

بسمه تعالی



دانشکده مهندسی و علوم کامپیوتر  
گزارش نهایی  
پروژه سکوی آنلاین شبکه اجتماعی  
رادیو بهشت  
درس مهندسی اینترنت

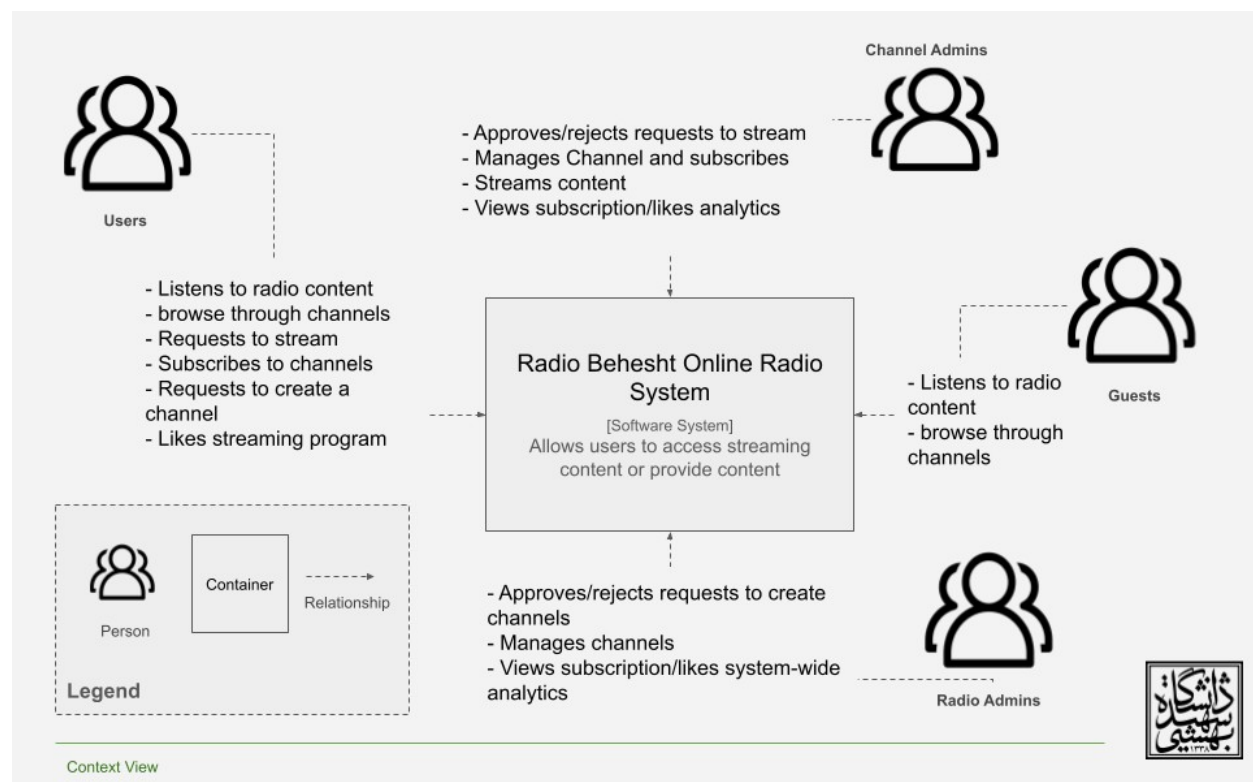
محسن کربلائی امینی، 98242128  
دی 1403

## توصیف پروژه:

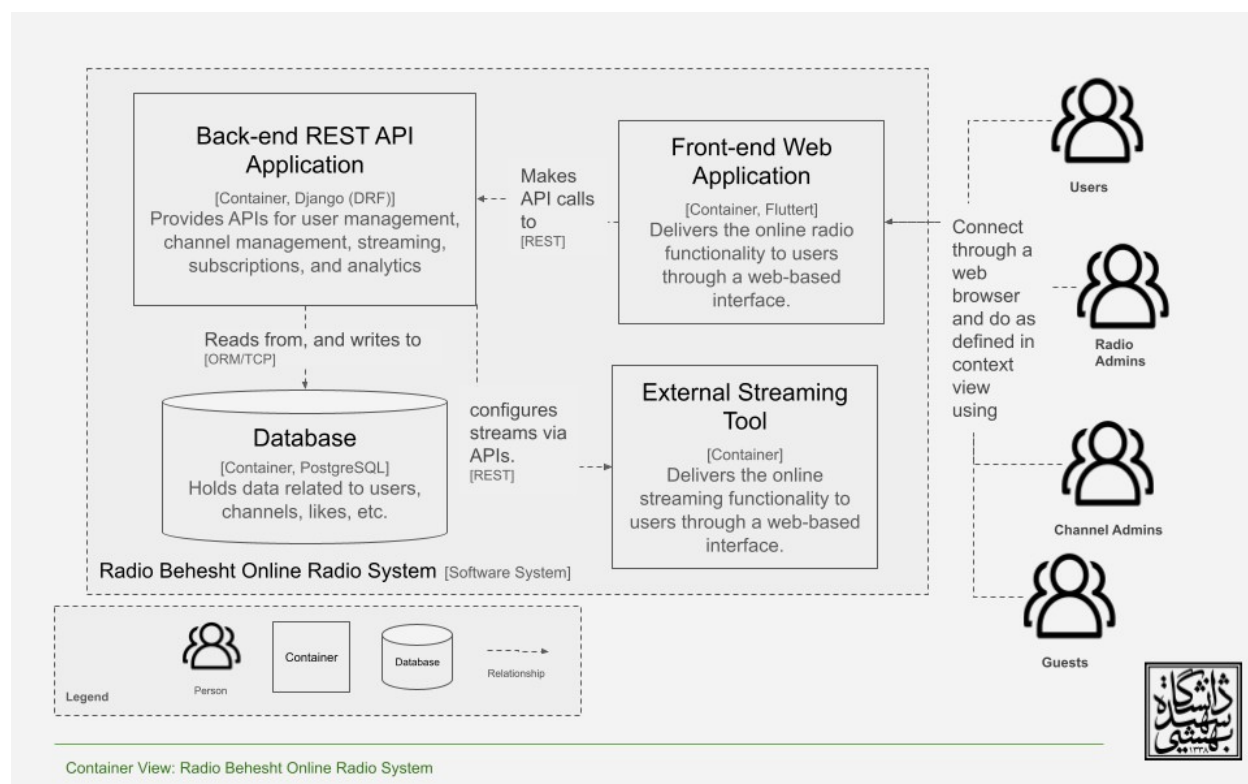
در این پروژه هدف ایجاد بستری برای تولید محتوا در سطح دانشگاه می‌باشد. این سکو دارای ایستگاه‌هایی برای هر دانشکده خواهد بود و کاربران می‌توانند ایستگاه‌ها را انتخاب، و به محتوایی که در قالب زنده از این ایستگاه‌ها پخش می‌شود، گوش کنند. همچنین کاربرانی دارای حساب بوده و وارد حساب خود شده‌اند، می‌توانند ارائه درخواست خود را از طریق فرمی برای اجرای برنامه در زمانی خاص ارسال کنند تا مورد بررسی مدیر این سکو قرار بگیرد. در صورت تایید، هماهنگی‌های لازم برای اجرای برنامه روی ایستگاه مورد درخواست انجام می‌شود. این سکو ارتباطات بین و درون دانشکده‌ای را ارتقا خواهد داد.

## معماری سامانه در قالب دو سطح اول مدل C4:

### :Context view



## :Container view



## اقدامات انجام شده در طول پروژه به صورت کلی:

بکند:

- شروع بکند با فریمورک django
- پیاده سازی احراز هویت بکند با استفاده از djoser
- آزمون احراز هویت بکند با استفاده از curl
- آزمون مشاهده و تغییر کاربران ساخته شده از طریق django admin
- ساخت یک model با نام RadioUser که یکی از نقش های سامانه می باشد.
- داکرایز شدن بکند و ایجاد فایل Dockerfile و docker-compose.yml
- پیکربندی CORS برای اتصال به فرانت اند
- ساخت API برای ثبت درخواست اجرای برنامه توسط کاربران
- ساخت API برای گرفتن لیست ایستگاه های موجود
- پیاده سازی API کاربر
- پیاده سازی تست های curl برای درخواست ها، لیست ایستگاه ها، و کاربر
- پیکربندی پنل ادمین جنگو برای درخواست ها و ایستگاه ها
- ایجاد تست های واحد

- ایجاد فرایند یکپارچه سازی مستمر و ادغام با تست واحد
- ایجاد بستر ارتباط امن از طریق HTTPS
- انجام تست های بار با استفاده از Jmeter

## فرانت اند:

- شروع فرانت با فریمورک NextJs
- ایجاد صفحه ریشه با یک لینک به صفحه ثبت نام
- ایجاد صفحه ثبت نام، به صورت راست چین
- تغییر خطالرسم به وزیرمتن
- داکرایز شدن فرانت اند و ایجاد فایل Dockerfile
- استفاده از MUI و کلاس های آن برای ایجاد کامپوننت های فرانت
- پیاده سازی بخش کننده
- پیاده سازی صفحه پروفایل
- پیاده سازی فرم درخواست اجرای برنامه
- پیاده سازی لیست ایستگاه ها
- قابلیت تغییر حالت تاریک و روشن در کد
- اتصال فرانت به بکند در لایه API های ایجاد شده
- تغییر صفحه ریشه بر اساس اینکه کاربر وارد شده یا خیر
- ایجاد بستر ارتباط امن از طریق HTTPS
- انجام تست های بار با استفاده از Jmeter

## در ادامه تصاویر مربوط به انجام آزمون های مختلف خدمت شما ارائه میگردد:

### آزمون های API با استفاده از Curl:

```

mohsen@user-HP: ~/University/internet-eng/radio_behesht_backend/tests$ cat *
1 #!/bin/bash
2 #
3 set -x

4 curl -X POST http://localhost:8000/api/auth/users/ \
5 -H "Content-Type: application/json" \
6 -d '{"username": "testuser", "password": "V4q57giibABXzx2S"}'

7 echo

8 export TOKEN=$(curl -X POST http://localhost:8000/api/auth/jwt/create \
9 -H "Content-Type: application/json" \
10 -d '{"username": "testuser", "password": "V4q57giibABXzx2S"}' | jq
-r '.access')

11 echo
12 export REFRESH_TOKEN=$(curl -X POST http://localhost:8000/api/auth/jw
t/create/ \
13 -H "Content-Type: application/json" \
14 -d '{"username": "testuser", "password": "V4q57giibABXzx2S"}' | jq
-r '.refresh')

15 echo

16 curl -X POST http://localhost:8000/api/auth/jwt/verify/ \
17 -H "Content-Type: application/json" \
18 -d '{"token": "${TOKEN}"}'

19 echo
20 curl -X POST http://localhost:8000/api/auth/jwt/refresh/ \
21 -H "Content-Type: application/json" \
22 -d '{"refresh": "${REFRESH_TOKEN}"}'

14 -d '{"username": "testuser", "password": "V4q57giibABXzx2S"}' | jq
-r '.refresh'

15 echo

16 curl -X POST http://localhost:8000/api/auth/jwt/verify/ \
17 -H "Content-Type: application/json" \
18 -d '{"token": "${TOKEN}"}'

19 echo
20 curl -X POST http://localhost:8000/api/auth/jwt/refresh/ \
21 -H "Content-Type: application/json" \
22 -d '{"refresh": "${REFRESH_TOKEN}"}'

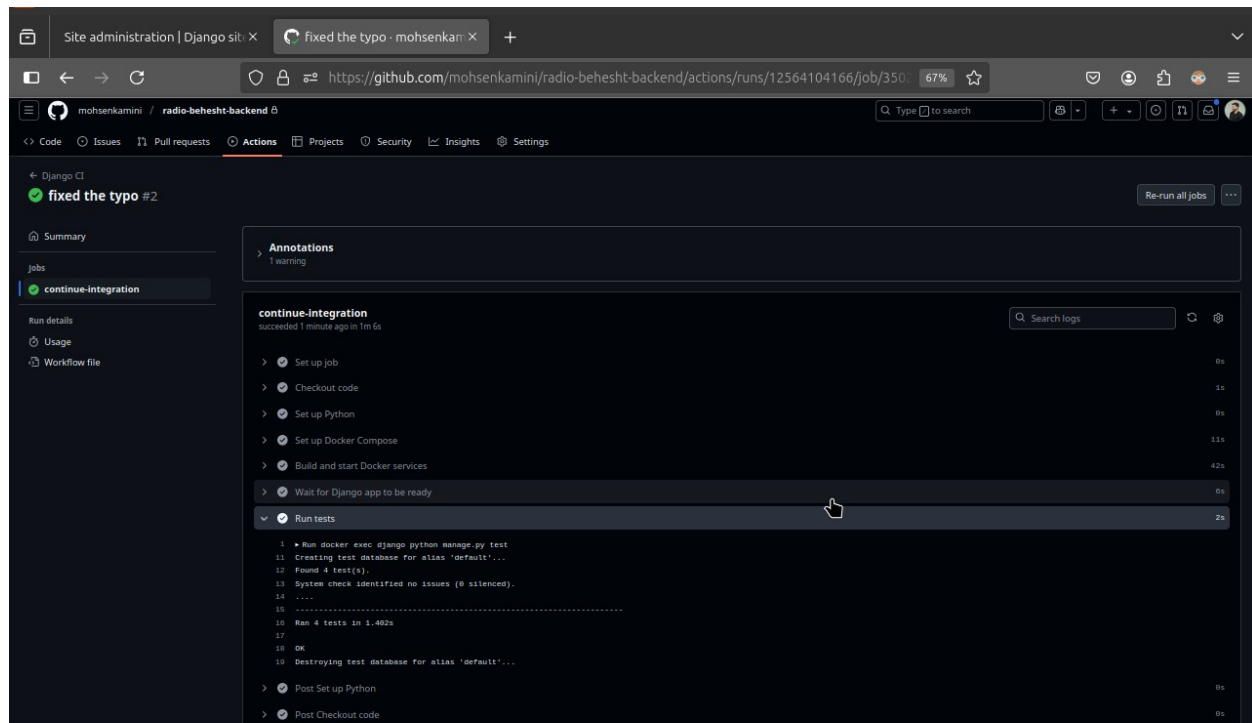
23 echo
24 #!/bin/bash
25 #
26 set -x

27 TOKEN=$1
28 curl -X GET http://localhost:8000/api/users/ \
29 -H "Authorization: Bearer $TOKEN"
30 curl http://localhost:8000/api/radio/stations/
31 #!/bin/bash
32 set -x
33 TOKEN=$1

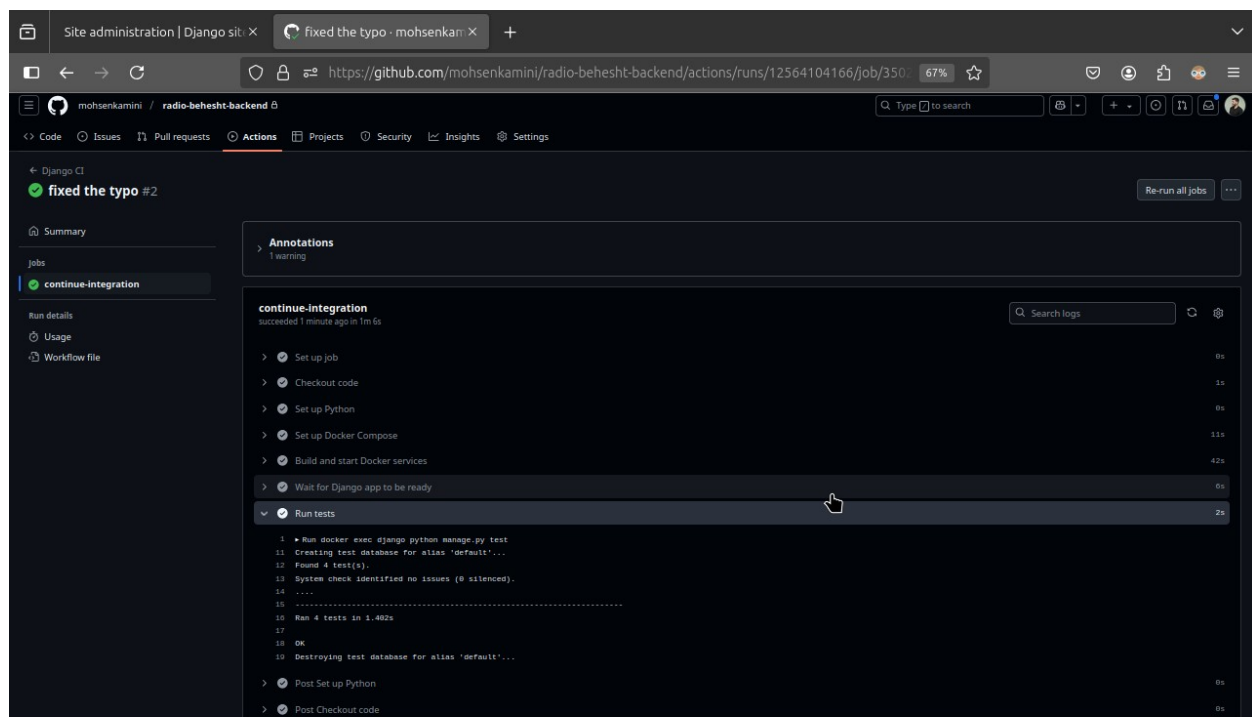
34 curl -X POST http://127.0.0.1:8000/api/stream/request \
35 -H "Content-Type: application/json" \
36 -H "Authorization: Bearer $TOKEN" \
37 -d '{
38   "program name": "My Program",
39   "description": "This is a test program.",
40   "schedule": "Every Thursday",
41   "start date": "2024-01-01",
42   "episodes": 10,
43   "contact": "example@example.com"
44 }'
45 "

```

## انجام آزمون واحد در فرایند CI:



## انجام آزمون عملکرد سرور و دریافت پاسخ اولیه در فرایند CI:



## ادغام با SonarQube

The screenshot shows the SonarQube Cloud interface for the 'Mohsen Karbalaei Amini' organization. The top navigation bar includes 'My Projects', 'My Issues', 'Explore', and a search icon. The user profile 'Mohsen Karbalaei Amini' is visible, along with a 'Free plan' badge and a LinkedIn link. The main content area displays a list of projects. Two projects are shown: 'radio-behesht-backend' and 'radio-behesht-frontend'. Both projects are marked as 'Passed' with a green checkmark. The 'radio-behesht-backend' project shows a last analysis on 1/1/2025 at 12:56 AM, with 486 Lines of Code, Python, and Docker. The 'radio-behesht-frontend' project shows a last analysis on 12/29/2024 at 4:14 PM, with 749 Lines of Code, JavaScript, and Docker. The left sidebar contains filters for Quality Gate (Passed: 2, Failed: 2), Reliability (A: 3, B: 1, C: 0, D: 0, E: 0), and Security. The bottom status bar shows the time as 01:37 on 2025-01-01.

The screenshot shows the 'Overview' page for the 'radio-behesht-backend' project in the SonarQube Cloud interface. The top navigation bar includes 'My Projects', 'My Issues', 'Explore', and a search icon. The user profile 'Mohsen Karbalaei Amini' is visible, along with a 'Free plan' badge and a LinkedIn link. The main content area displays the project overview. The project name 'radio-behesht-backend' is shown with a green 'R' icon. The last analysis is on Jan 01, 2025, with 486 Lines of Code. The project is marked as 'Passed' with a green checkmark. The 'Main Branch Status' section shows a 'Passed' status with a green checkmark and the text 'Enjoy your sparkling clean code!'. The 'Main Branch Evolution since 10 days ago' section shows '3 Issues' and tabs for 'Issues', 'Coverage', and 'Duplications'. The bottom status bar shows the time as 01:40 on 2025-01-01.

## انجام آزمون بار با استفاده از Jmeter و نتایج آن:

View Results in Table.jmx (/home/mohsen/University/internet-eng/View Results in Table.jmx) - Apache JMeter (2.13.19691222)

File Edit Search Run Options Help

Test Plan  
Thread Group  
HTTP Request GET /  
HTTP Request POST /api/auth/users/  
View Results in Table  
Aggregate Graph  
Summary Report  
Response Time Graph  
WorkBench

### Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only: ☐ Errors ☐ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request ...	100	6695	661	10279	2912.05	0.00%	4.9/sec	201.86	41852.0
HTTP Request ...	100	56	4	175	34.01	0.00%	5.1/sec	1.15	231.0
TOTAL	200	3376	4	10279	3906.28	0.00%	9.9/sec	202.72	21041.5

☐ Include group name in label?  ☒ Save Table Header

View Results in Table.jmx (/home/mohsen/University/internet-eng/View Results in Table.jmx) - Apache JMeter (2.13.19691222)

File Edit Search Run Options Help

Test Plan  
Thread Group  
HTTP Request GET /  
HTTP Request POST /api/auth/users/  
View Results in Table  
WorkBench

### View Results in Table

Name: View Results in Table

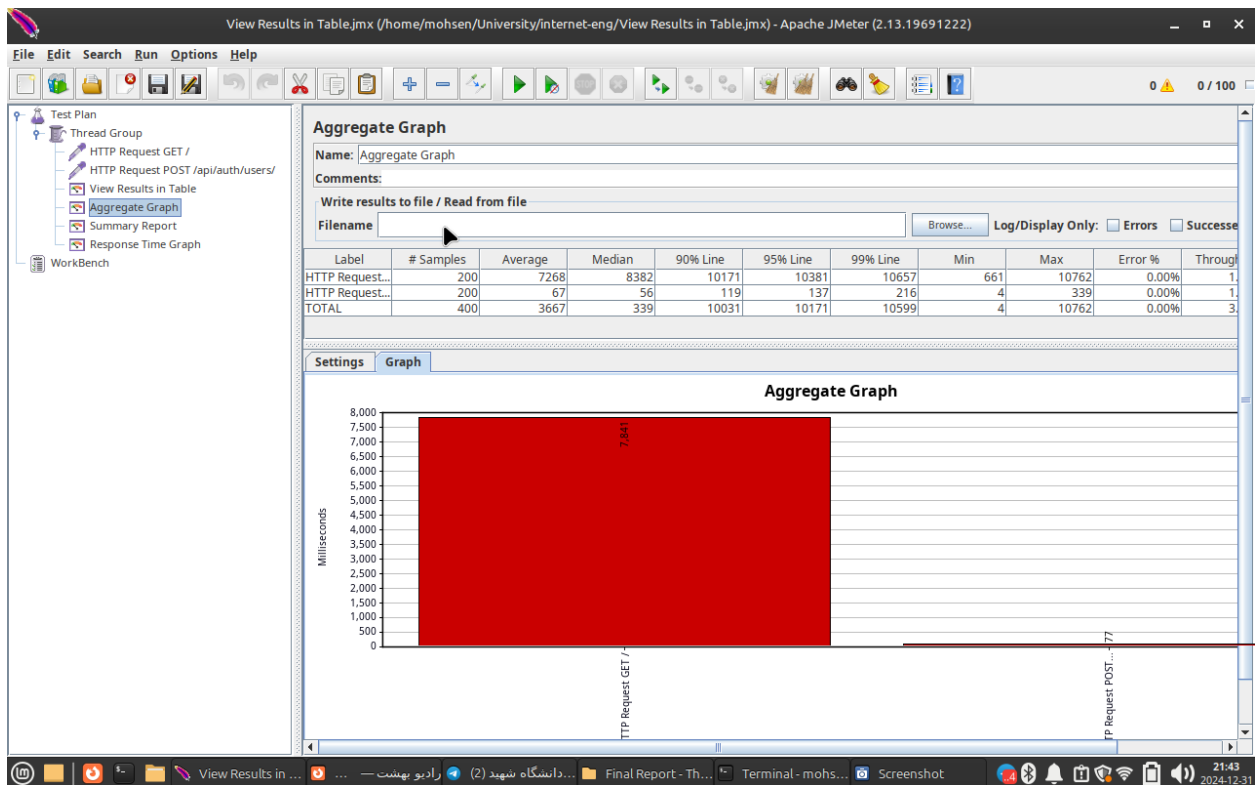
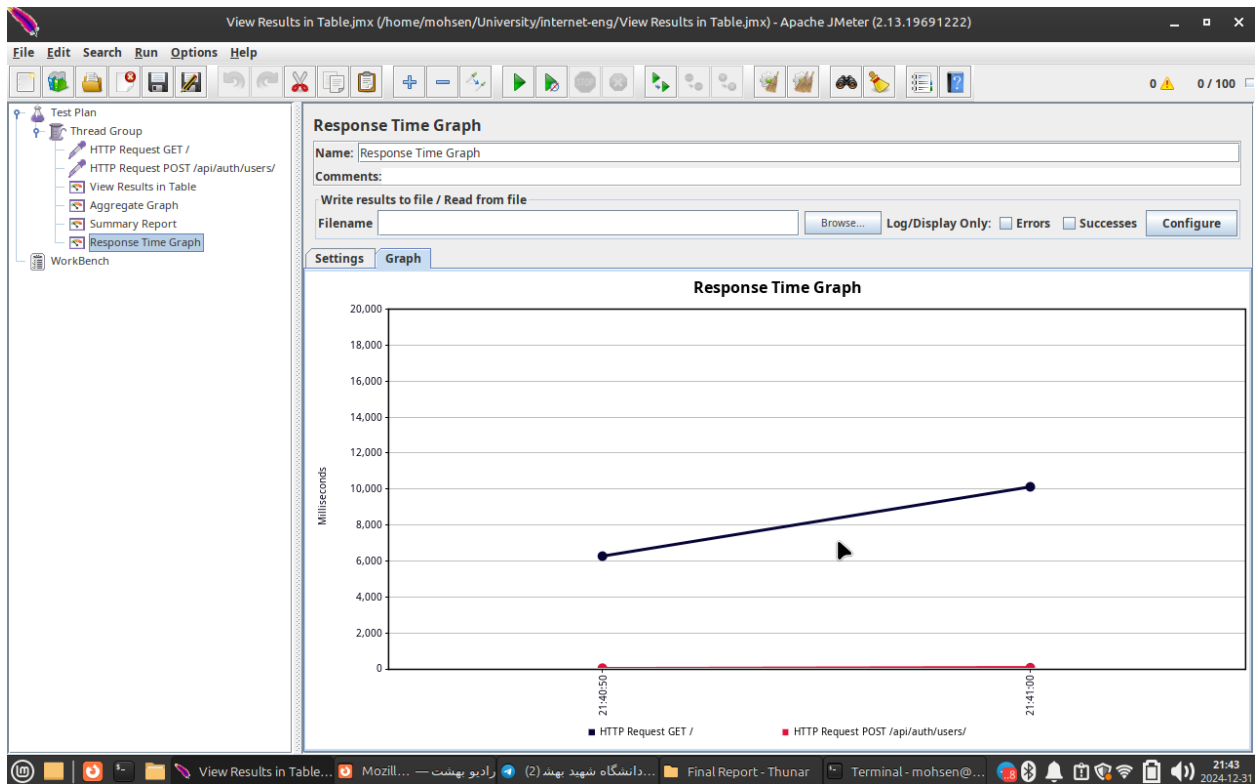
Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only: ☐ Errors ☐ Successes

Sample #	Start Time	Thread ...	Label	Sample Time(...)	Status	Bytes	Latency	Connect Time(...)
334	21:37:06.249	Thread ...	HTTP Request GET /	14381	🟢	41852	14200	20
335	21:37:06.149	Thread ...	HTTP Request GET /	14496	🟢	41852	14369	18
336	21:37:06.455	Thread ...	HTTP Request GET /	14206	🟢	41852	14070	12
337	21:37:06.034	Thread ...	HTTP Request GET /	14641	🟢	41852	14494	4
338	21:37:05.930	Thread ...	HTTP Request GET /	14760	🟢	41852	14605	23
339	21:37:07.064	Thread ...	HTTP Request GET /	13640	🟢	41852	13478	6
340	21:37:20.727	Thread ...	HTTP Request POST /api/auth/use...	114	🟢	231	114	0
341	21:37:20.731	Thread ...	HTTP Request POST /api/auth/use...	120	🟢	231	119	0
342	21:37:20.725	Thread ...	HTTP Request POST /api/auth/use...	126	🟢	231	126	0
343	21:37:20.585	Thread ...	HTTP Request POST /api/auth/use...	276	🟢	231	267	0
344	21:37:20.750	Thread ...	HTTP Request POST /api/auth/use...	111	🟢	231	111	0
345	21:37:20.830	Thread ...	HTTP Request POST /api/auth/use...	121	🟢	231	121	0
346	21:37:20.835	Thread ...	HTTP Request POST /api/auth/use...	149	🟢	231	149	0
347	21:37:20.833	Thread ...	HTTP Request POST /api/auth/use...	152	🟢	231	152	0
348	21:37:20.842	Thread ...	HTTP Request POST /api/auth/use...	145	🟢	231	145	0
349	21:37:07.168	Thread ...	HTTP Request GET /	13825	🟢	41852	13691	10
350	21:37:07.268	Thread ...	HTTP Request GET /	13855	🟢	41852	13837	6
351	21:37:21.107	Thread ...	HTTP Request POST /api/auth/use...	97	🟢	231	97	0
352	21:37:21.238	Thread ...	HTTP Request POST /api/auth/use...	38	🟢	231	38	0
353	21:37:07.476	Thread ...	HTTP Request GET /	13820	🟢	41852	13801	4
354	21:37:21.404	Thread ...	HTTP Request POST /api/auth/use...	131	🟢	231	131	0
355	21:37:07.381	Thread ...	HTTP Request GET /	14239	🟢	41852	14158	7
356	21:37:07.586	Thread ...	HTTP Request GET /	14089	🟢	41852	13959	12
357	21:37:21.634	Thread ...	HTTP Request POST /api/auth/use...	216	🟢	231	216	0
358	21:37:21.726	Thread ...	HTTP Request POST /api/auth/use...	126	🟢	231	126	0
359	21:37:07.792	Thread ...	HTTP Request GET /	14092	🟢	41852	14066	15
360	21:37:07.896	Thread ...	HTTP Request GET /	14010	🟢	41852	13967	11
361	21:37:07.681	Thread ...	HTTP Request GET /	14247	🟢	41852	14186	9

☐ Scroll automatically? ☐ Child samples? No of Samples 400 Latest Sample 12 Average 3015 Deviation 5335





## برقراری ارتباط امن از طریق HTTPS در فرانت و بکند:

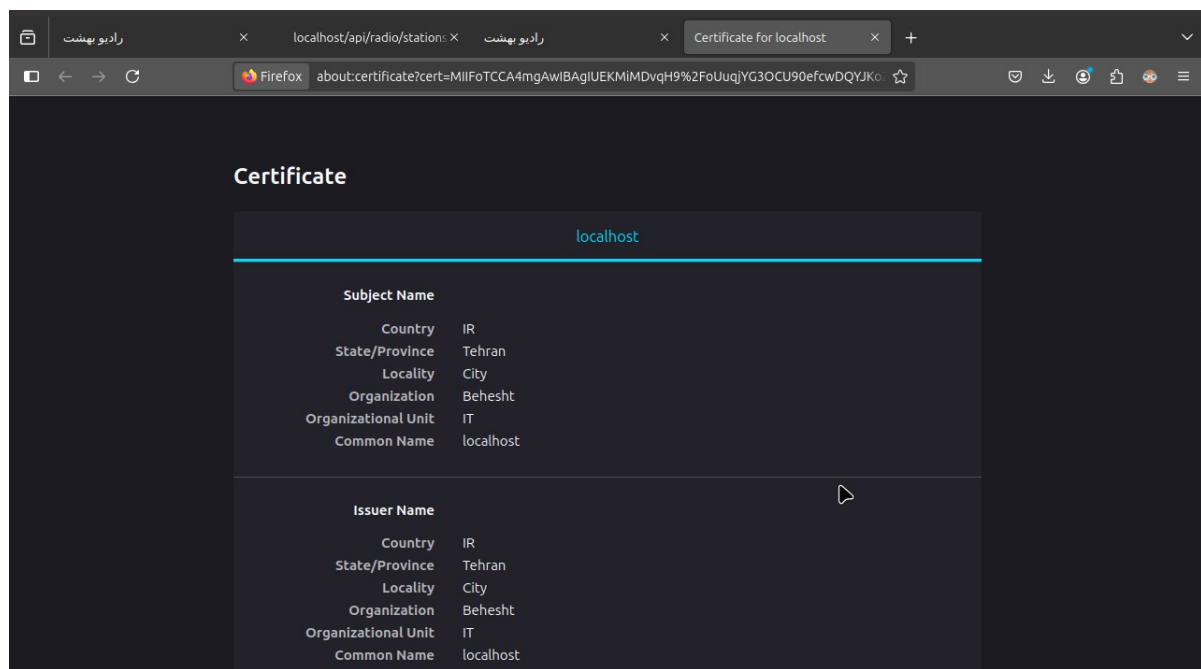
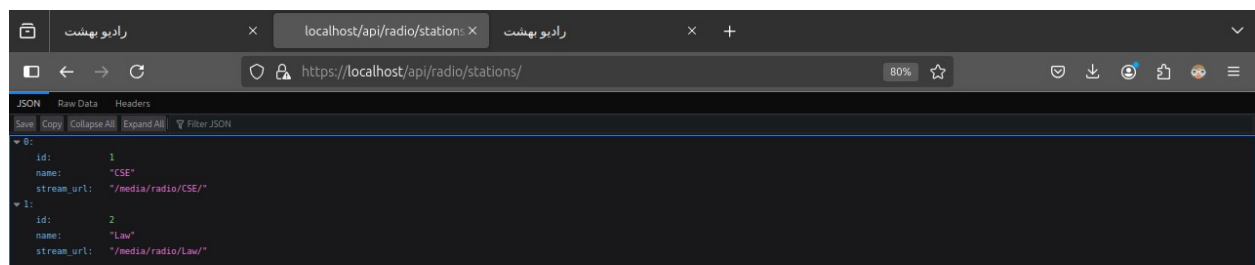


### ورود به رادیو بهشت

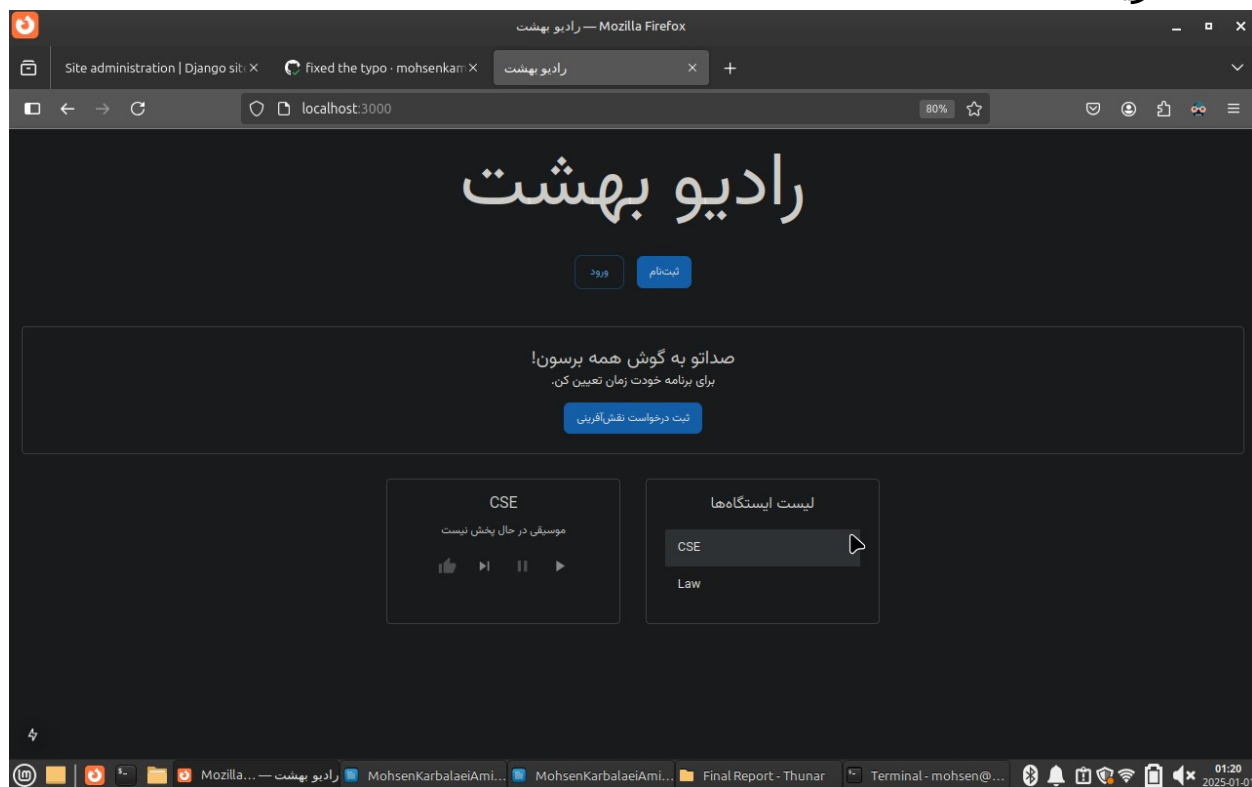
نام کاربری

رمز عبور

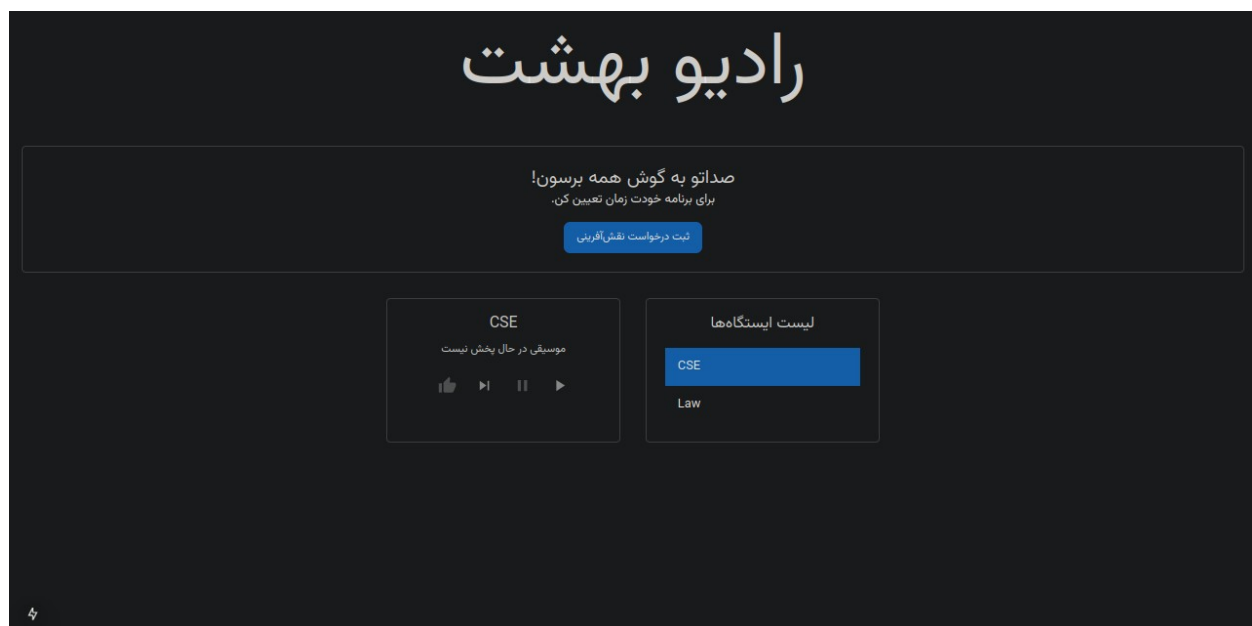
ورود



## و نهایتا تصاویری از محیط سامانه خدمت شما ارائه می‌گردد: صفحه ریشه:



صفحه ریشه بعد از ورود کاربر به همراه پخش‌کننده و پنجره درخواست برای اجرا:



## صفحه فرم درخواست اجرای برنامه:

درخواست نقش آفرینی و اجرای برنامه در رادیو بهشت

\* نام برنامه

\* توضیحات

\* برنامه زمان بندی

\* تاریخ شروع mm / dd / yyyy

\* تعداد قسمت ها

\* اطلاعات تماس

ارسال درخواست

## صفحه پروفایل:

پروفایل کاربری

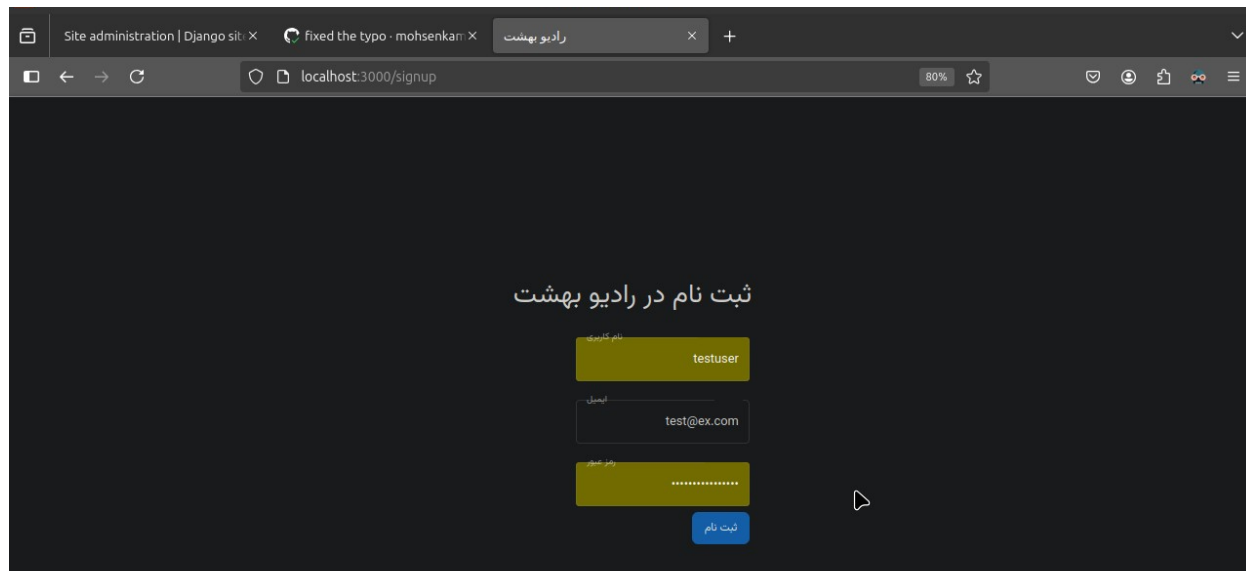
نام کاربری: testuser

ایمیل: اطلاعاتی وارد نشده است.

بیوگرافی: اطلاعاتی وارد نشده است.

تاریخ تولد: اطلاعاتی وارد نشده است.

## صفحه ثبت نام:



ثبت نام در رادیو بهشت

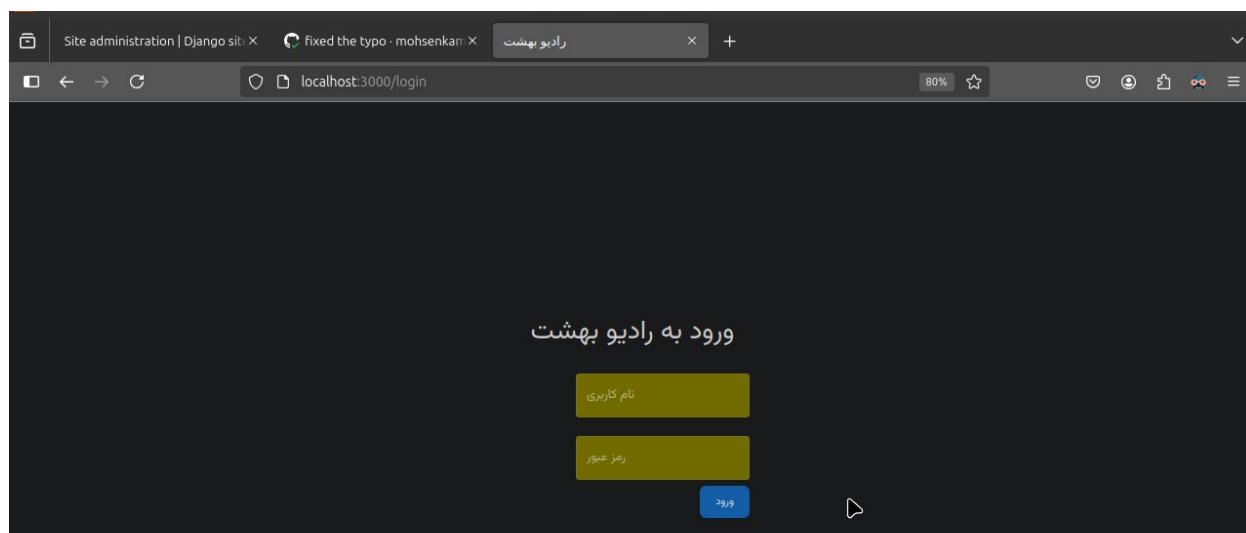
نام کاربری: testuser

ایمیل: test@ex.com

رمز عبور: .....

ثبت نام

## صفحه ورود:



ورود به رادیو بهشت

نام کاربری

رمز عبور

ورود

## مستندات تحویل شده:

- دو سطح اول از مستند معماری محصول بر اساس مدل C4
- تکمیل مراحل اولیه سند گوگل داک برای توصیف پروژه
- رعایت رسم و رسومات کامیت در گیت
- تحویل دو سری گزارش پیشرفت پروژه
- مستند گزارش نهایی (این سند)

## روند یادگیری فناوری‌ها:

در ابتدا یادگیری بکند با انجام دو پروژه تستی روی جنگو انجام شد. در ابتدا یک پروژه تستی پخش‌کننده موسیقی که از S3 موسیقی را دریافت و پخش میکرد را پیاده کردم. این پروژه به صورت تمام جنگو بود و جدای از بکند پیاده شده، فرانت هم با استفاده از تمپلیت‌های HTML و CSS روی خود جنگو مدیریت شد. برای انجام این کار از یک پلی‌لیست در یوتیوب، مستندات اولیه و مثال‌های جنگو به همراه هوش مصنوعی مولد استفاده کردم تا دید خوبی از فضای فریمورک جنگو بگیرم و همچنین دید اولیه نسبت به فرانت رو تکمیل‌تر کنم.

سپس، برای پروژه اصلی، کمی در مورد Django Rest Framework از مستندات اینترنت خواندم و یکی دو ویدیو هم در خصوص ساختار کد در DRF دیدم تا با ساختار مدل‌ها، ویوها، روتینگ و سربالایزشن آشنا بشم.

در خصوص یادگیری در استانداردها و API Design در درس مهندسی نرم‌افزار بخش‌هایی رو داشتیم.

در خصوص یادگیری داکر و مباحث دواپسی، حدوداً ۵ سال سابقه کاری داشتم به عنوان مهندس دواپس که در طول این مدت، با انجام کارهای عملی در محیط‌های پروداکشن، خواندن تجارب برتر (best practices)، بنچمارک‌ها و ... این دانش رو کسب کردم و در این پروژه بکار بردم.

## مراجع یادگیری:

- <https://nextjs.org/docs/app/getting-started/project-structure>
- <https://medium.com/@eyuthedev/authenticating-your-django-rest-framework-with-djoser-9ce5cb370d7>

[https://www.youtube.com/watch?v=qcJZN1pvG6A&list=PL0Zuz27SZ-6NamGNr7dEqzNFEcZ\\_FAUVX](https://www.youtube.com/watch?v=qcJZN1pvG6A&list=PL0Zuz27SZ-6NamGNr7dEqzNFEcZ_FAUVX) •  
ChatGPT •  
stackoverflow •