

Table of Contents

API END POINTS.....	1
API TESTING (RIGHT DATA).....	1
I. REGISTER NEW USER (ROLE USER).....	1
II. REGISTER NEW ADMIN (ROLE ADMIN).....	2
III. LOGIN ADMIN / USER (ADMIN) OAUTH	3
IV. GET USER (AUTH USER).....	4
V. REFRESH TOKEN	4
VI. GET ALL POSTS DETAILS	6
VII. GET POST BY ID {ID}	6
VIII. ADD POST (ONLY ADMIN CAN ADD POSTS)	7
IX. EDIT POST (ONLY ADMIN CAN EDIT POSTS)	7
X. DELETE Post	7
XI. EDIT POST BY USER ROLE	8
XII. LOGOUT.....	8
API TESTING (WRONG DATA).....	9
I. REGISTER USER	9
a. <i>Already register User with wrong role (“admin”, “user”)</i>	9
II. LOGIN WITH WRONG CREDENTIALS	9
III. LOGIN WITHOUT CLIENT_ID AND CLIENT_SECRET	10
IMPLEMENTATION.....	11
1. PASSPORT INSTALLATION & IMPLEMENTATION	11
2. UPDATE AUTHSERVICEPROVIDER	11
3. CHECKUSERROLE.....	12
4. .ENV (FILE CONFIGURATION)	13
SCREENSHOTS FOR API TESTING.....	15

By – Sijan Mahato

API End Points

- a. base_url =><http://127.0.0.1:8000/>
- b. client_id =>019b841a-ae32-71b4-9d68-bb89f466ec3f
- c. client_secret => GN0hy4sMLbTuYVYm8mQMUqy51BYzmrC5V1SciMcj

Method	Endpoint	Description	Auth Required
POST	/api/auth/register	Register new user	No
POST	/api/auth/login	Login (personal token)	No
POST	/api/auth/login/oauth	Login (password grant)	No
POST	/api/auth/refresh	Refresh token	No
GET	/api/auth/user	Get user details	Yes
POST	/api/auth/logout	Logout (revoke token)	Yes
POST	/api/auth/logout-all	Logout all devices	Yes
GET	/api/posts	View all Posts	Yes
GET	/api/posts/{id}	View Posts by id	Yes
POST	/api/posts	Add New Posts	Yes
PUT	/api/posts/{id}	Edit Posts by id	Yes
DELETE	/api/posts/{id}	Delete Posts by id	Yes

API TESTING (RIGHT DATA)

i. Register New User (role user)

```
{  
    "name": "user",  
    "email": "user@gmail.com",  
    "password": "password@123",  
    "password_confirmation": "password@123",  
    "role": "user"  
}
```

Output-

```
{  
    "message": "User registered successfully",  
    "user": {  
        "name": "user",  
        "email": "user@gmail.com",  
        "role": "user",  
        "updated_at": "2026-01-04T03:36:20.000000Z",  
        "created_at": "2026-01-04T03:36:20.000000Z",  
        "id": 3  
    },  
    "token": {  
        "access_token": "GN0hy4sMLbTuYVYm8mQMUqy51BYzmrC5V1SciMcj",  
        "refresh_token": "019b841a-ae32-71b4-9d68-bb89f466ec3f",  
        "exp": 1721431200  
    }  
}
```

```

        "access_token":  

        "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIwMTliODQxYS05MDQxLTcwYjYtOTkzNS0xNmZi  

        ZmM0YWJmZDgiLCJqdGkiOiIyNDBmMDcxOTc0NWU20GQ0Y2QyNmE2M2ZhM2QzMID0YzBiYWI30TdkZTQ5Y2RmMD  

        kyM2EZY2U2ZWQ1NWFiYjJmYT0NDfHzTEyYmViYWEzSIsImlhCI6MTc2NzQ5Nzc4MS4wMjE1MzEsIm5iZiI6  

        MTc2NzQ5Nzc4MS4wMjE1MzIsImV4cCI6MTc4MzEzNjE4MS4wMTU20DcsInN1YiI6IjMiLCJzY29wZXMiOltdfQ  

        .pguSy4oJjp6C13DxqE73MvKQAajiWSG1jzNy-07EWaM-FKeRgw91og6KtanW8PLzyrGUKRnNPjgt-  

        fQePGL_t6sBoUfvUs2iuJW5875sZgpS0BMhOBdyLPdUQ51-  

        L3BozDBVp8q0LCCfAc55hWXz2S7y92yUExufGVJUYr2JMUtSh7JUZYVaH3pFrGuqaQLZ7-  

        gIKdUmsMr962JSYHSqYwp7ioNFsqt-NL6F-6xvjiILe7KVdyABhAgczCa0FCPaDIbWgHxPUT1RG-  

        chcDanBo0tN09hvF-  

        fRpBvZWIEJNVIsGhZJSa1cFn5wTZTaanRo20hBk8lQL2paikJdLPro_LiUtDJstF1wD3d0HRuE6G61-  

        6fgqfNAbDHMTEv8EwA0oH_BcWJk31smDjpxAlgS0Gr9_c0VAHHdcJUWvbjpPQXMuy1-  

        HpsBHEi05znCqDNUAgqhc0TiHGVA6q0FCiGAq3xfEJVji1nWhtftumGTk1EMjBhYT09g1KKPcC1zKm60V51sgf  

        qqIXE_EAOZLh78dHkD-  

        J0zWUcIwvdrsRIyTmZILH_86Rmi0gUB0pXhJU3M2a0q_0YA96sFNBUeUClhovvUMFeBtwVJI7rKuEcFXLzVleU  

        rDci2AabK5QKVwyZzA171iJY3RARQfiWRDh080LvJTGjqFlRF67MC-BrkB4",  

        "token_type": "Bearer",  

        "expires_in": 15638399.967919001  

    }  

}

```

ii. Register New admin (role admin)

```
{
    "name": "admin",
    "email": "admin@gmail.com",
    "password": "password@123",
    "password_confirmation": "password@123",
    "role": "admin"
}
```

Output –

```
{
    "message": "User registered successfully",
    "user": {
        "name": "admin",
        "email": "admin@gmail.com",
        "role": "admin",
        "updated_at": "2026-01-04T03:38:09.000000Z",
        "created_at": "2026-01-04T03:38:09.000000Z",
        "id": 4
    },
    "token": {
        "access_token":  

        "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIwMTliODQxYS05MDQxLTcwYjYtOTkzNS0xNmZi  

        ZmM0YWJmZDgiLCJqdGkiOiIjInjNlNDZiZTI0NjQxNGVmNjBkNjM1MDE1YzgxN2U4ZWY4NDM1NGU2YmVmNWE1Yj
        .pguSy4oJjp6C13DxqE73MvKQAajiWSG1jzNy-07EWaM-FKeRgw91og6KtanW8PLzyrGUKRnNPjgt-  

        fQePGL_t6sBoUfvUs2iuJW5875sZgpS0BMhOBdyLPdUQ51-  

        L3BozDBVp8q0LCCfAc55hWXz2S7y92yUExufGVJUYr2JMUtSh7JUZYVaH3pFrGuqaQLZ7-  

        gIKdUmsMr962JSYHSqYwp7ioNFsqt-NL6F-6xvjiILe7KVdyABhAgczCa0FCPaDIbWgHxPUT1RG-  

        chcDanBo0tN09hvF-  

        fRpBvZWIEJNVIsGhZJSa1cFn5wTZTaanRo20hBk8lQL2paikJdLPro_LiUtDJstF1wD3d0HRuE6G61-  

        6fgqfNAbDHMTEv8EwA0oH_BcWJk31smDjpxAlgS0Gr9_c0VAHHdcJUWvbjpPQXMuy1-  

        HpsBHEi05znCqDNUAgqhc0TiHGVA6q0FCiGAq3xfEJVji1nWhtftumGTk1EMjBhYT09g1KKPcC1zKm60V51sgf  

        qqIXE_EAOZLh78dHkD-  

        J0zWUcIwvdrsRIyTmZILH_86Rmi0gUB0pXhJU3M2a0q_0YA96sFNBUeUClhovvUMFeBtwVJI7rKuEcFXLzVleU  

        rDci2AabK5QKVwyZzA171iJY3RARQfiWRDh080LvJTGjqFlRF67MC-BrkB4",  

        "token_type": "Bearer",  

        "expires_in": 15638399.967919001
    }
}
```

```

kzZTVjYTc5ZjNkMWZlZTI0MmZlNjJjYjk5YzYyNjVjZSIsImlhdCI6MTc2NzQ5Nzg40S4xNjk1MzgsIm5iZiI6
MTc2NzQ5Nzg40S4xNjk1MzksImV4cCI6MTc4MzEzNjI40S4xNjQ5NjUsInN1YiI6IjQiLCJzY29wZXMi0ltdfQ
.F03_UMXeyDtPkWLhD1SX6xj7uU4Tws4uCDmFw2q-RBDUFansVgerRBWZ-tJq2x-
bFgljWJIYnyuwcx18EhSdIHSpBHzY9F3RAaF-
1wSWDIOEN38UUo53KnEdQoyo8eNbe_bCyySvdVYRoNnehPiCbqb3cAZQthAvxGmeEFtYRrb1taqt4qzJgaY5uU
e8WQQSQ0jwj8UUfKV_z2Tip52u0lYkfoLkG3ZjL_V-
PiXwlvCJnQ0aVL6Je9ngr0XDAL7BbhkfkT_tMhIdwGU8WRJh0E4L5ZAk3wGHUpHRQAJpPT5J_8U9jCb0bRxS0s
1a7w5RVpz9KmdrBTinPjkU9Q8V2rMWd8dv6UowQuKaysfbmtC243_00tLyJcR_oSJLLJEdfURjFqJ7yFhDSKX
i0DgmcTl0vVcw2_IW29p51A2LB0GxN_YHNDPb7KNyKnHxKirUgveWm6g1Kirx2Tb2Y13DaIgW5S4tveBeYAvnn
svF5G1Y2Ka_DXsdqQzTbbLuFIJlabIJ0wBNo-
KF7nvcY0IKVT1eSNxVFCr0X6It1r_132PlZh_QC4oPV2dM6_xpoе9wHFbu2n1hJB1v2SKX7fUD1iRmTAGNQmu3
o07e1jtYcPM6DUqRFHh6Qzd_mXIh6_c37b0ZSCtmFSMz20pUR59o1NN3UfyCm7mS5t4QJq1-sA",
    "token_type": "Bearer",
    "expires_in": 15638399.823144
}
}

```

iii. Login admin / user (admin) oauth

```
{
  "email": "sizan@gmail.com",
  "password": "password@123",
  "client_id": "{{client_id}}",
  "client_secret": "{{client_secret}}"
}
```

Output –

```
{
  "message": "Login successful",
  "user": {
    "id": 1,
    "name": "Sizan",
    "email": "sizan@gmail.com",
    "email_verified_at": null,
    "created_at": "2026-01-03T13:45:17.000000Z",
    "updated_at": "2026-01-03T13:45:17.000000Z",
    "role": "admin"
  },
  "token": {
    "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIwMTliODQxYS1hZTMyLTcxYjQtOWQ2OC1iYjg5
ZjQ2NmVjM2YiLCJqdGkiOiJkYTkwOWQyNzcxNDE3N2MwZDk2MzVmMzg5NTgzZTg10GQ0ZGUzNzRiZGY5YzIi
NlOTI2NTMzY2ZjN2UwY2I2NzQ1NDBjNjg1YTI3YTEyMCIsImlhdCI6MTc2NzQ1MTc3Ny4z0DQyMjMsIm5iZiI6
MTc2NzQ1MTc3Ny4z0DQyMjQsImV4cCI6MTc20Dc0Nzc3Ny4zNzkz0DUsInN1YiI6IjEiLCJzY29wZXMi0ltdfQ
.DCs0Q-iVLBwvJ1R2pL4zZtgQps1I9Vm_HGX8RvvL0aSgNer-
hSAWwukIicVV8Gu9FPEhp59ehaUzQEc8wQIpW0xVRbgWQ-
```

```

hsjlQZmLlc32RIL2UPjeS8dptrnjwLgsMU_LYIqVTGv2bFgLVBN_I9gL2cDzhx2JGHxIoPbqS0VznCVSWHD9Lez
ZCvjdNQpPK_8CR9PVkD4WX4pMALH1TSP7LN_mTqAyI2FDW08cRbkXv_Sef0s244U8BKgNmflKuYSspN1G15sQX
jzd8W0z9vnkwRFQgwr0h9_JVy19UqrvJWZnp0Bti8YLNkP0Xd7d0Nqox7i1Hf5JPaEJq7oWp2t_8f2oJY3ceVw
co8Ehd73DXN7C_0xfX0ysC20ozCxdaUnqlf5jzf3VLRa5C37eH-
DjLGakbyfvEiI0GW70p9pm9bI19hP0eUImf1n18hvpNJ04tYQS-3I9HhR8ndWqUP1l61NpGcORiG-
QIrLG1iCi-onLxkgdk2vABBAYK-
BXXixDsW6nhFaEUNzoSHsffWNWB85gJY0y_7CZ7lgXh1DeQ49e1TIDB7Kp7PmbDtpG8JCjHs1SyvbMOHDMDa1k
XxkL0f932DDAJ9IuNiNRn0zAYhGUTiheNsLy2hjkiiWEzbva7A8sDzn8xxL_cWlxctBxrH_CfY5vCwSIoQXnRL
w5w",
    "refresh_token": "def50200e7f6c41ccd118965bc1aa495a1e2d6cae54ce8f8179d35c763b47fa048d52c34fbac2449530a3
18e3a91771d55512ac8e6891b8272cd57db5847bc45254325cac6106d066d2609b541e3a0d229690fec4c5
4a64cc5c47c8e1e41a5dfb577237fc258c608b729927fc4220179e4e399b22487073835be6adf92a21f250
5c7b9f742593ba52327f88c2382d20807f2ed70eb1737e46563de74f9eec3a6598adfbda93d67c94cb2403
33008b48c3cc3709f78e641bcc8b0b2f972070f63306acc865fbc6de5fe88ab441dd8b80849d830479fae
a6839ca427740877b397cde3136ae7be6585b5d27c484cf480244b1f7b40802a3794a189f02c7e09a89f1d
0e693e1a7732ccb424d164b41c85903bea516a02302cf7939587036af9de00f68a48bd46f0ea0e0801a6
0079af4c2d1f746073415ddd5e6c0c00f26026572f5a395b50f01f972a1351cc71e01d6adda472a641b636
5a2b5b5c2dc5e9b5e3b22cdacc082d800e857528f2158c786a7be2df972cdf43b3a5164bed5336733c88bd
233a24a",
    "token_type": "Bearer",
    "expires_in": 1296000
}
}

```

iv. Get User (auth user)

a. Use Bearer Token

Output -

```
{
  "user": {
    "id": 4,
    "name": "admin",
    "email": "admin@gmail.com",
    "email_verified_at": null,
    "created_at": "2026-01-04T03:38:09.000000Z",
    "updated_at": "2026-01-04T03:38:09.000000Z",
    "role": "admin"
  }
}
```

v. Refresh Token

a. Use previous Token (refresh token)

```
{
  "refresh_token": "def502004ff2abad7d7e7cfea93678d4dd72ab3152303d14c58062c2e868b8109bc57c11539c937eabb2d9292890ca3c198d876580973345953ff3c13987a26e196dc38b55441d604ed278c76c2aee85b0607fe8c36d91c786e4a4cf190a9714d31aac2efbe67ead049e7f14a4b319e67d669669afe7228fa5e9948493f6e3f67d76cec4ed44ded1c67ff8cf549e5c4b09b0456fa5401366c96d72c4e44655211d57ad4c5fdb2fb2c353435b95839a782c2d287fb2fb9fe5de5770236bb81eb60c6be0865299a1b1c9fcf552a6064b94e16537a0437ff5c31f6b10ece496e6cb732d72e080ad52a0232e85aa4c9eb0e1af2b17dc01a44d2f8d04fd86ffae5e730e7f124602272deab2ba597ce4b660ce4313000f1b8c591bbf9888e3c938764c8fdeda0a4326adf12817243e188a63107db39292181ea3b3d9ff69afe6c18ba56b022ebf03b48475f0d3b335e5f3721af0b993dda8b1bdfaadee95811d684514ee9cf3d7a4789d73171443dae5f30ae65e4218919f1d250c3fb7e871543838593be942",
  "client_id": "{{client_id}}",
  "client_secret": "{{client_secret}}"
}
```

Output –

```
{
  "token_type": "Bearer",
  "expires_in": 1296000,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiIwMTliODQxYS1hZTMyLTcxYjQtOWQ20C1iYjg5ZjQ2NmVjM2YiLCJqdGkiOiJmNDQyZW0ZGRjMGV1NDh1ZDdmNWE2ZDUxMjhjZmZhNDIyNDk2Mjg3OGY1MDc0NTFlMjQzYzkwn2IzTZh0TLjZGM2ZTk1MjExOTUyMmY3YSIsImhlhdCI6MTc2NzQ50Dk1My4w0DE50SwibmJmIjoxNzY3NDk40TUzLjA4MTk5MSwiZXhwIjoxNzY4Nzk00TUzLjA30TQyNywic3ViIjoiNCIsInNjb3BlcyI6W119.oeh-p_sfNykKkgo98JW038CaXp2B3jpFiB6w_73nNdYNsKAgg1SN2dNRBA9jn5g54P_Q1c42Zlm1tuiCcou0AlepS9qRiN0mIBik_UUwDdbaJ4kpEKpMiyoapSgyirekkG4d14TWOMK0BqRLutUb3wJ-cLFWdRB409G0hAc7bfuklr1a_W9t1lwKMF22kuRYnJ-bRd51vbv5BFTTUFMRb9dBOUJvZKDnu1wM3QukwEPQUchVm4ryULb6210_-15NncXn1fu9EXlhx1B5-X-wY8uu5wHgvUvgDqXew3iZAphAwixorACQXVl0hWkqY50tbyG3Y9e9J6NZU2vyxks1RDmbSoD6It8QKu7QSYrRbF607k-Rj4Mz3_fj8ebRpRmu63z1J_eXE6bL3IX_A_0JHMgrUkDmqSxIFZUGXKQKdHJN1ETc6DQeEUNlnPc3y96kUwV9z15YoNUquwHMBvEkoE-2q-J6INpMFrstK0pSzYz3-0CLygmPwPrteNqT4wc11PkfQ-p2nDh_NZhHCNr5ladwlyG5ij7HN0JbdZPTETHU5e3Yir6bBpaBBoennGp-65n100dZNIJwaXMqKW0sG86s67epbDCyJKOTEfVG2qhtVNpEM97jQDTg05Z99Lm6YNoqc1QHqi7eNDs4M6UQRme-xd5enEv5pjnuNwh0",
  "refresh_token": "def5020034ce82048a31929103e77af19ad3bbad8e5173593bc1f512cc89cd795a1dbaf7d7bb8fcfa0f5aedbd30e02c87575a55d2418b5957976edfc0fde6d52f4f348288ca3882954b4d60e9d8e5c831c233f64fef8fb51c495abe05dd87d9ec15dad6a2c0ae7fb79839678d0a8ac5ce96d9fbb355907eafcc6e1d82aceff27d03713c29686db2fad8ce11ec530b57d66004bb49090f77729cef54a707b40b92c58e4e503782d59d9ab9ac58f364e341173fc6a80b192ecd1dbc21f17a7e5b41cb0a4574b3bb6bd93a7c8b1fc87bffd30e8176f8a8e49194f41830d2072fe9dcee50203669da1fff95a5bc917a57b2236fd237c1d051d2718f145ef4f2d18c2a2f32590c5efb68b41576d578cb7a2ca38801e002d33f0df378135f9de50d5428d185e17acdfb844fb84543f1578b97f2fb618a19f286558e7854a7f73d476db23128817a313693b25502485d703be1c39900e46c9a975c1b1bb1cb7ebcb463eae49bd3a5409b14416c3cff885448da34c0fd8c53eb91d0b3704f6d60ab9cf1bf830d61"
```

vi. Get All posts details

- a. User Bearer Token

```
[  
  {  
    "id": 1,  
    "user_id": 1,  
    "title": "Updated Post Title",  
    "content": "Updated content of the post",  
    "created_at": "2026-01-03T13:51:12.000000Z",  
    "updated_at": "2026-01-03T13:53:25.000000Z",  
    "user": {  
      "id": 1,  
      "name": "Sizan",  
      "email": "sizan@gmail.com",  
      "email_verified_at": null,  
      "created_at": "2026-01-03T13:45:17.000000Z",  
      "updated_at": "2026-01-03T13:45:17.000000Z",  
      "role": "admin"  
    }  
  }  
]
```

vii. Get Post by id {id}

- a. Use Bearer Token for this as well

```
{  
  "id": 1,  
  "user_id": 1,  
  "title": "Updated Post Title",  
  "content": "Updated content of the post",  
  "created_at": "2026-01-03T13:51:12.000000Z",  
  "updated_at": "2026-01-03T13:53:25.000000Z",  
  "user": {  
    "id": 1,  
    "name": "Sizan",  
    "email": "sizan@gmail.com",  
    "email_verified_at": null,  
    "created_at": "2026-01-03T13:45:17.000000Z",  
    "updated_at": "2026-01-03T13:45:17.000000Z",  
    "role": "admin"  
  }  
}
```

viii. Add Post (only admin can add posts)

- a. Use Bearer Token

```
{  
  "title": "My New Post",  
  "content": "This is the content of my post"  
}
```

Output –

```
{  
  "user_id": 4,  
  "title": "My New Post",  
  "content": "This is the content of my post",  
  "updated_at": "2026-01-04T04:00:58.000000Z",  
  "created_at": "2026-01-04T04:00:58.000000Z",  
  "id": 2  
}
```

ix. Edit post (only admin can edit posts)

- a. Use Bearer Token

```
{  
  "title": "Updated Post Title by id 2",  
  "content": "Updated content of the post of id 2"  
}
```

Output –

```
{  
  "id": 2,  
  "user_id": 4,  
  "title": "Updated Post Title by id 2",  
  "content": "Updated content of the post of id 2",  
  "created_at": "2026-01-04T04:00:58.000000Z",  
  "updated_at": "2026-01-04T04:03:33.000000Z"  
}
```

x. Delete Post

- a. Use Bearer Token for auth

```
{  
  "message": "Post deleted successfully"  
}
```

xi. Edit post by user role

```
{  
  "title": "Updated Post Title by id 2",  
  "content": "Updated content of the post of id 2"  
}
```

Output –

```
{  
  "error": "Insufficient permissions",  
  "message": "You do not have the required role to access this resource."  
}
```

xii. Logout

- a. Use Bearer Token (access token)

```
{  
  "message": "Successfully logged out"  
}
```

API TESTING (WRONG DATA)

i. Register User

- a. Already register User with wrong role (“admin”, “user”)

```
{  
    "name": "admin",  
    "email": "admin@gmail.com",  
    "password": "password@123",  
    "password_confirmation": "password@123",  
    "role": "admins"  
}
```

Output –

```
{  
    "message": "The email has already been taken. (and 1 more error)",  
    "errors": {  
        "email": [  
            "The email has already been taken."  
        ],  
        "role": [  
            "The selected role is invalid."  
        ]  
    }  
}
```

ii. Login with wrong Credentials

```
{  
    "email": "admin123@gmail.com",  
    "password": "password@123",  
    "client_id": "{{client_id}}",  
    "client_secret": "{{client_secret}}"  
}
```

Output –

```
{  
    "message": "The provided credentials are incorrect.",  
    "errors": {  
        "email": [  
            "The provided credentials are incorrect."  
        ]  
    }  
}
```

iii. Login without client_id and client_secret

```
{  
  "email": "admin@gmail.com",  
  "password": "password@123"  
}
```

Output –

```
{  
  "message": "The client id field is required. (and 1 more error)",  
  "errors": {  
    "client_id": [  
      "The client id field is required."  
    ],  
    "client_secret": [  
      "The client secret field is required."  
    ]  
  }  
}
```

IMPLEMENTATION

1. Passport Installation & implementation

a. In Terminal

- composer require laravel/passport
- php artisan passport:install
- after installation use => php artisan migrate
- php artisan passport:client --personal {creates a *Personal Access Client* in Laravel Passport.}
- php artisan passport:client --password {creates a *Password Grant Client* in Laravel Passport.}
- after that it will generate client_id and client_secret

2. update AuthServiceProvider

- /app/Providers/AuthServiceProvider.php

```
class AuthServiceProvider extends ServiceProvider
{
    /**
     * The policy mappings for the application.
     *
     * @var array<class-string, class-string>
     */
    protected $policies = [
        //
    ];

    /**
     * Register any authentication / authorization services.
     */
    public function boot(): void
    {
        $this->registerPolicies();

        // Register password grant client
        Passport::enablePasswordGrant();

        // Configure token expiration times
        Passport::tokensExpireIn(now()->addDays(15));
        Passport::refreshTokensExpireIn(now()->addDays(30));
        Passport::personalAccessTokensExpireIn(now()->addMonths(6));
    }
}
```

3. CheckUserRole

- a. It Check the role for the user whether it is admin or user.

```
class CheckUserRole
{
    /**
     * Handle an incoming request.
     *
     * @param \Closure(\Illuminate\Http\Request):
     * (\Symfony\Component\HttpFoundation\Response) $next
     */
    public function handle(Request $request, Closure $next, string $role): Response
    {
        if (!$request->user() || $request->user()->role !== $role) {
            return response()->json([
                'error' => 'Insufficient permissions',
                'message' => 'You do not have the required role to access this
resource.'
            ], 403);
        }

        return $next($request);
    }
}
```

4. .env (file Configuration)

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:taYqrHfl5Kxo+g/QVuMnZjl07HzEgxxU0NQ48m5Y0UE=
APP_DEBUG=true
APP_URL=http://localhost

APP_LOCALE=en
APP_FALLBACK_LOCALE=en
APP_FAKE_LOCALE=en_US

APP_MAINTENANCE_DRIVER=file
# APP_MAINTENANCE_STORE=database

# PHP_CLI_SERVER_WORKERS=4

BCRYPT_ROUNDS=12

LOG_CHANNEL=stack
LOG_STACK=single
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=auth_test
DB_USERNAME=root
DB_PASSWORD=

SESSION_DRIVER=database
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=database

CACHE_STORE=database
# CACHE_PREFIX=

MEMCACHED_HOST=127.0.0.1

REDIS_CLIENT=phpredis
```

```
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=log
MAIL_SCHEME=null
MAIL_HOST=127.0.0.1
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false

VITE_APP_NAME="${APP_NAME}"

# Passport OAuth 2.0
PASSPORT_PASSWORD_CLIENT_ID=019b841a-ae32-71b4-9d68-bb89f466ec3f
PASSPORT_PASSWORD_CLIENT_SECRET=GN0hy4sMLbTuYVYm8mQMUqy51BYzmrC5V1SciMcj
```

Screenshots for API Testing

POST regi ○ New E My C POST Log ○ [CONFLICT ○ POST log ○ POST log ○ f + New Environment

POST register

POST login

POST Login oauth

GET user

POST logout

POST logout all

POST refresh

GET posts

GET posts id

POST add post

PUT edit post

Variables in request

base_url http://localhost:8000

client_id 019b841a-ae32-71b4-9...

client_secret GNOhy4sMLbtUVYyM8...

My Collection

No variables defined in this collection. Add

G Globals

No global variables in this workspace. Add

Local Vault

Store your API secrets locally in vault. Set up vault

Figure 1 – Register API Testing

POST regi ○ New E My C POST Log ○ [CONFLICT ○ POST log ○ POST log ○ f + New Environment

POST register

POST login

POST Login oauth

GET user

POST logout

POST logout all

POST refresh

GET posts

GET posts id

POST add post

PUT edit post

Variables in request

base_url http://localhost:8000

client_id 019b841a-ae32-71b4-9...

client_secret GNOhy4sMLbtUVYyM8...

All variables

New Environment

base_url http://localhost:8000

client_id 019b841a-ae32-71b4-9...

client_secret GNOhy4sMLbtUVYyM8...

Enter value

My Collection

No variables defined in this collection. Add

G Globals

No global variables in this workspace. Add

Local Vault

Store your API secrets locally in vault. Set up vault

Figure 2 – Login API Testing

Figure 3 – View user (authentication Required)

Figure 4 – Refresh Token

The screenshot shows the Postman interface with a collection named "My Collection". A GET request is made to `http://localhost:8000/api/posts`. The response status is 200 OK, and the response body is a JSON array of two post objects. The first post has id 1, user_id 1, title "Updated Post Title", content "Updated content of the post", created_at "2026-01-03T13:51:12.000000Z", updated_at "2026-01-03T13:53:25.000000Z", and user object. The second post has id 2, user_id 4, title "Updated Post Title by id 2", content "Updated content of the post of id 2", created_at "2026-01-04T04:03:58.000000Z", updated_at "2026-01-04T04:03:33.000000Z", and user object.

```

[{"id": 1, "user_id": 1, "title": "Updated Post Title", "content": "Updated content of the post", "created_at": "2026-01-03T13:51:12.000000Z", "updated_at": "2026-01-03T13:53:25.000000Z", "user": {"id": 1, "name": "Sizan", "email": "sizan@gmail.com", "email_verified_at": null, "created_at": "2026-01-03T13:45:17.000000Z", "updated_at": "2026-01-03T13:45:17.000000Z", "role": "admin"}], [{"id": 2, "user_id": 4, "title": "Updated Post Title by id 2", "content": "Updated content of the post of id 2", "created_at": "2026-01-04T04:03:58.000000Z", "updated_at": "2026-01-04T04:03:33.000000Z", "user": {"id": 4, "name": "admin", "email": "admin@gmail.com", "email_verified_at": null}}]

```

Figure 5 – View Posts {ALL}

The screenshot shows the Postman interface with a collection named "My Collection". A GET request is made to `http://localhost:8000/api/posts/1`. The response status is 200 OK, and the response body is a single post object with id 1, user_id 1, title "Updated Post Title", content "Updated content of the post", created_at "2026-01-03T13:51:12.000000Z", updated_at "2026-01-03T13:53:25.000000Z", and user object.

```

{"id": 1, "user_id": 1, "title": "Updated Post Title", "content": "Updated content of the post", "created_at": "2026-01-03T13:51:12.000000Z", "updated_at": "2026-01-03T13:53:25.000000Z", "user": {"id": 1, "name": "Sizan", "email": "sizan@gmail.com", "email_verified_at": null, "created_at": "2026-01-03T13:45:17.000000Z", "updated_at": "2026-01-03T13:45:17.000000Z", "role": "admin"}}

```

Figure 6 – View Post by {id}

The screenshot shows the Postman interface with the following details:

- Request:** POST to `{{base_url}}/api/posts`
- Authorization:** Bearer Token
- Response Status:** 201 Created
- Response Body (JSON):**

```

1
2   "user_id": 5,
3   "title": "My New Post",
4   "content": "This is the content of my post",
5   "updated_at": "2026-01-04T08:36:32.000000Z",
6   "created_at": "2026-01-04T08:36:32.000000Z",
7   "id": 3
8
  
```

Figure 7 – Add Post

The screenshot shows the Postman interface with the following details:

- Request:** PUT to `{{base_url}}/api/posts/2`
- Authorization:** Bearer Token
- Response Status:** 200 OK
- Response Body (JSON):**

```

1
2   "id": 2,
3   "user_id": 4,
4   "title": "Updated Post Title by id 2",
5   "content": "Updated content of the post of id 2",
6   "updated_at": "2026-01-04T08:58:00.000000Z",
7   "created_at": "2026-01-04T08:03:33.000000Z"
8
  
```

Figure 8 – Edit post (admin) only