Spremembe Algator
Marko Amon

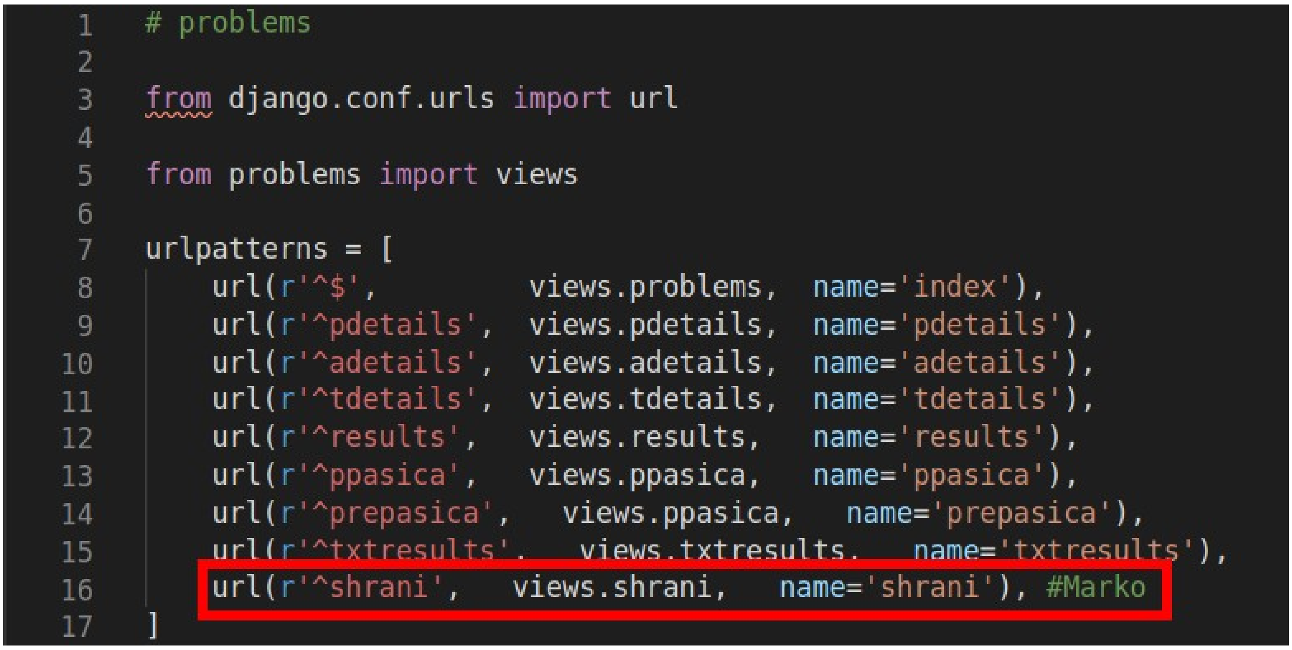
Datoteke:

- → **AlgatorWeb/problems/**
  - ↳ urls.py
  - ↳ views.py
  - ↳ **templates**
    - ↳ pdetails.html
- → **static/js/**
  - ↳ saving_descriptions.js
- → **templates/**
  - ↳ base.html
- → **??/python2.7/bleach/**
  - ↳ sanitizer.py

Kode so na voljo v datoteki **koda.txt**.

1. Spremeba *problems/urls.py*

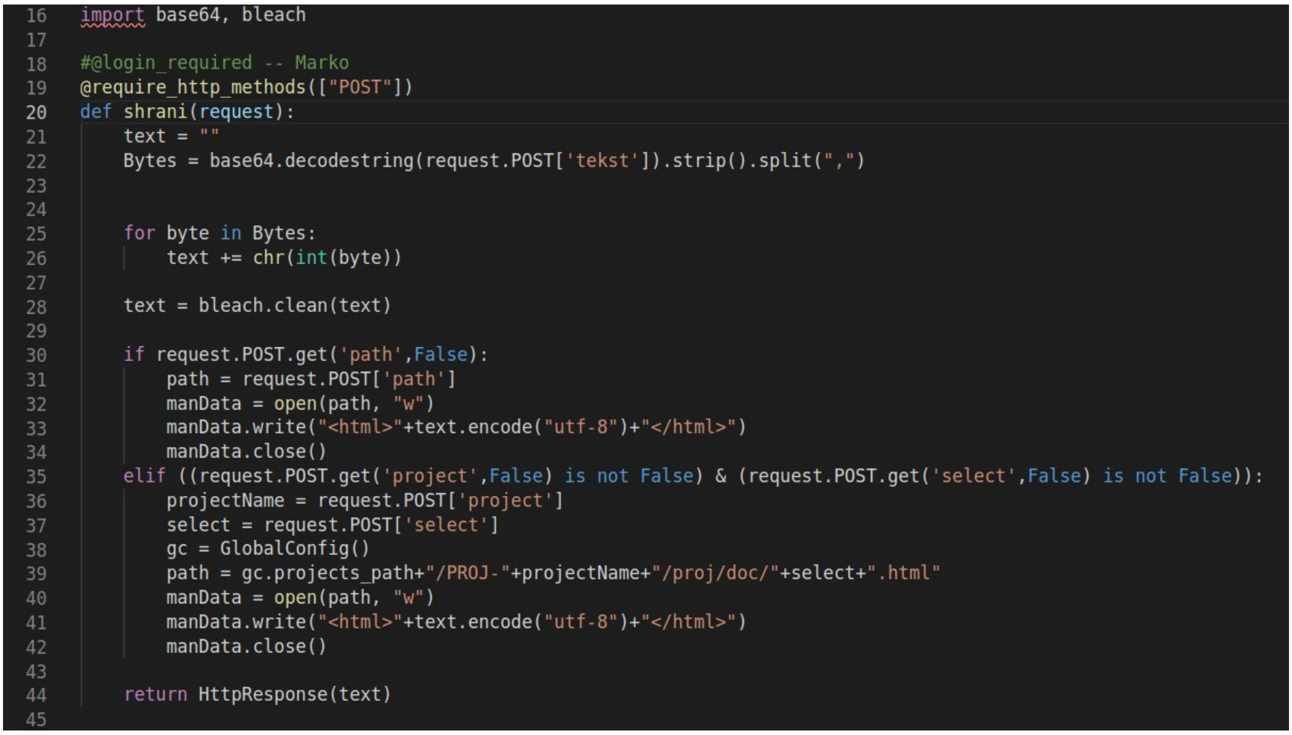
Dodamo vrstico, ki je označena na sliki. Koda je na voljo pod št. “1.”.



```
# problems

from django.conf.urls import url

from problems import views

urlpatterns = [
    url(r'^$',          views.problems,  name='index'),
    url(r'^pdetails',  views.pdetails,  name='pdetails'),
    url(r'^adetails',  views.adetails,  name='adetails'),
    url(r'^tdetails',  views.tdetails,  name='tdetails'),
    url(r'^results',   views.results,   name='results'),
    url(r'^ppasica',   views.ppasica,   name='ppasica'),
    url(r'^prepasica',   views.ppasica,   name='prepasica'),
    url(r'^txtresults',   views.txtresults,   name='txtresults'),
    url(r'^shrani',   views.shrani,   name='shrani'), #Marko
]
```

2. Sprememba *problems/views.py*

Dodamo funkcijo shrani in import base64 ter bleach.
Bleach moramo namestiti s “pip install”, natančnejša navodila so na voljo na naslovu https://github.com/mozilla/bleach. Koda je na voljo pod št. “2.”.

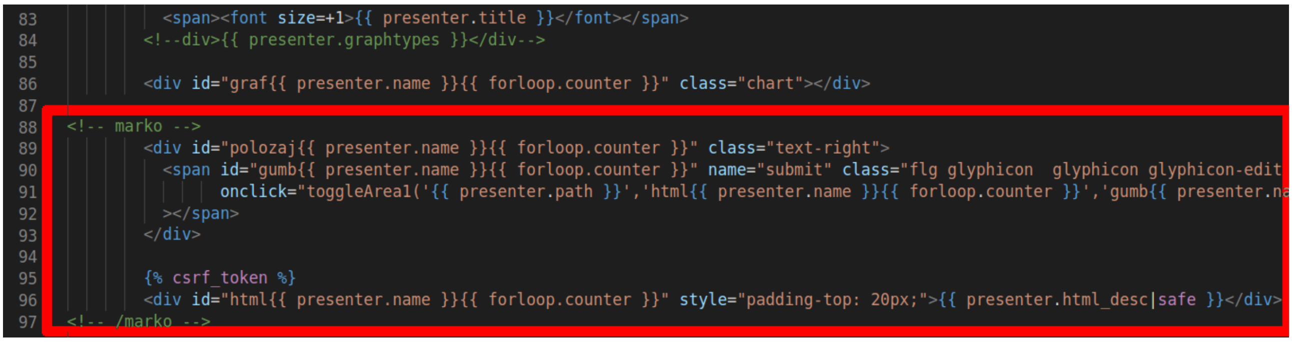


```
import base64, bleach

#@login_required -- Marko
@require_http_methods(["POST"])
def shrani(request):
    text = ""
    Bytes = base64.decodestring(request.POST['tekst']).strip().split(",")


    for byte in Bytes:
        text += chr(int(byte))

    text = bleach.clean(text)

    if request.POST.get('path',False):
        path = request.POST['path']
        manData = open(path, "w")
        manData.write("<html>"+text.encode("utf-8")+"</html>")
        manData.close()
    elif ((request.POST.get('project',False) is not False) & (request.POST.get('select',False) is not False)):
        projectName = request.POST['project']
        select = request.POST['select']
        gc = GlobalConfig()
        path = gc.projects_path+"/PROJ-"+projectName+"/proj/doc/"+select+".html"
        manData = open(path, "w")
        manData.write("<html>"+text.encode("utf-8")+"</html>")
        manData.close()

    return HttpResponse(text)

```

3. Sprememba *problems/templates/pdetails.html*

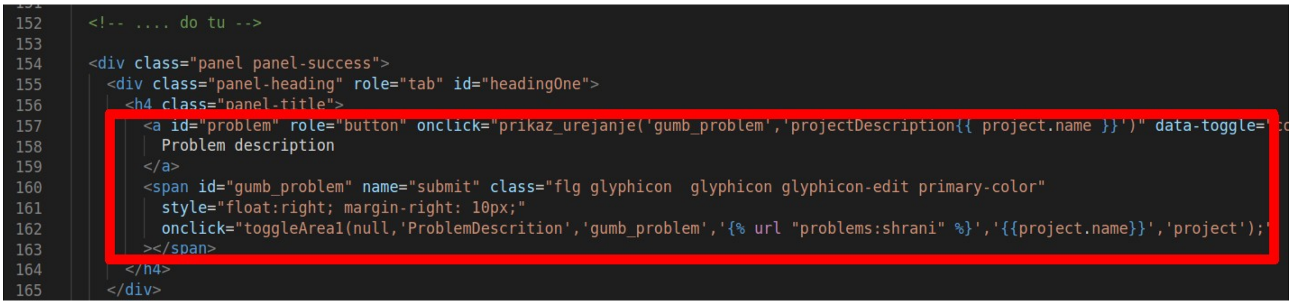
K importom javascript datotek dodamo import javascript kode, potrebne za shranjevanje datotek pod številko “3.0.1”.

V nadaljevanju poiščemo prikaz besedila pod grafom in div, kjer prikazujemo besedilo nadomestimo s kodo pod številko “3.1.1”. Koda je uporabna samo pri opisih grafov, z njo dodamo gumb za urejanje in omogočimo prikaz editorja.



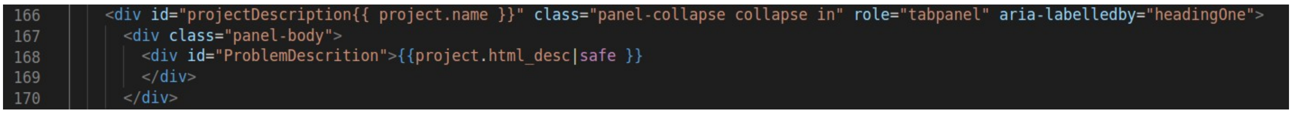
```
                <span><font size=+1>{{ presenter.title }}</font></span>
            <!--div>{{ presenter.graphtypes }}</div-->

            <div id="graf{{ presenter.name }}{{ forloop.counter }}" class="chart"></div>

<!-- marko -->
            <div id="polozaj{{ presenter.name }}{{ forloop.counter }}" class="text-right">
                <span id="gumb{{ presenter.name }}{{ forloop.counter }}" name="submit" class="flg glyphicon  glyphicon glyphicon-edit
                      onclick="toggleArea1('{{ presenter.path }}','html{{ presenter.name }}{{ forloop.counter }}','gumb{{ presenter.na
                ></span>
            </div>

            {% csrf_token %}
            <div id="html{{ presenter.name }}{{ forloop.counter }}" style="padding-top: 20px;">{{ presenter.html_desc|safe }}</div>
<!-- /marko -->
```

Podobno kodo uporabimo potem še za ostala besedila, kjer se zadeve rahlo razlikujejo. Prvo poiščemo <h4> element, ki ima class “panel-title” in znotraj <h4> oznak takoj po naslovu izven <a> oznake dodamo kodo pod številko “3.2.1”. S tem smo dodali gumb v vrstico z naslovom.



```
<!-- .... do tu -->

<div class="panel panel-success">
  <div class="panel-heading" role="tab" id="headingOne">
    <h4 class="panel-title">
      <a id="problem" role="button" onclick="prikaz_urejanje('gumb_problem','projectDescription{{ project.name }}')" data-toggle="c
        Problem description
      </a>
      <span id="gumb_problem" name="submit" class="flg glyphicon  glyphicon glyphicon-edit primary-color"
        style="float:right; margin-right: 10px;"
        onclick="toggleArea1(null,'ProblemDescrition','gumb_problem','{% url "problems:shrani" %}','{{project.name}}','project');
      ></span>
    </h4>
  </div>
```

Za urejanje besedila moramo dodati kodo pod številko “3.2.2”.



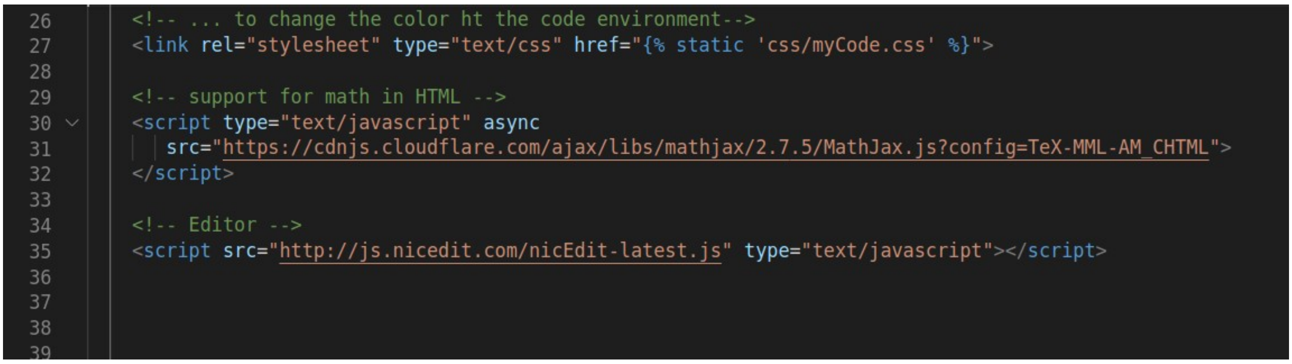
```
  <div id="projectDescription{{ project.name }}" class="panel-collapse collapse in" role="tabpanel" aria-labelledby="headingOne">
    <div class="panel-body">
      <div id="ProblemDescrition">{{project.html_desc|safe }}
      </div>
    </div>
```

4. Ustvarimo *static/js/saving_descriptions.js*

Dodamo javascript kodo pod številko “4.”.

5. Sprememba *templates/base.html*

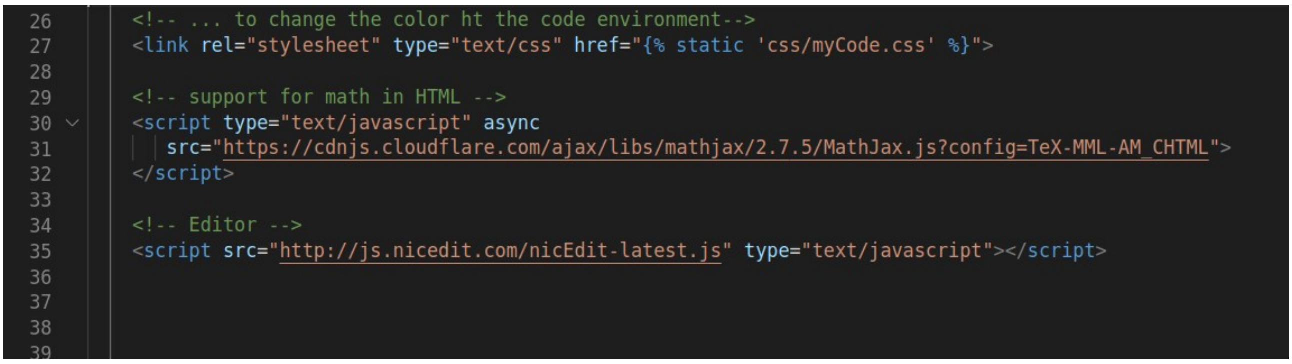
Pod importe skript dodamo še skripto, za naš editor.



```
<!-- ... to change the color ht the code environment-->
<link rel="stylesheet" type="text/css" href="{% static 'css/myCode.css' %}">

<!-- support for math in HTML -->
<script type="text/javascript" async
  src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/MathJax.js?config=TeX-MML-AM_CHTML">
</script>

<!-- Editor -->
<script src="http://js.nicedit.com/nicEdit-latest.js" type="text/javascript"></script>
```

6. *Sprememba* *sanitizer.py*

Prepišemo datoteko s **sanitizer.py**, ki sem jo dodal v zip datoteki. Lokacija datoteke je odvisna od lokacije pythona.



```
<!-- ... to change the color ht the code environment-->
<link rel="stylesheet" type="text/css" href="{% static 'css/myCode.css' %}">

<!-- support for math in HTML -->
<script type="text/javascript" async
  src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/MathJax.js?config=TeX-MML-AM_CHTML">
</script>

<!-- Editor -->
<script src="http://js.nicedit.com/nicEdit-latest.js" type="text/javascript"></script>
```