**Step-by-Step Demo (With Actions & Proof)**
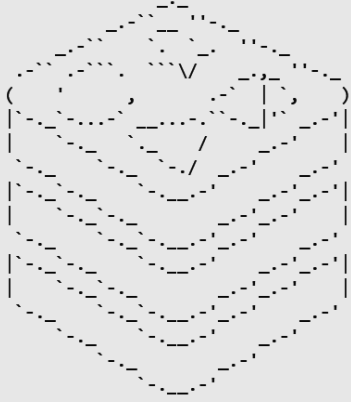**STEP 1: Run All Required Services**
**A. Run Redis**
In a terminal:
**redis-server**

```
PS D:\django-project> redis-server
[11620] 25 Jun 14:24:03.304 # Warning: no config file specified, using the default config.
 In order to specify a config file use D:\Redis\redis-server.exe /path/to/redis.conf
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 3.0.504 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )      Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|      Port: 6379
 |    `-._   `._    /     _.-'    |      PID: 11620
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

[11620] 25 Jun 14:24:03.317 # Server started, Redis version 3.0.504
[11620] 25 Jun 14:24:03.317 * The server is now ready to accept connections on port 6379
```

**Screenshot: redis_running.png**

**B. Run Celery Worker**
In a second terminal:
**celery -A internship_project worker --loglevel=info**

```
PS D:\django-project> celery -A internship_project worker --loglevel=info

 -------------- celery@AlimsComputer v5.5.3 (immunity)
--- ***** -----
-- ******* ---- Windows-10-10.0.22621-SP0 2025-06-25 14:25:24
- *** --- * ---
- ** ---------- [config]
- ** ---------- .> app:         internship_project:0x1d0434d7f70
- ** ---------- .> transport:   redis://localhost:6379/0
- ** ---------- .> results:     disabled://
- *** --- * --- .> concurrency: 8 (prefork)
-- ******* ---- .> task events: OFF (enable -E to monitor tasks in this worker)
--- ***** -----
 -------------- [queues]
                .> celery           exchange=celery(direct) key=celery


[tasks]
  . api.tasks.send_email_task

[2025-06-25 14:25:24,360: INFO/MainProcess] Connected to redis://localhost:6379/0
[2025-06-25 14:25:24,360: INFO/MainProcess] mingle: searching for neighbors
[2025-06-25 14:25:25,039: INFO/SpawnPoolWorker-3] child process 19788 calling self.run()
[2025-06-25 14:25:25,039: INFO/SpawnPoolWorker-2] child process 9872 calling self.run()
[2025-06-25 14:25:25,055: INFO/SpawnPoolWorker-5] child process 35936 calling self.run()
[2025-06-25 14:25:25,072: INFO/SpawnPoolWorker-1] child process 36268 calling self.run()
[2025-06-25 14:25:25,099: INFO/SpawnPoolWorker-4] child process 21012 calling self.run()
[2025-06-25 14:25:25,125: INFO/SpawnPoolWorker-7] child process 28812 calling self.run()
[2025-06-25 14:25:25,142: INFO/SpawnPoolWorker-8] child process 35548 calling self.run()
[2025-06-25 14:25:25,142: INFO/SpawnPoolWorker-6] child process 14608 calling self.run()
[2025-06-25 14:25:25,387: INFO/MainProcess] mingle: all alone
[2025-06-25 14:25:25,434: INFO/MainProcess] celery@AlimsComputer ready.
```

**Screenshot: celery_worker_started.png**

## C. Run Django Server

In a third terminal:

**python manage.py runserver**

```
PS D:\django-project> venv\Scripts\Activate
(venv) PS D:\django-project> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 25, 2025 - 14:26:01
Django version 4.2.23, using settings 'internship_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

**Screenshot: django_server_running.png**

## STEP 2: Login & Access Endpoints
## A. Get JWT Token (Login)

POST http://127.0.0.1:8000/api/token/
Content-Type: application/json

Body:
{
  "username": "your_username",

"password": "your_password"
}

```
1   HTTP/1.1 200 OK
2   Date: Wed, 25 Jun 2025 09:03:04 GMT
3   Server: WSGIServer/0.2 CPython/3.8.6
4   Content-Type: application/json
5   Vary: Accept
6   Allow: POST, OPTIONS
7   X-Frame-Options: DENY
8   Content-Length: 483
9   X-Content-Type-Options: nosniff
10  Referrer-Policy: same-origin
11  Cross-Origin-Opener-Policy: same-origin
12
13 ∨{
14    "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV
      4cCI6MTc1MDkyODU4NCwiaWF0IjoxNzUwODQyMTg0LCJqdGkiOiI2MTk1M2QxMjcxOWI0MWMwODk4ZWY5MmY5N
      2NmY2ZlOCIsInVzZXJfaWQiOjF9.dcPFbRMeTpdNI2B5-ga3Tyep7PS0klRC1RPwtlt1y04",
15    "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhw
      IjoxNzUwODQyNDg0LCJpYXQiOjE3NTA4NDIxODQsImp0aSI6Ijg3YTc2NGViOGQ2ZjQ1YzdhYzg4YzQxZjg0NT
      UwZjljIiwidXNlcl9pZCI6MX0.A-UXH3pBXCtt_bh9Yuqg236LJq_053opg__hSTnFm-E"
16  }
```

**Screenshot: jwt_token_response.png**

**B. Call /api/public/ (No Auth Needed)**
GET http://127.0.0.1:8000/api/public/
Response:
{
  "message": "This is a public endpoint"
}

```
1   HTTP/1.1 200 OK
2   Date: Wed, 25 Jun 2025 09:04:25 GMT
3   Server: WSGIServer/0.2 CPython/3.8.6
4   Content-Type: application/json
5   Vary: Accept
6   Allow: OPTIONS, GET
7   X-Frame-Options: DENY
8   Content-Length: 39
9   X-Content-Type-Options: nosniff
10  Referrer-Policy: same-origin
11  Cross-Origin-Opener-Policy: same-origin
12
13 ∨{
14    "message": "This is a public endpoint"
15  }
```

**Screenshot: public_endpoint.png**

**C. Call /api/protected/ (JWT Token Required)**
**Headers:**
Authorization: Bearer your_token_here
**Response:**
{
  "message": "Hello, your_username! This is a protected endpoint."
}

```
1    HTTP/1.1 200 OK
2    Date: Wed, 25 Jun 2025 09:04:38 GMT
3    Server: WSGIServer/0.2 CPython/3.8.6
4    Content-Type: application/json
5    Vary: Accept
6    Allow: OPTIONS, GET
7    X-Frame-Options: DENY
8    Content-Length: 60
9    X-Content-Type-Options: nosniff
10   Referrer-Policy: same-origin
11   Cross-Origin-Opener-Policy: same-origin
12
13  ∨{
14      "message": "Hello wafiyaalim! This is a protected endpoint"
15   }
```

**Screenshot: protected_endpoint.png**

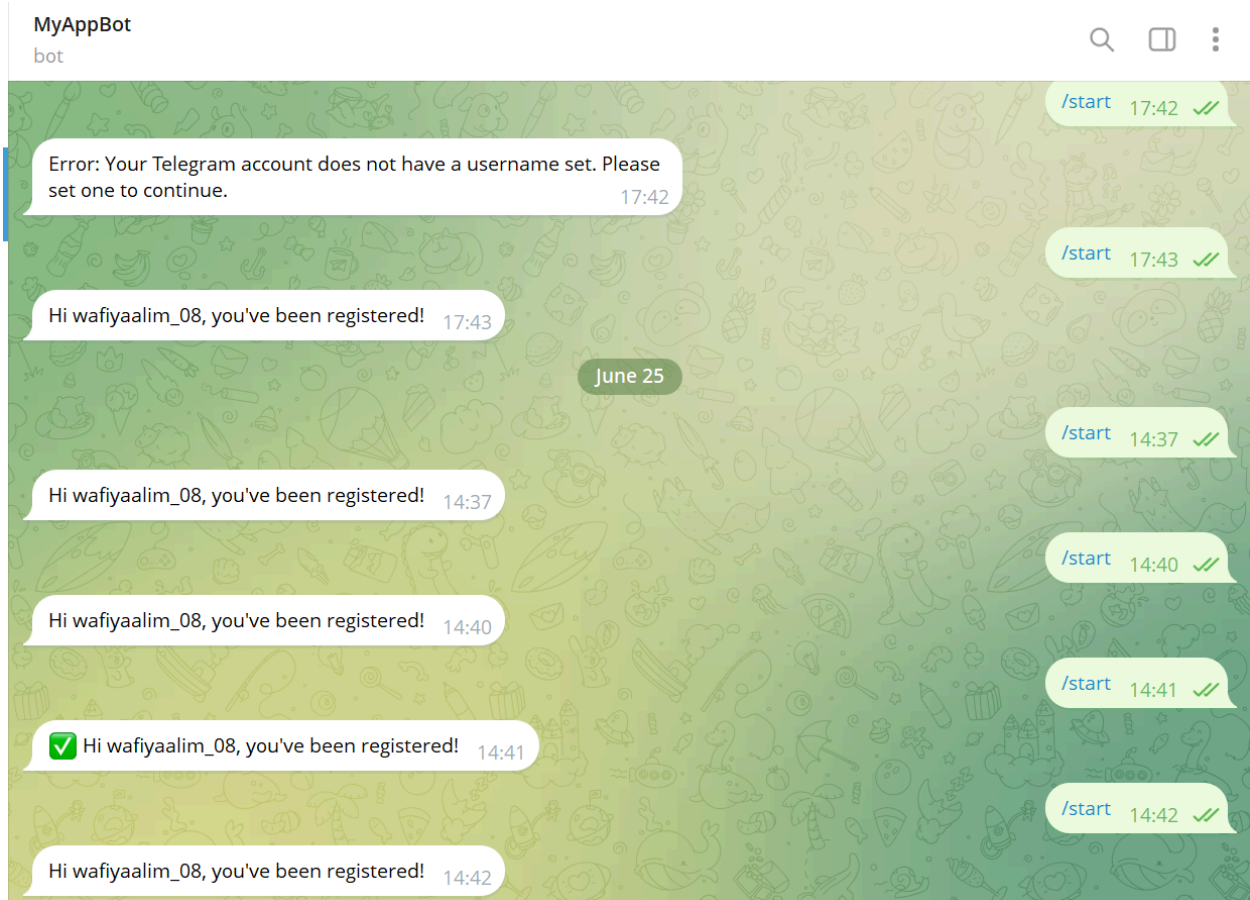**STEP 3: Telegram Bot Demo**
**A. Run the Bot**
python bot.py

```
(venv) PS D:\django-project> python bot.py
Starting Telegram bot...
Bot is now running. Send /start in Telegram.
```

**Screenshot: telegram_bot_running.png**
B. Open Telegram and Type /start
Your bot should respond with:
Hi your_username, you've been registered!

**Screenshot: telegram_start_reply.png**

**STEP 4: Verify Telegram Username Is Stored**
**A. Check Django Admin or Shell**
python manage.py shell
from api.models import TelegramUser
TelegramUser.objects.all()
**Output:**
<QuerySet [<TelegramUser: your_username>]>

```
(venv) PS D:\django-project> python manage.py shell
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)]
in32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from api.models import TelegramUser
>>> TelegramUser.objects.all()
<QuerySet [<TelegramUser: wafiyaalim_08>]>
```

**Screenshot: telegram_db_saved.png**

**STEP 6: Access /api/telegram-users/ (Protected API)**
**Make this request:**
GET http://127.0.0.1:8000/api/telegram-users/
Authorization: Bearer your_token_here
Response:
[
  {
    "username": "your_username",
    "created_at": "2025-06-21T12:43:56.849Z"
  }
]

```
1   HTTP/1.1 200 OK
2   Date: Wed, 25 Jun 2025 09:07:20 GMT
3   Server: WSGIServer/0.2 CPython/3.8.6
4   Content-Type: application/json
5   Vary: Accept
6   Allow: OPTIONS, GET
7   X-Frame-Options: DENY
8   Content-Length: 73
9   X-Content-Type-Options: nosniff
10  Referrer-Policy: same-origin
11  Cross-Origin-Opener-Policy: same-origin
12
13 ∨ [
14 ∨   {
15       "username": "wafiyaalim_08",
16       "created_at": "2025-06-21T12:13:36.624232Z"
17     }
18   ]
```

**Screenshot: telegram_users_api.png**