
61       <th> Veličina </th>
62     </tr>
63     {%for i in txmempool%}
64     <tr>
65       <td> <form action="{{ url_for('transaction') }}" method="post" >
66         <button class="btn btn-outline-success" type="submit" name = 'tx2' value="{{i[0]}}"> {{i[0]}} </button>
67       </form>
68     </td>
69     <td> {{i[2]}} </td>
70     <td> {{i[1]}} bytes </td>
71   </tr>
72   {%endfor%}
73 </table>
74 </span>
75 <br/>
76 <br/>
77 <span>
78   <h5>Saznajite više o: <br/>
79   <li> <a href="/address_form"> <i class="fa fa-address-card w3-large" > adresi </i> </li>
80   <li> <a href="/transaction_form"> <i class="fa fa-exchange w3-large"> transakciji </i> </li>
81   <li> <a href="/block_form"> <i class="fa fa-cube w3-large" > bloku </i> </li>
82 </span>
83
84 {% endblock content %}
```

Slika 18: Home.html

Na Slika 19 je prikazan kôd koji se koristi za formu unosa transakcije (ista stvar je i za forme unosa adrese i bloka).


```
1 {% extends "layout.html" %}
2
3 {% block content %}
4
5 <br/>
6 <form action="{% url_for('transaction') %}" method="post" class="d-flex">
7   <input class="form-control me-2" type="search" placeholder="Upišite transakciju" aria-label="Search" type="text" name="transakcija">
8   <button class="btn btn-outline-success" type="submit">Pretraži!</button>
9 </form>
10 <br/>
11 <br/>
12 <p>Želite istražiti nešto drugo? <a href="/home" /> <i class="fa fa-home w3-large" > Početna </i> </p>
13
14 {% endblock content %}
15
```

Slika 19: Transaction_form.html

Za kraj, na Slika 20 je vidljiv kôd za stranicu error_page.html koja se pojavi ukoliko su unešene krive informacije o bloku/adresi/transakciji.

```
1 {% extends "layout.html" %}
2
3 {% block content %}
4
5 <style type="text/css">
6   p{
7     display: grid;
8     background-color: white;
9     border-radius: 10px;
10    padding: 10px;
11  }
12 </style>
13
14 <br/>
15 <h3>Greška 404</h3> <br/>
16 <p>
17   Ups! Nije moguće prikazati informacije za ovu pretragu. <br/>
18   Izgleda da ste unijeli neispravne podatke o bloku, adresi ili transakciji! <br/>
19 </p>
20 <p>Želite pokušati ponovno sa ispravnim podacima? <a href="/home" /> <i class="fa fa-home w3-large" > Početna </i>
21
22 {% endblock content %}

```

Slika 20: Error_page.html