

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

**Aim** : An online learning platform named offers a variety of courses to students. To protect students' personal information and course materials, requires users to log in before accessing their courses. The platform uses IBM Cloud App ID for managing user authentication and authorization.

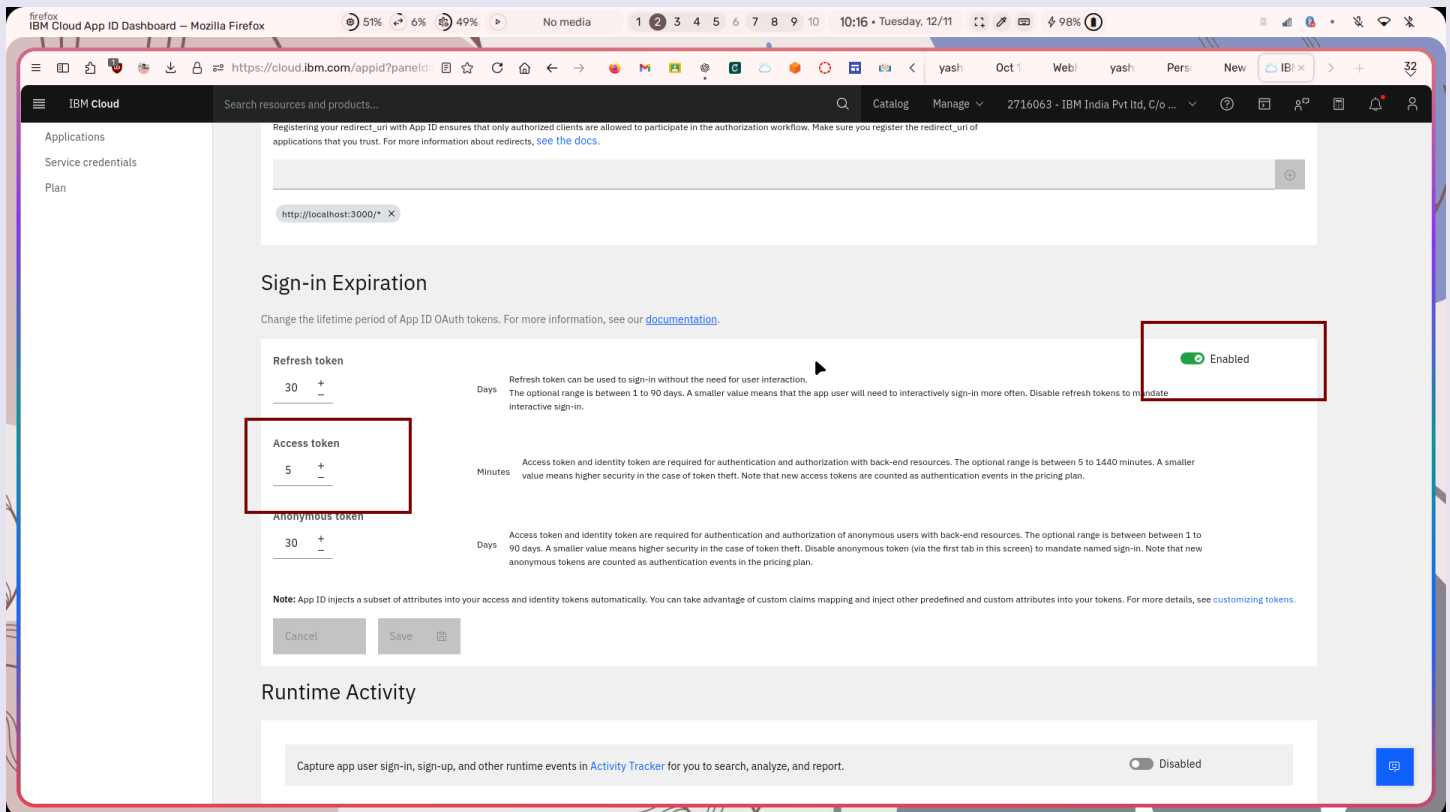
**13.1:** To ensure a smooth user experience, wants users to stay logged in while studying without frequent re-authentication prompts. Since access tokens have a short expiration time (e.g., 15 minutes), the platform uses refresh tokens to generate new access tokens when the original token expires, allowing users to stay logged in and access resources seamlessly. Demonstrate how the refresh token can be used to regenerate the access token without any need to login again.

**13.2:** As part of enhancing authentication for your application, set up an on IBM Cloud to connect external user repositories to your IBM Cloud account. This will allow users to log in with their organization's credentials, enabling seamless access without needing individual account invites. Create a new and configure it and finally test the configuration by logging in using a user from the external directory to ensure that authentication works as expected.

**Steps and Screenshots :**

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 71  
CS Practical 13

## 1. Turn on Refresh Token settings for AppID instance and set minimum duration of Access Token



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

2. Use previous code of practical 11 to access the node app after logging into AppID, which return access, identity, refresh tokens in console after login, copy them

Code :

```
const express = require('express');
const session = require('express-session');
const passport = require('passport');
const WebAppStrategy = require('ibmcloud-appid').WebAppStrategy;
const app = express();
var port = 3000;
app.use(session({
  secret: '123456',
  resave: true,
  saveUninitialized: true
}));
app.use(passport.initialize());
app.use(passport.session());

passport.serializeUser((user, cb) => cb(null, user));
passport.deserializeUser((user, cb) => cb(null, user));

passport.use(new WebAppStrategy({
  tenantId: "",
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

```
    clientId: "",
    secret: "",
    oauthServerUrl:
"https://au-syd.appid.cloud.ibm.com/oauth/v4/",
    redirectUri: "http://localhost:3000/appid/callback"
  }));

app.get('/appid/login',
passport.authenticate(WebAppStrategy.STRATEGY_NAME, {
  successRedirect: '/',
  forceLogin: true
}));

//
// Handle callback
app.get('/appid/callback',
passport.authenticate(WebAppStrategy.STRATEGY_NAME, {
  keepSessionInfo: true }));
//app.use(passport.authenticate(WebAppStrategy.STRATEGY_NAME));
app.use('/api', (req, res, next) => {
  if (req.user) {
    next();
  } else {
```

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

```
        res.status(401).send("Unauthorized");
    }
});

app.get('/api/user', (req, res) => {
    //console.log(res);
    console.log(req.session[WebAppStrategy.AUTH_CONTEXT]);
    res.json({
        user: {
            name: req.user.name
        }
    });
});

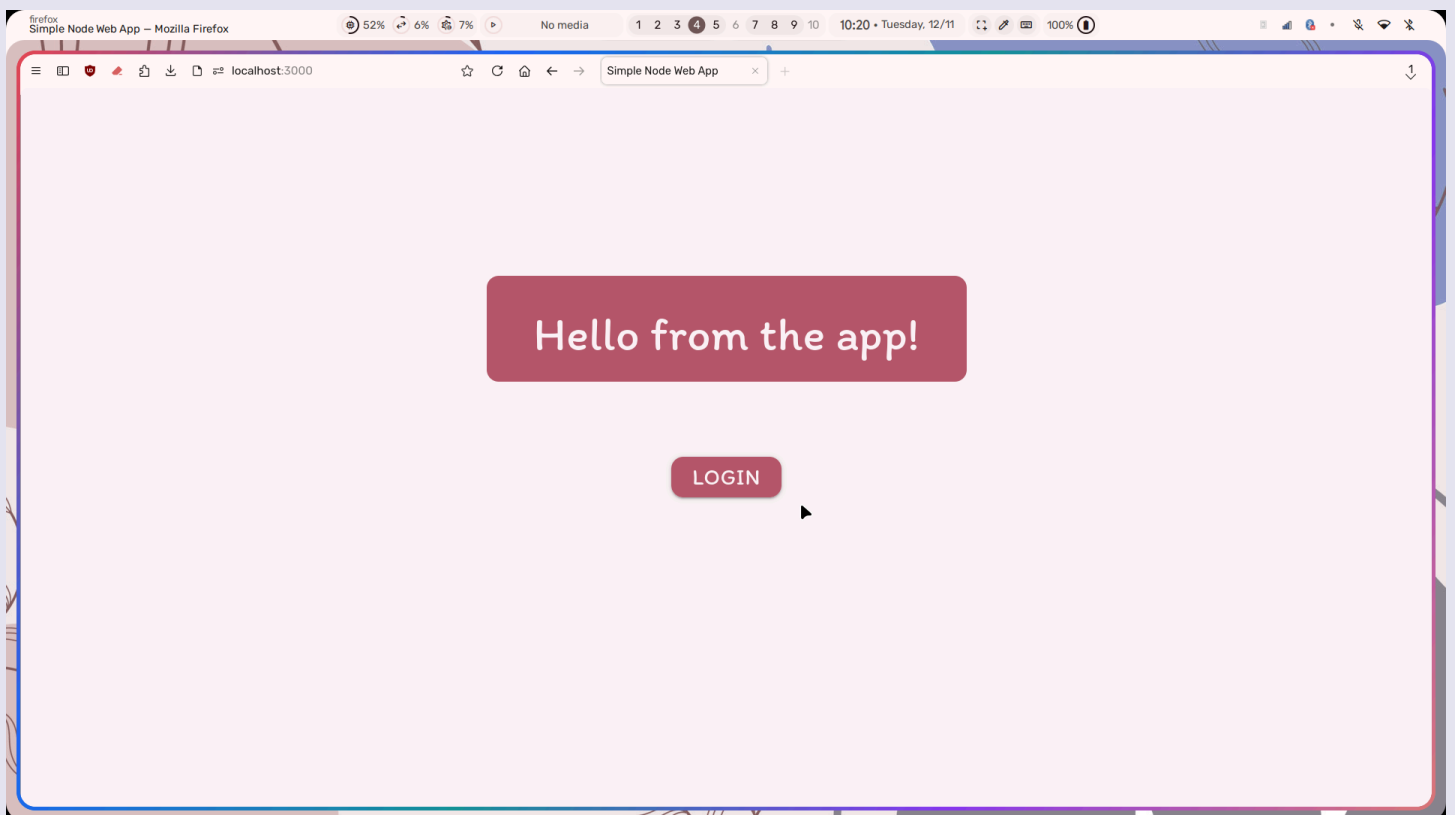
});

app.get("/appid/logout", function (req, res) {
    req.logout(function (err) {
        if (err) { console.log(err); }
        res.redirect("/");
    });
});
});
```

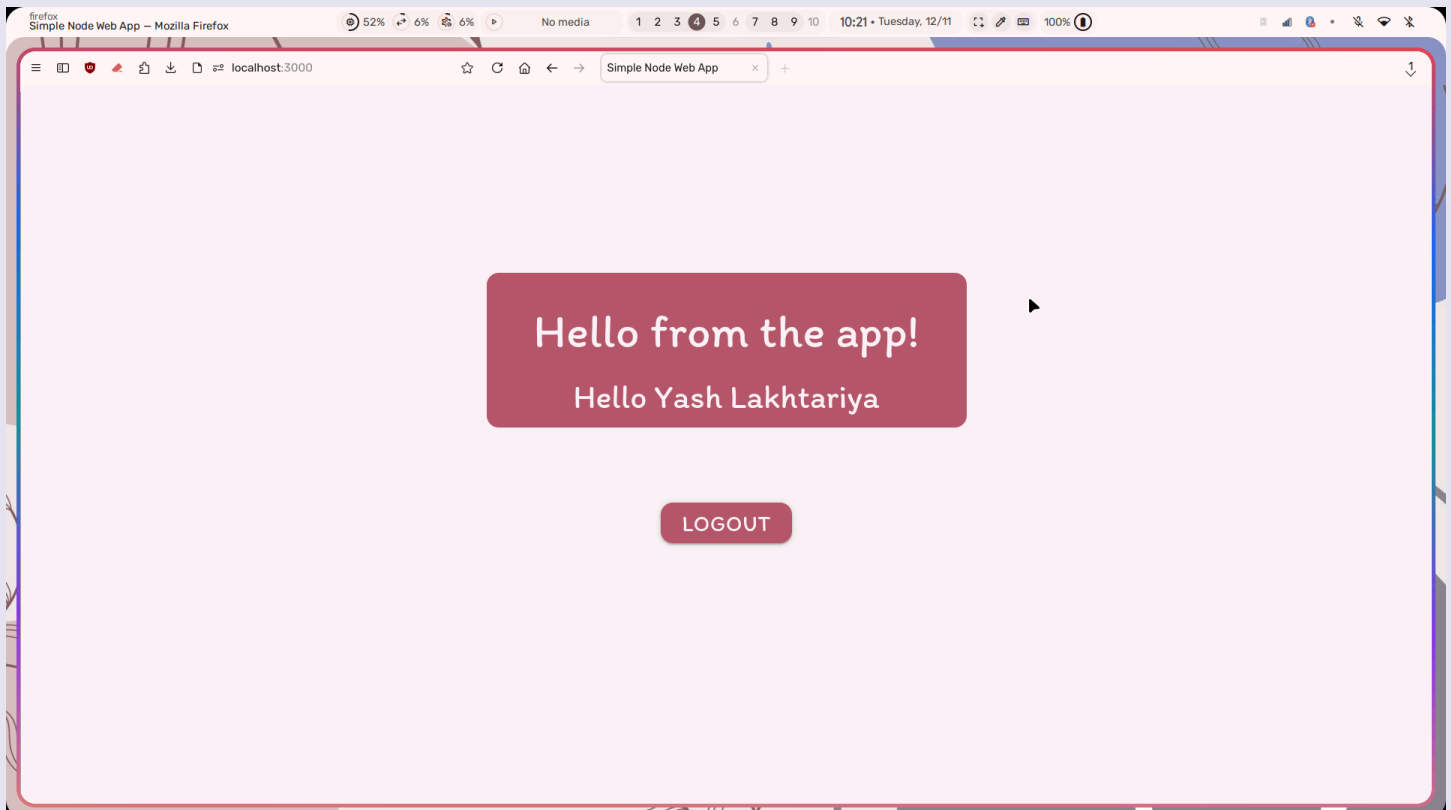
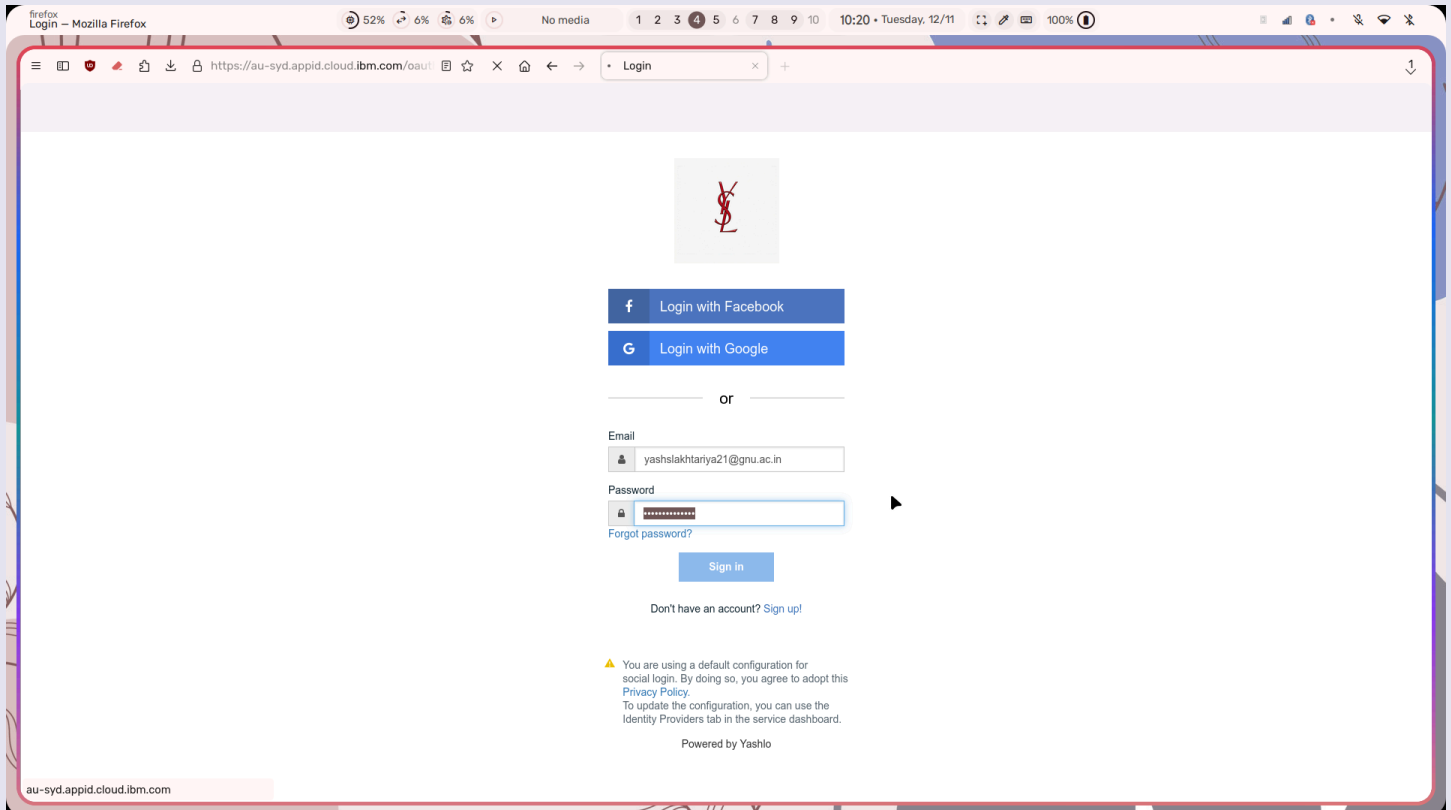
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

```
// Serve static resources
```

```
app.use(express.static('./public'));  
app.listen(port);
```



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 71  
CS Practical 13

```
code-unt-handler
app.js - Practical_13 - Visual Studio Code
52% 6% 4% No media 1 2 3 4 5 6 7 8 9 10 10:23 • Tuesday, 12/11 100%

File Edit Selection View Go Run Terminal Help
Practical_13

EXPLORER
Practical_13
  node_modules
  public
    index.html
  app.js
  app2.js
  package-lock.json
  package.json

OUTLINE
PORTS SQL CONSOLE GIT LENS TERMINAL COMMENTS

[2024-11-12T10:21:06.560] [INFO] appid-token-util - token version is valid
[2024-11-12T10:21:06.560] [INFO] appid-token-util - token aud: 08a70b32-5b28-4162-b9da-66a6d35e9cec
[2024-11-12T10:21:06.560] [INFO] appid-token-util - token iss: https://au-syd.appid.cloud.ibm.com/oauth/v4

{
  refreshToken: '...',
  accessToken: '...'
}

accessTokenPayload: {
  iss: 'https://au-syd.appid.cloud.ibm.com/oauth/v4',
  exp: 1731307207,
  aud: [ ... ]
}
```



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

3. Run another code to use WebApp Strategy to access data using access token copied above

Code :

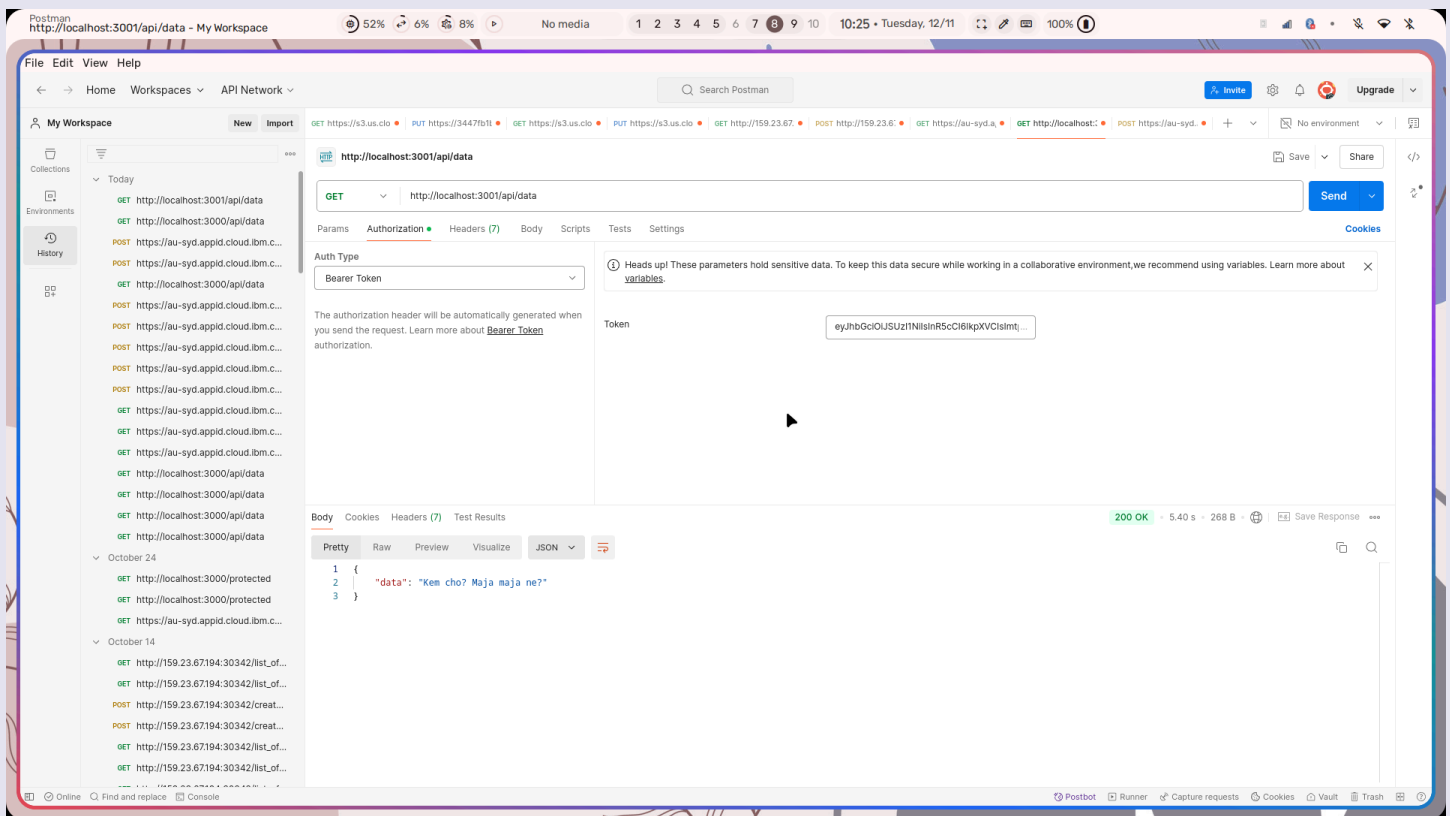
```
const express = require('express');
const passport = require('passport');
const APIStrategy = require('ibmcloud-appid').APIStrategy;
const app = express();
app.use(passport.initialize());
passport.use(new APIStrategy({
  oauthServerUrl:
    "https://au-syd.appid.cloud.ibm.com/oauth/v4/",
}));

// Protect the whole app
app.use(passport.authenticate(APIStrategy.STRATEGY_NAME, {
  session: false
}));

// The /api/data API used to retrieve protected data
app.get('/api/data', (req, res) => {
  res.json({
    data: 'Kem cho? Maja maja ne?'
  });
});
```

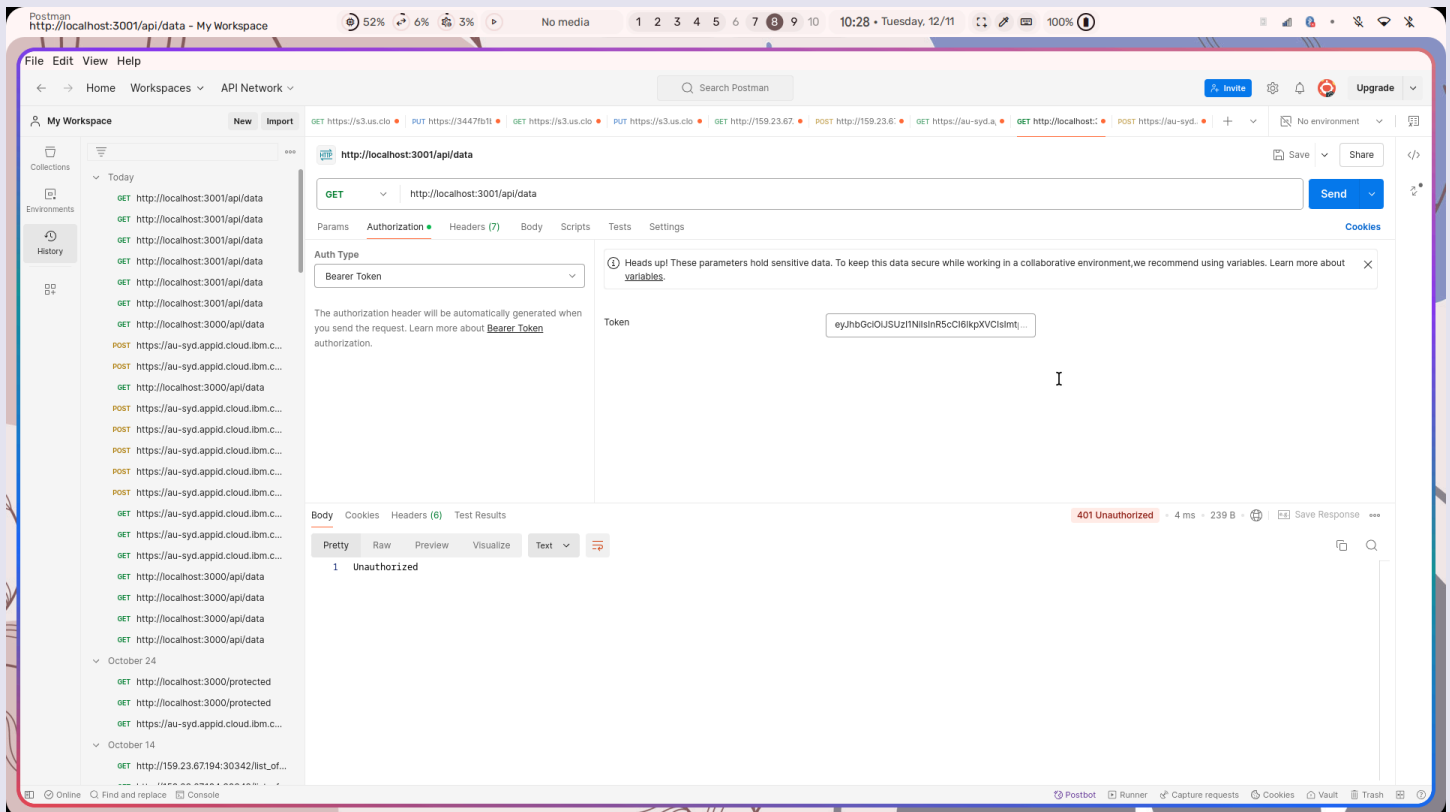
Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

```
app.listen(3001, () => {  
  console.log('Listening on http://localhost:3001');  
});
```



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 71**  
**CS Practical 13**

4. Now, after specified duration, the request will fail due to access token being expired



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 71  
CS Practical 13

5. Now, to get access token again without logging in again, the refresh token can be used to regenerate it.

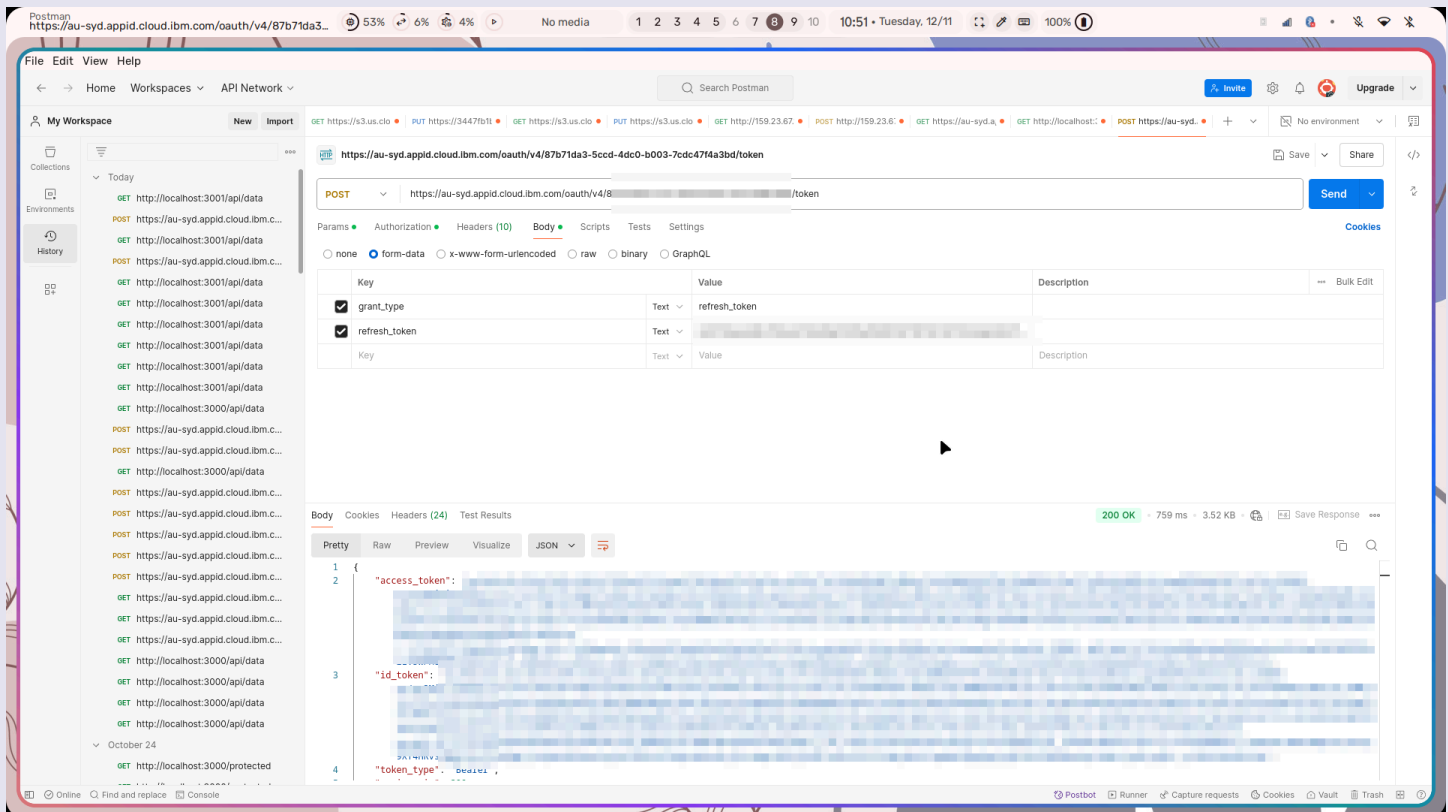
Request : POST

API endpoint : <oAuthURL>/token

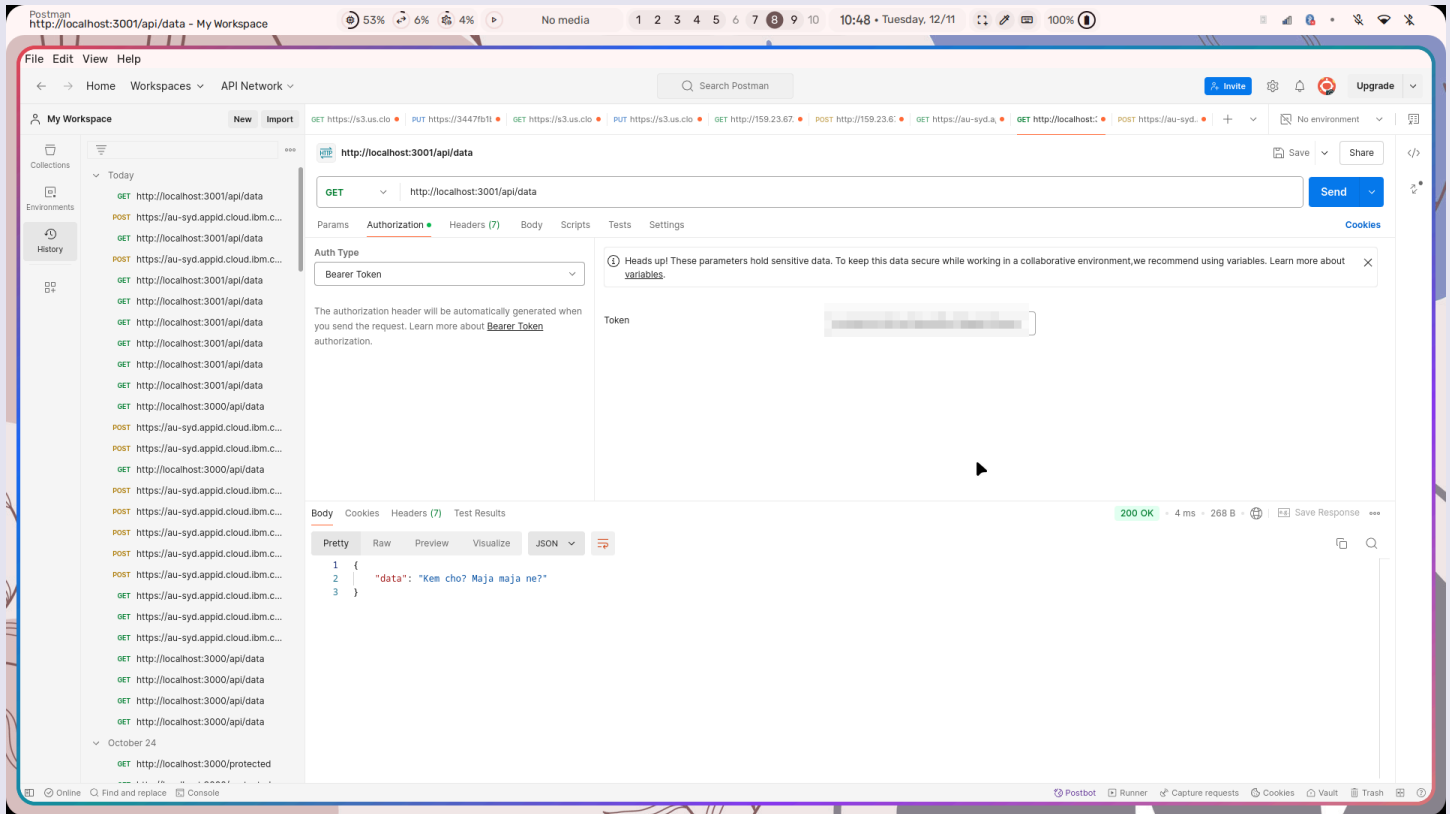
Body : (type form data)

grant\_type : refresh\_token

refresh\_token : <refresh\_token>

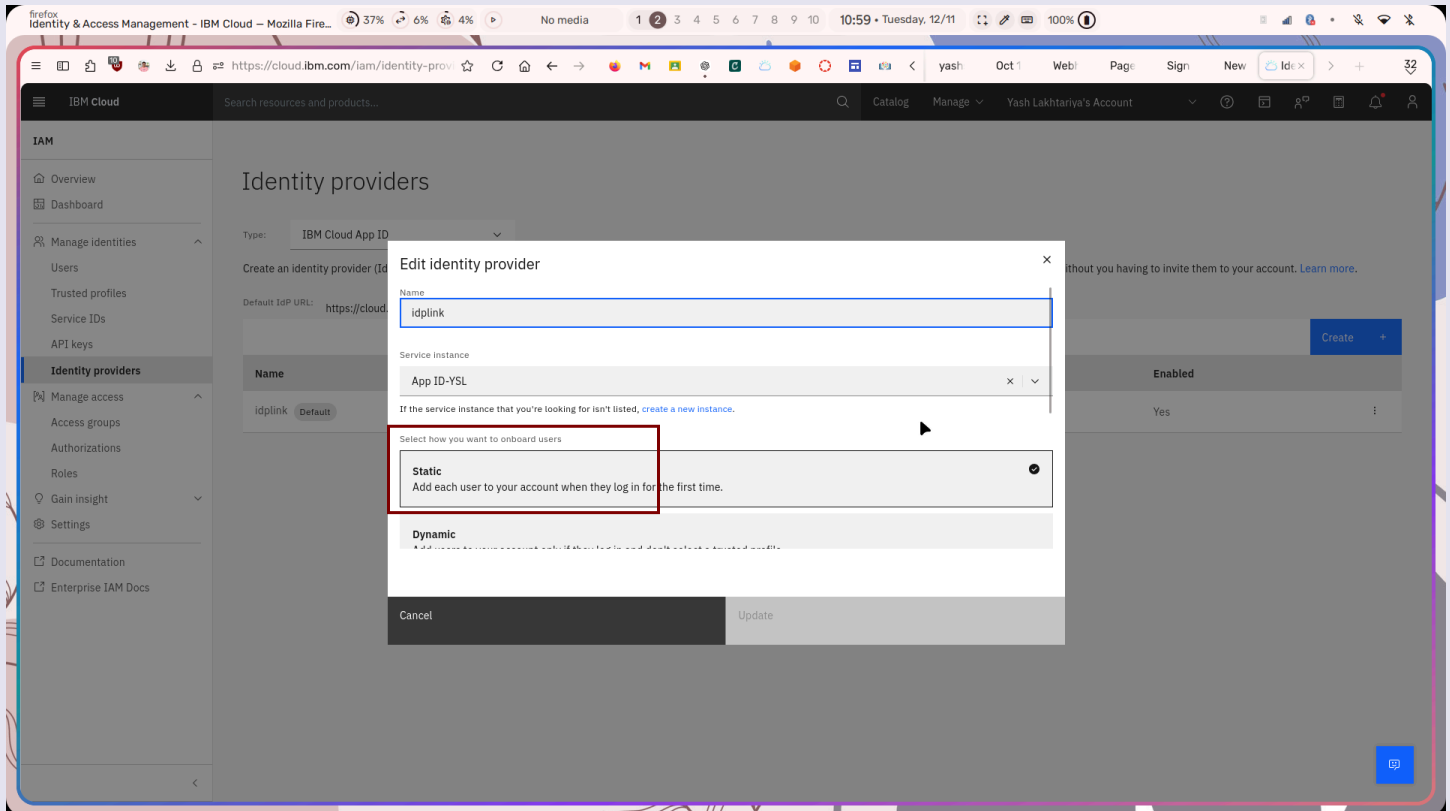


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 71  
CS Practical 13

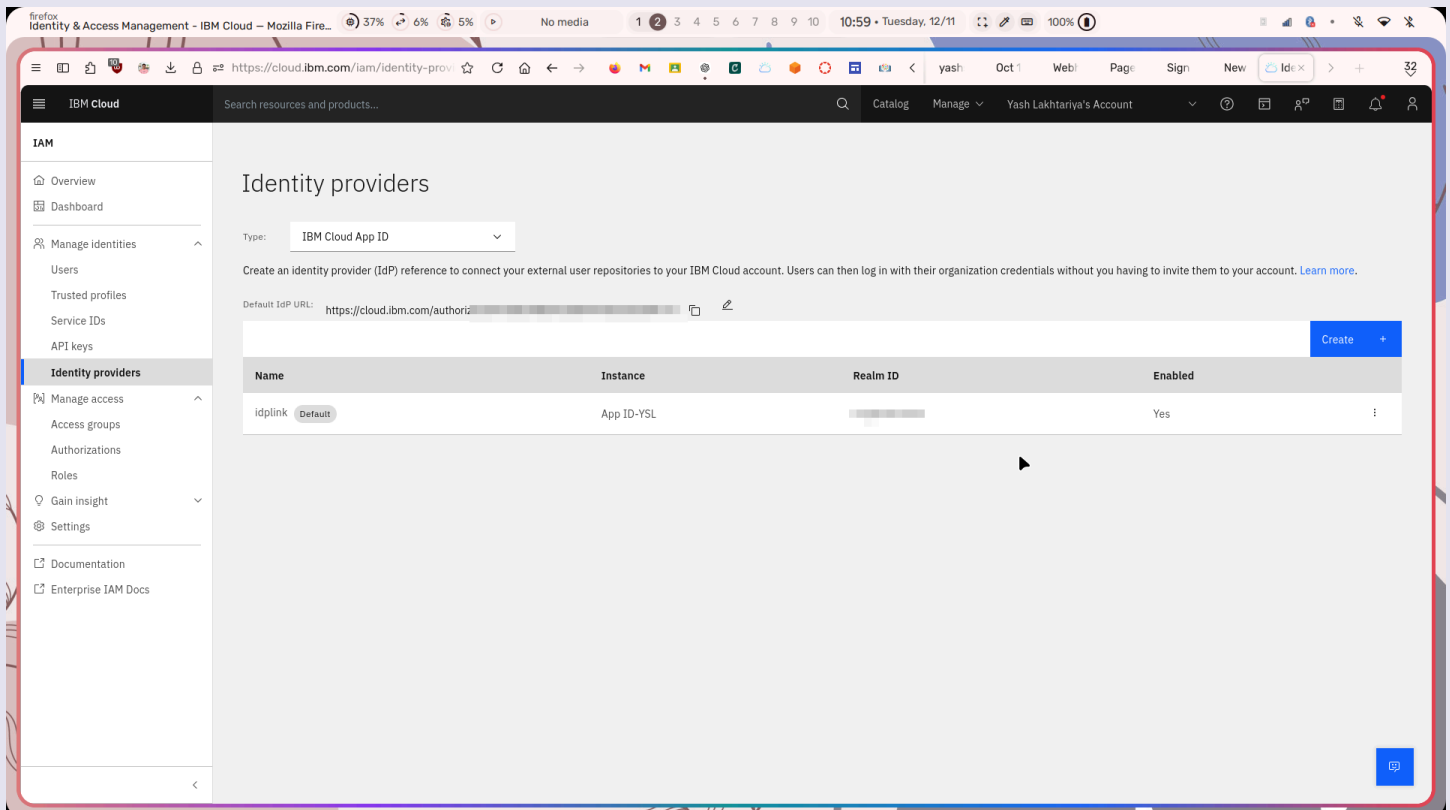
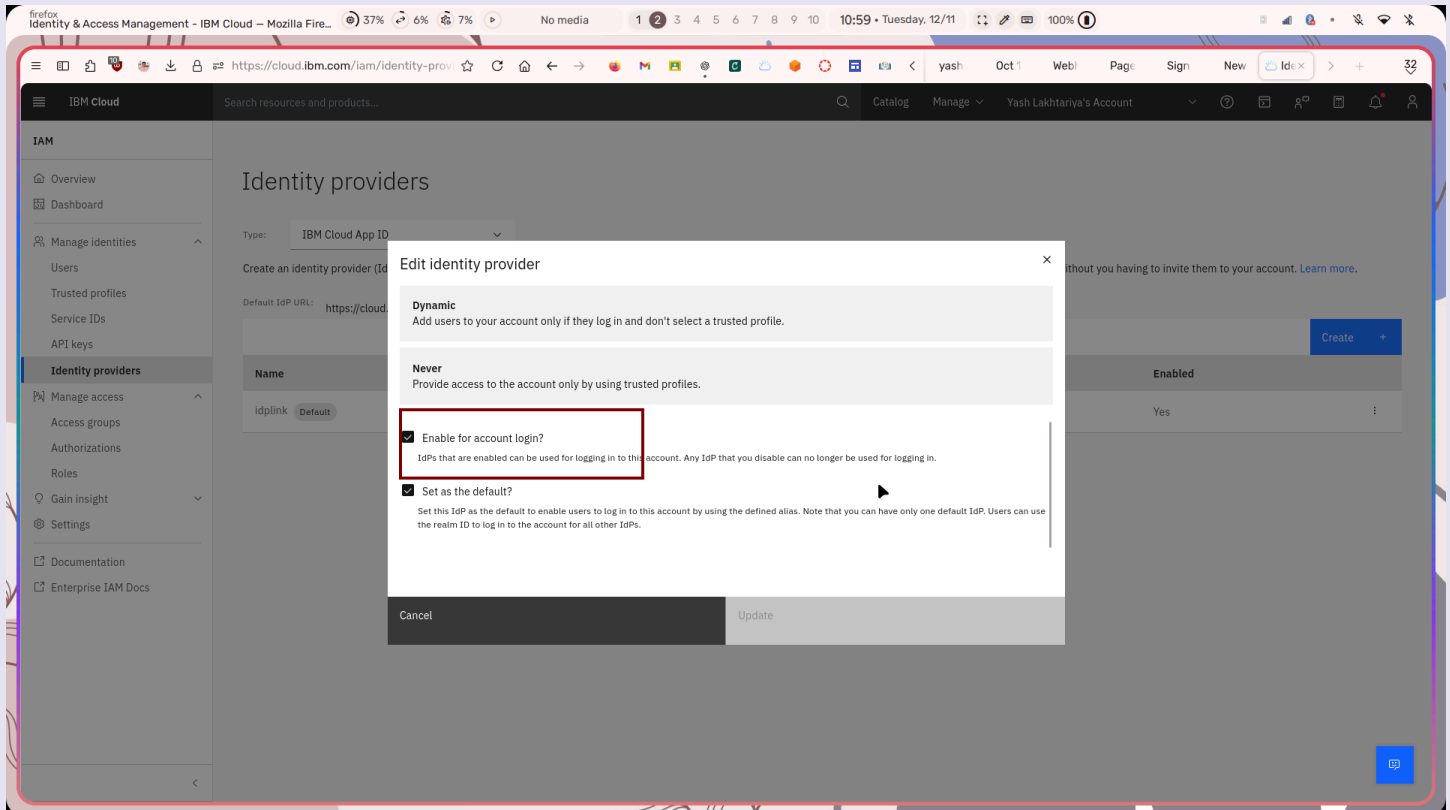


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

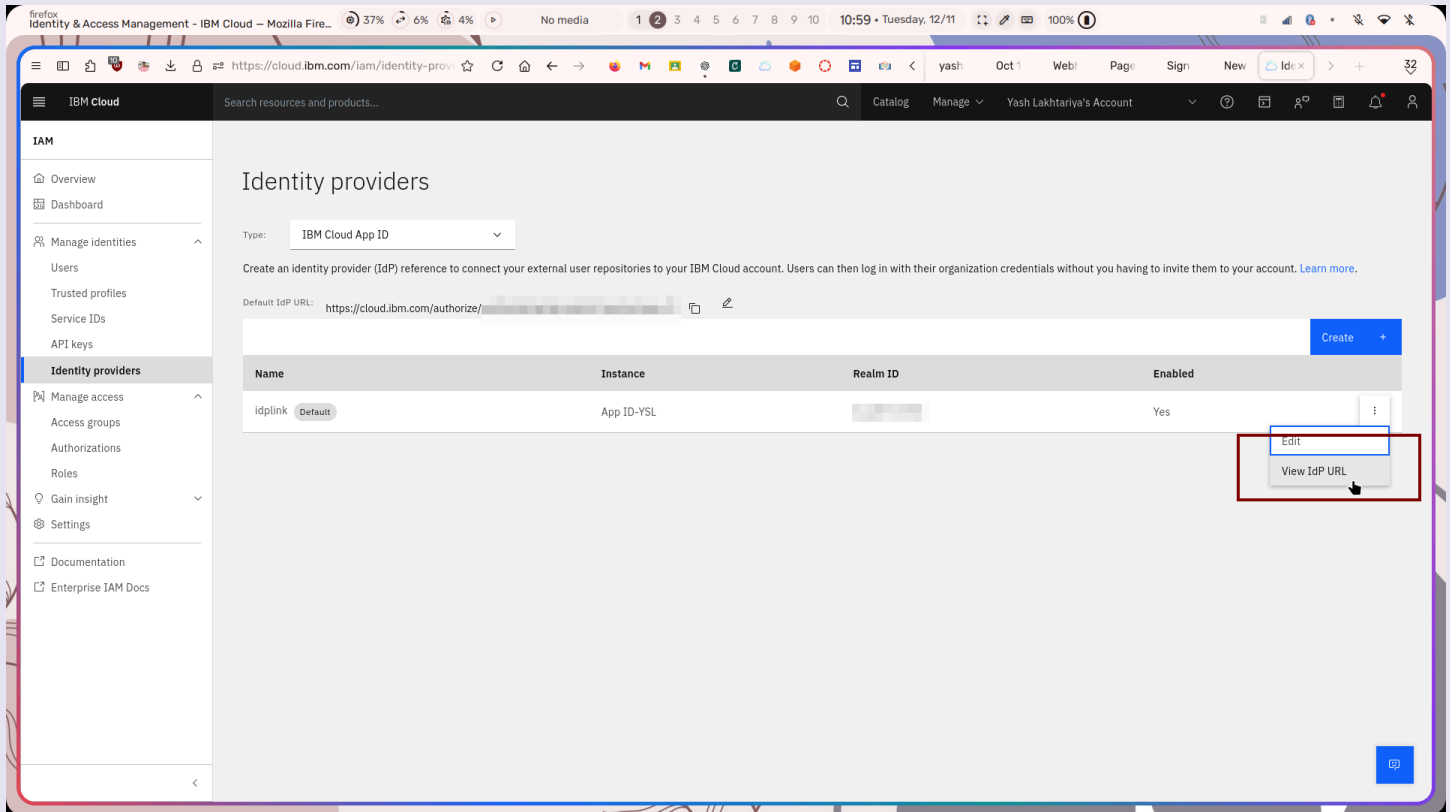
6. For reliable access using IAM, we can use identity provider link to automate adding IAM users on their signup and first login (access to resources must be specified explicitly for that user, it just allows to login)



Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 71  
CS Practical 13

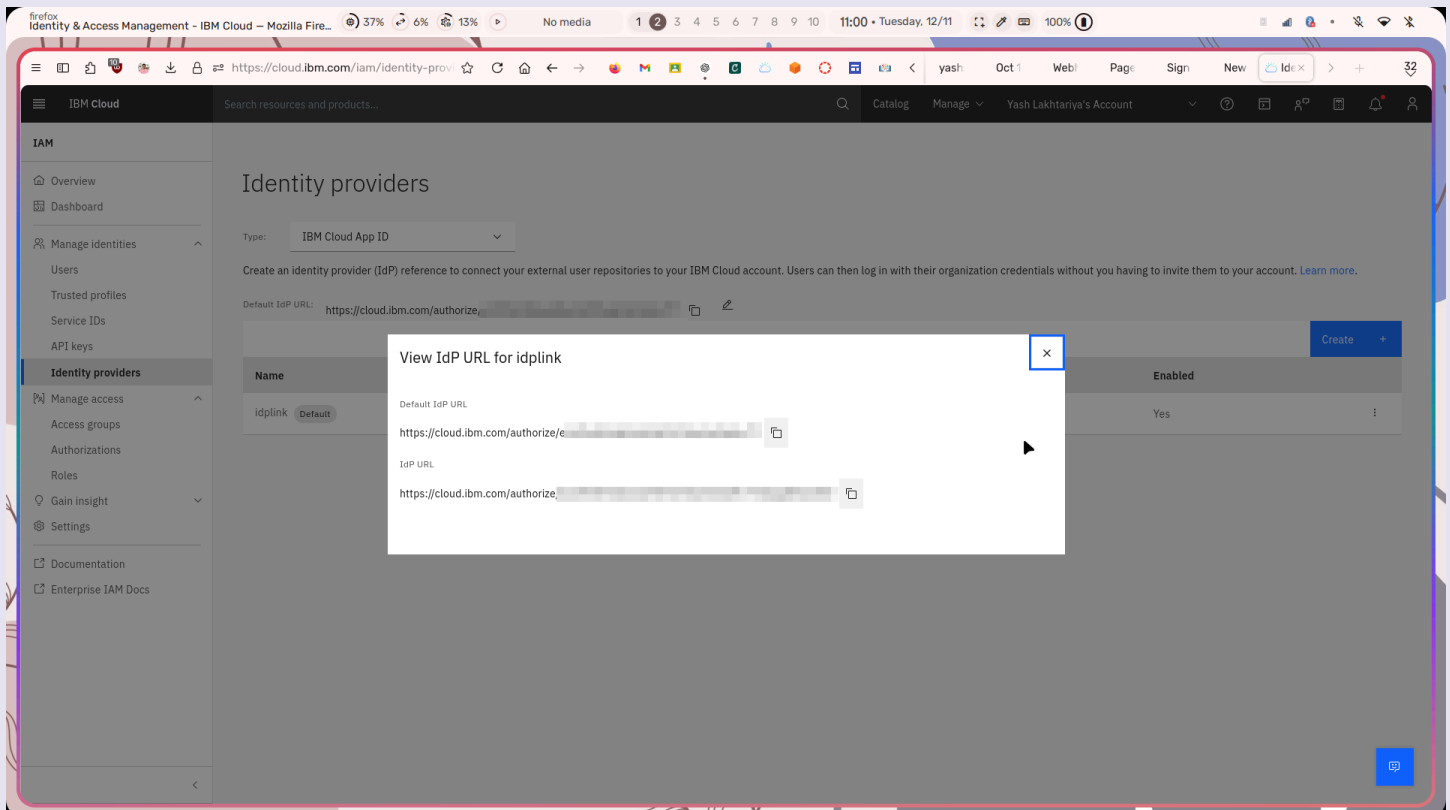


Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA Batch - 71  
CS Practical 13



This screenshot shows the 'Identity providers' page in the IBM Cloud IAM console. The left sidebar contains navigation links for Overview, Dashboard, Manage identities, Users, Trusted profiles, Service IDs, API keys, Identity providers, Manage access, Access groups, Authorizations, Roles, Gain insight, Settings, Documentation, and Enterprise IAM Docs. The main content area is titled 'Identity providers' and includes a 'Type' dropdown set to 'IBM Cloud App ID'. Below this is a description and a 'Default IdP URL' field. A table lists the existing identity providers, with one entry 'idplink' (Default) having an 'Instance' of 'App ID-YSL' and an 'Enabled' status of 'Yes'. A red box highlights the 'Edit' and 'View IdP URL' buttons for the 'idplink' provider.

Name	Instance	Realm ID	Enabled
idplink (Default)	App ID-YSL		Yes



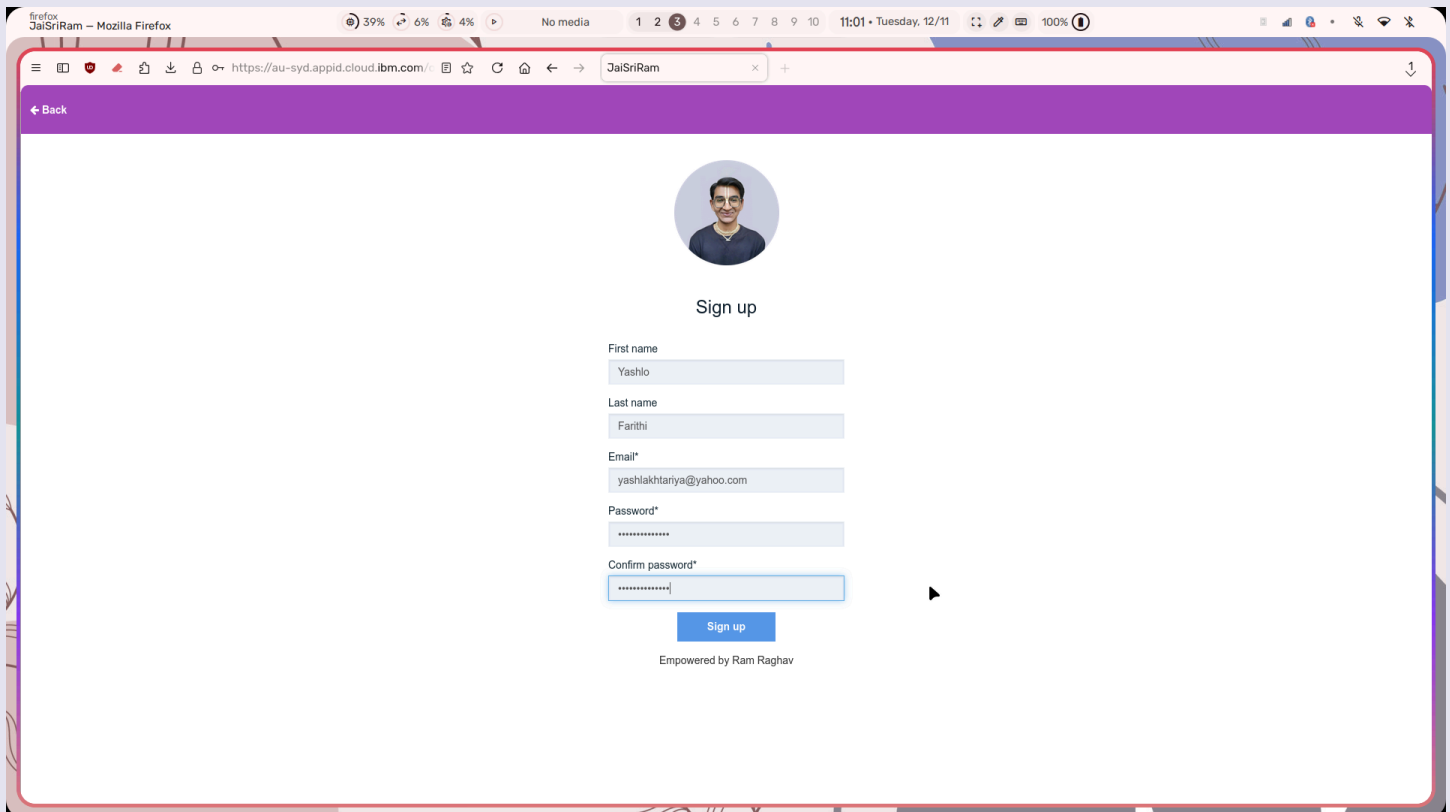
This screenshot shows the same 'Identity providers' page, but with a modal window open titled 'View IdP URL for idplink'. The modal displays the 'Default IdP URL' and the 'IdP URL' for the 'idplink' provider. The background page is dimmed, and the 'Edit' button from the previous screenshot is now visible in the top right corner of the modal.

Name	Instance	Realm ID	Enabled
idplink (Default)	App ID-YSL		Yes



**Name - Yash Lakhtariya**  
**Enrollment number - 21162101012**  
**Branch - CBA      Batch - 71**  
**CS Practical 13**

7. Open this link or share with anyone to add user on signup and first login using AppID provider



The screenshot shows a web browser window with the address bar displaying `https://au-syd.appid.cloud.ibm.com/`. The page has a purple header with a 'Back' button. Below the header, there is a circular profile picture of a man with glasses. Underneath the picture, the text 'Sign up' is displayed. The form contains the following fields:

- First name:
- Last name:
- Email\*:
- Password\*:
- Confirm password\*:

Below the form is a blue 'Sign up' button. At the bottom of the page, it says 'Empowered by Ram Raghav'.

Name - Yash Lakhtariya  
Enrollment number - 21162101012  
Branch - CBA      Batch - 71  
CS Practical 13

