

Case study Management

Use AWS and develop a Simple Server less application to Upload Files to S3

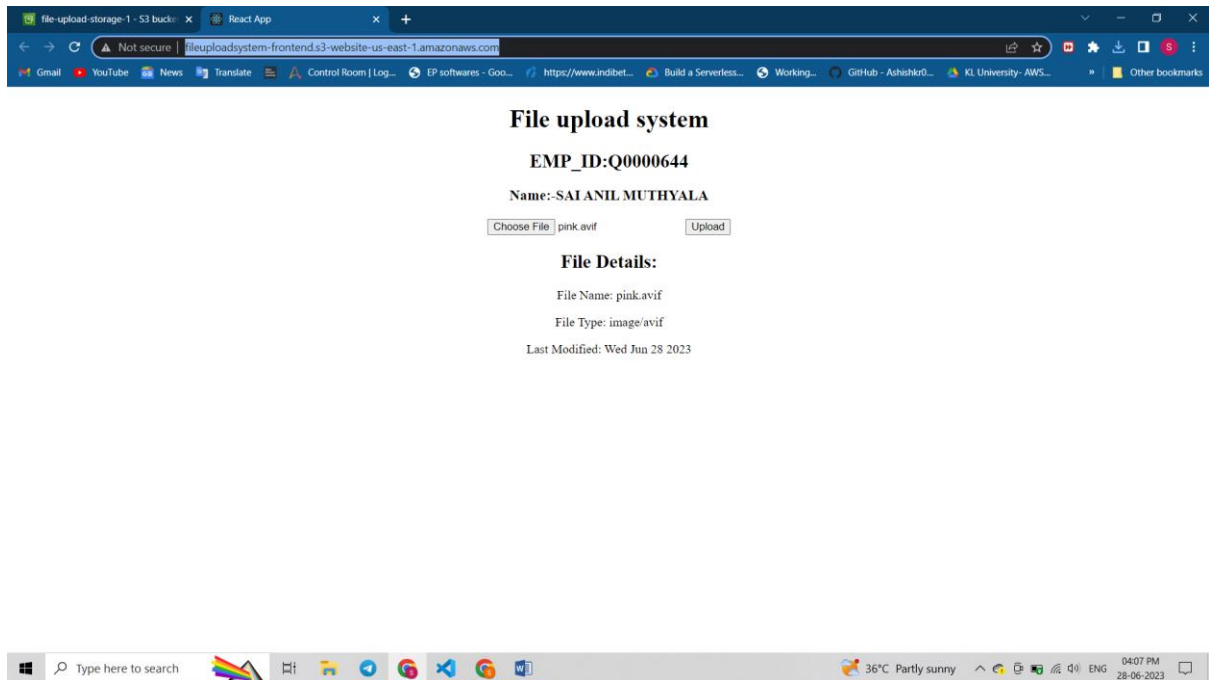
Services used :- S3, AWS Lambda, API Gateway, AWS IAM

AWS S3:-

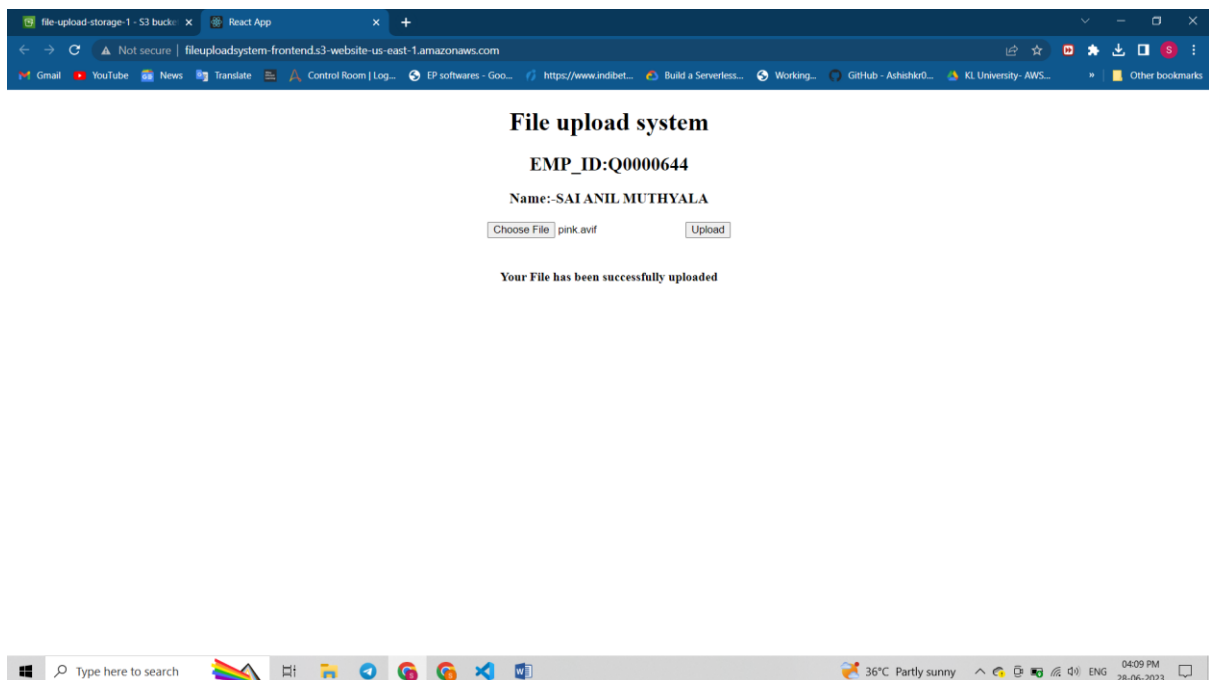
1. **Two S3 buckets are created.**
2. **One bucket is used for hosting the frontend application created by using React.Js**
3. **Another bucket is used for storing files uploaded by user.**

Front end:

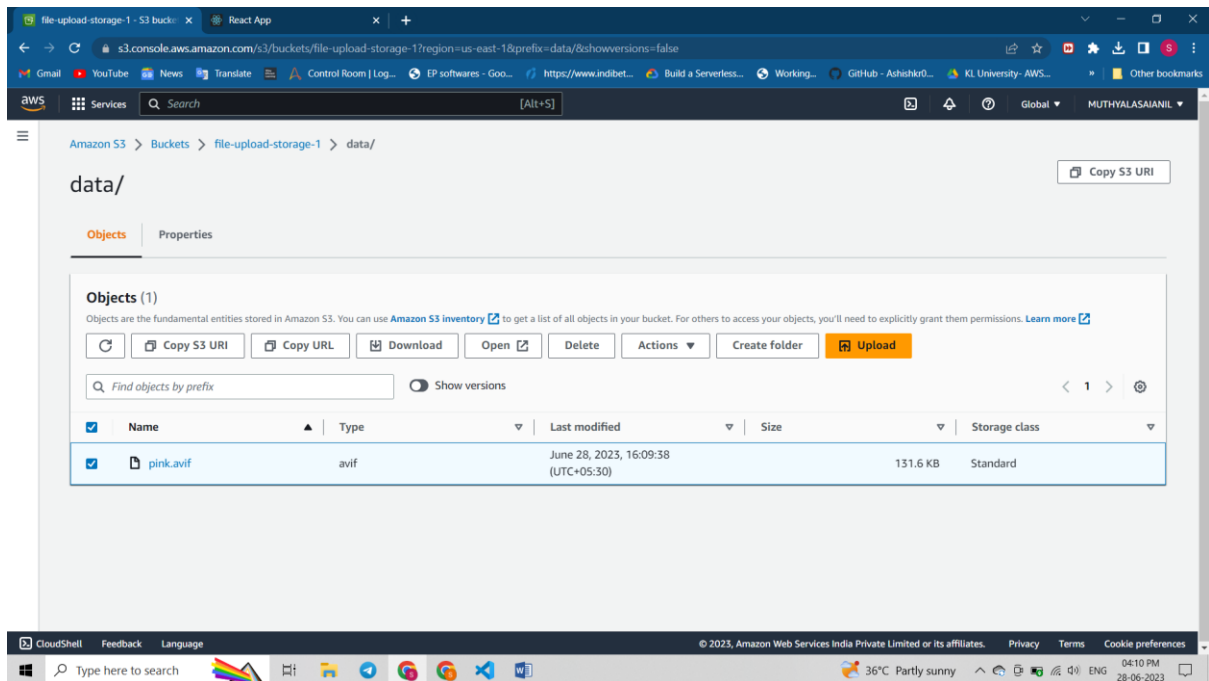
After hosting the website on AWS by using static website hosting. It looks like this:



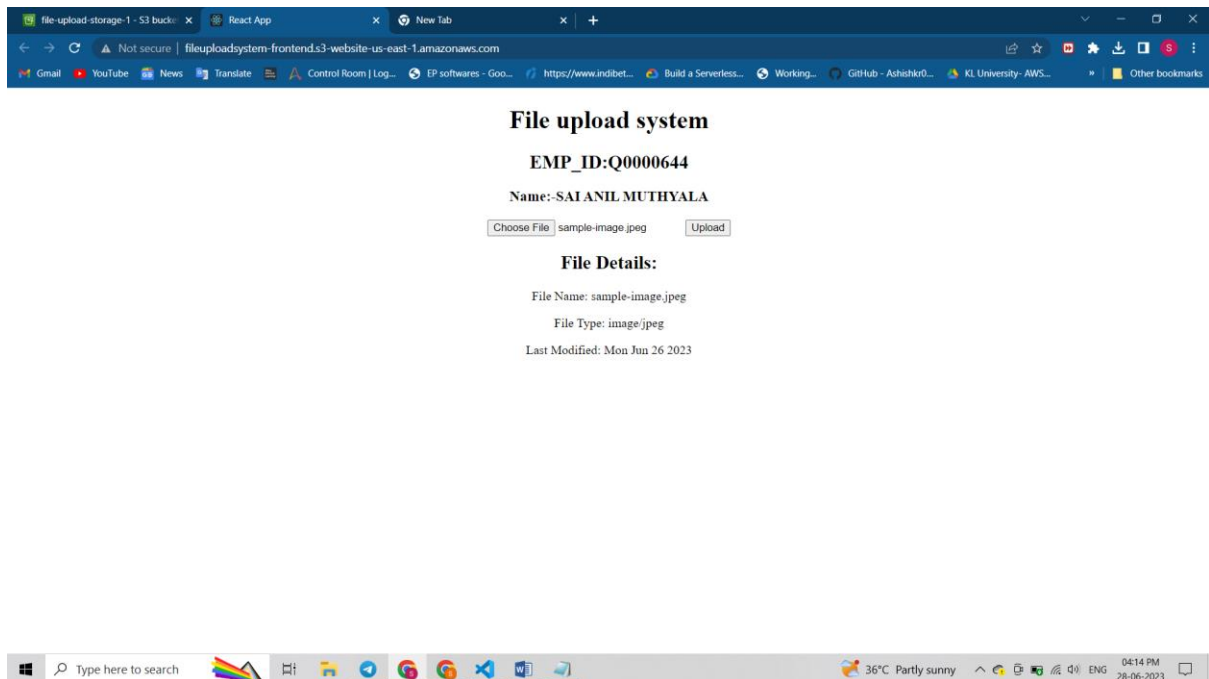
Here, we can see that the status as “the user uploaded file has been successfully upload into S3 bucket”



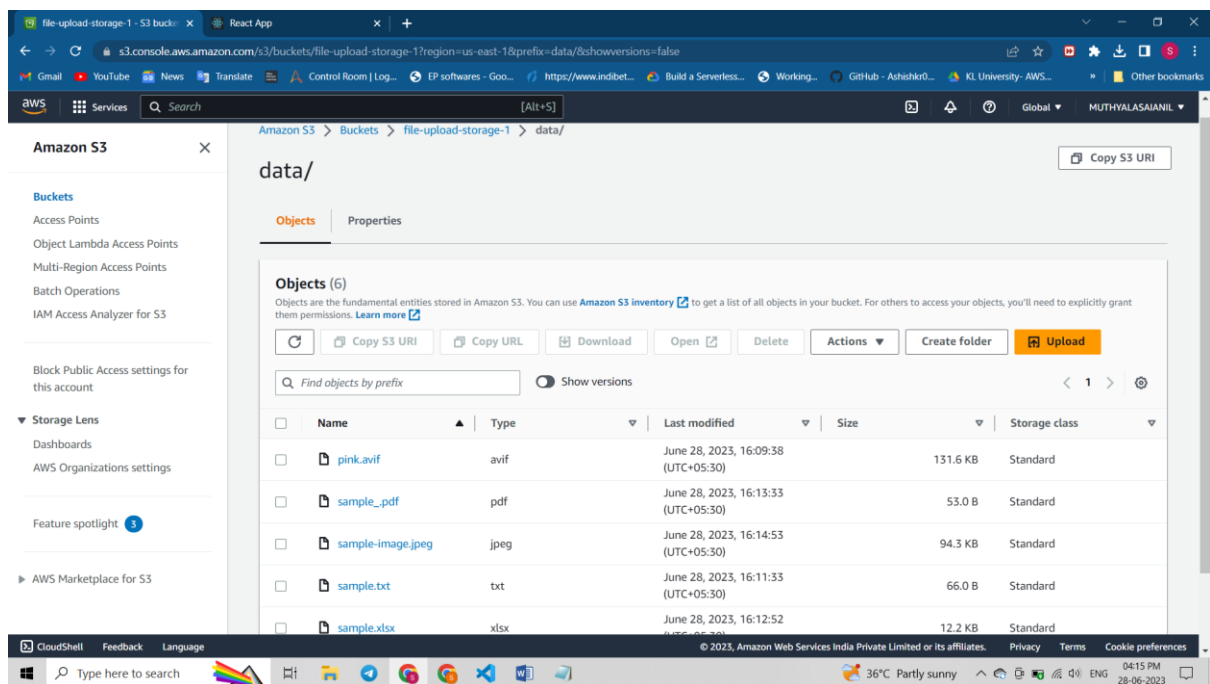
Here, we can see uploaded file from frontend has been successfully stored into backend S3 bucket.



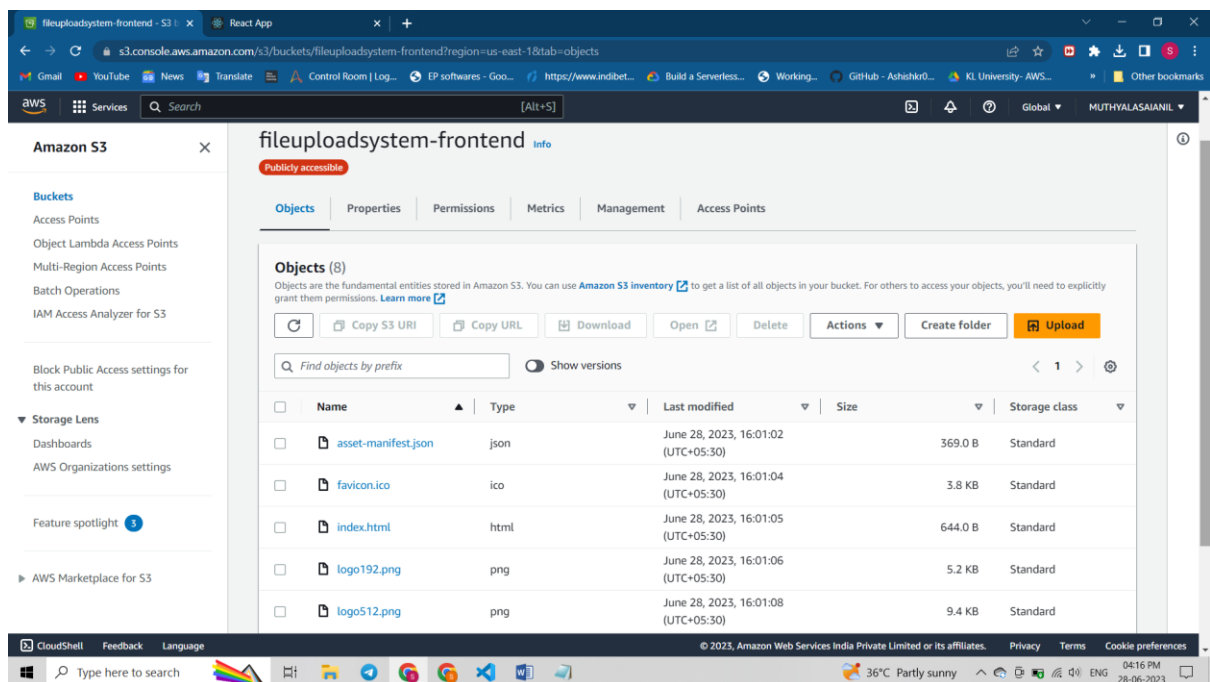
Here, user is uploading .jpeg file into S3 bucket

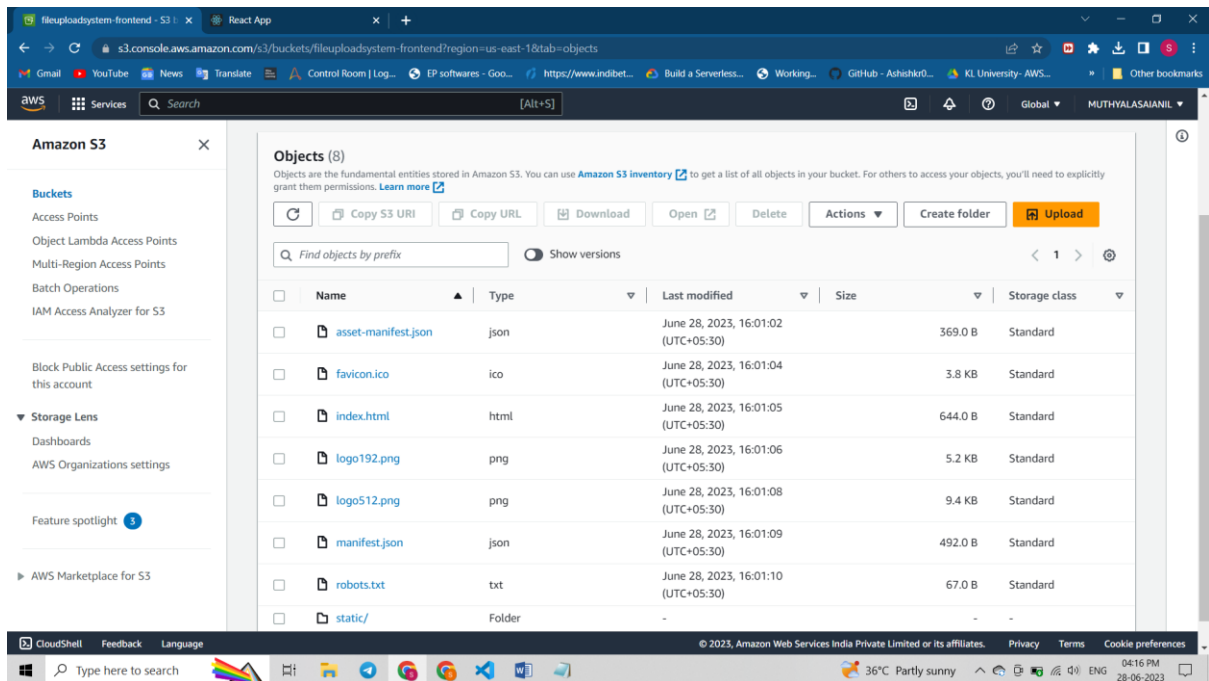


We can store any file type like (.jpeg,.txt,.xlsx,.pdf,.avif,...) into backend S3 bucket.

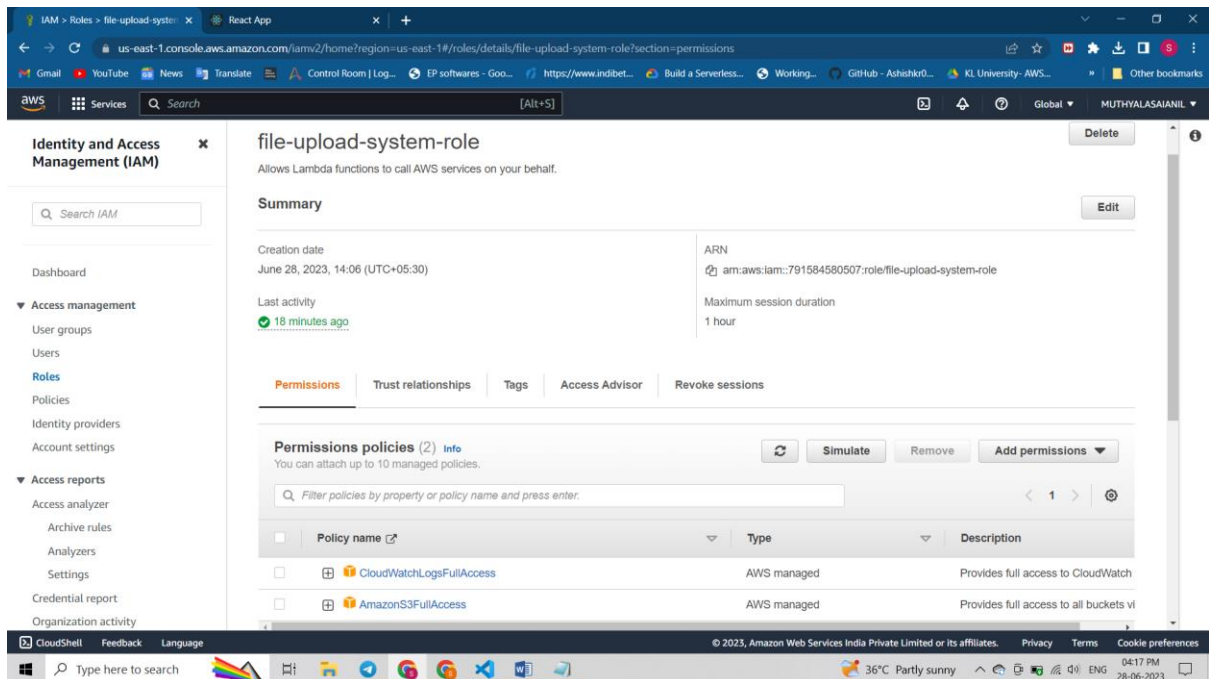


We created one bucket for front end for storing files like (.html,.css,.js, /static folder) for hosting the frontend application.



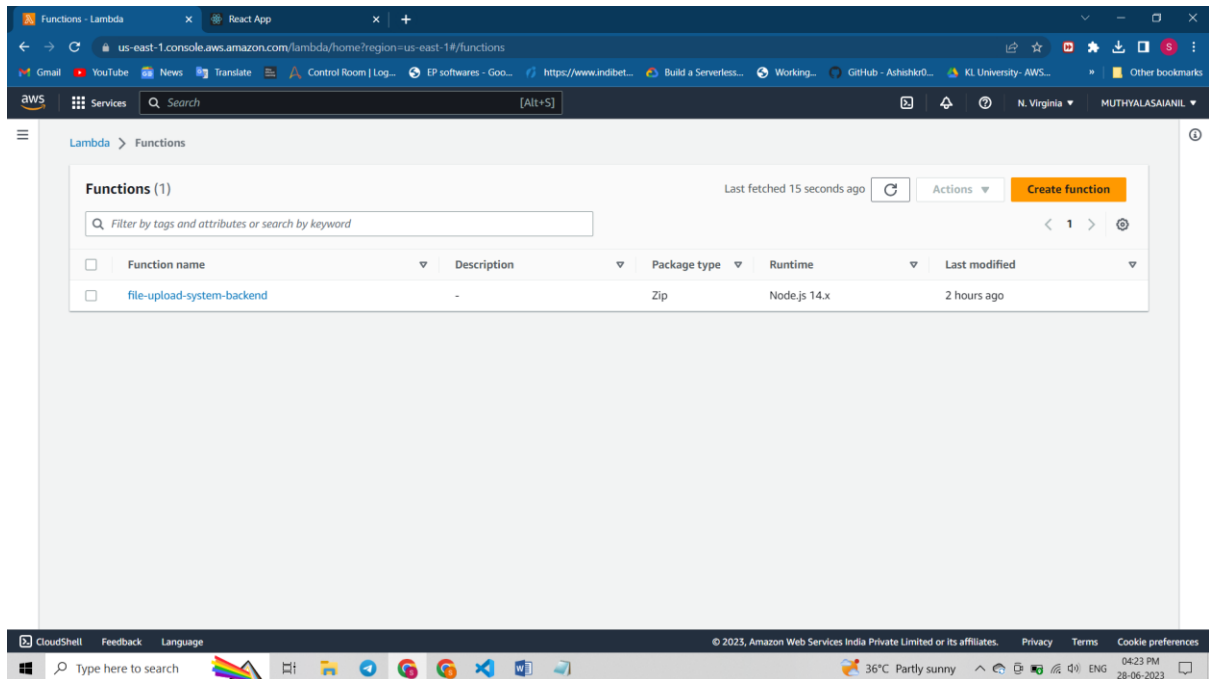


Here, IAM Role for Lambda with S3 and CloudWatchLogsFullAccess

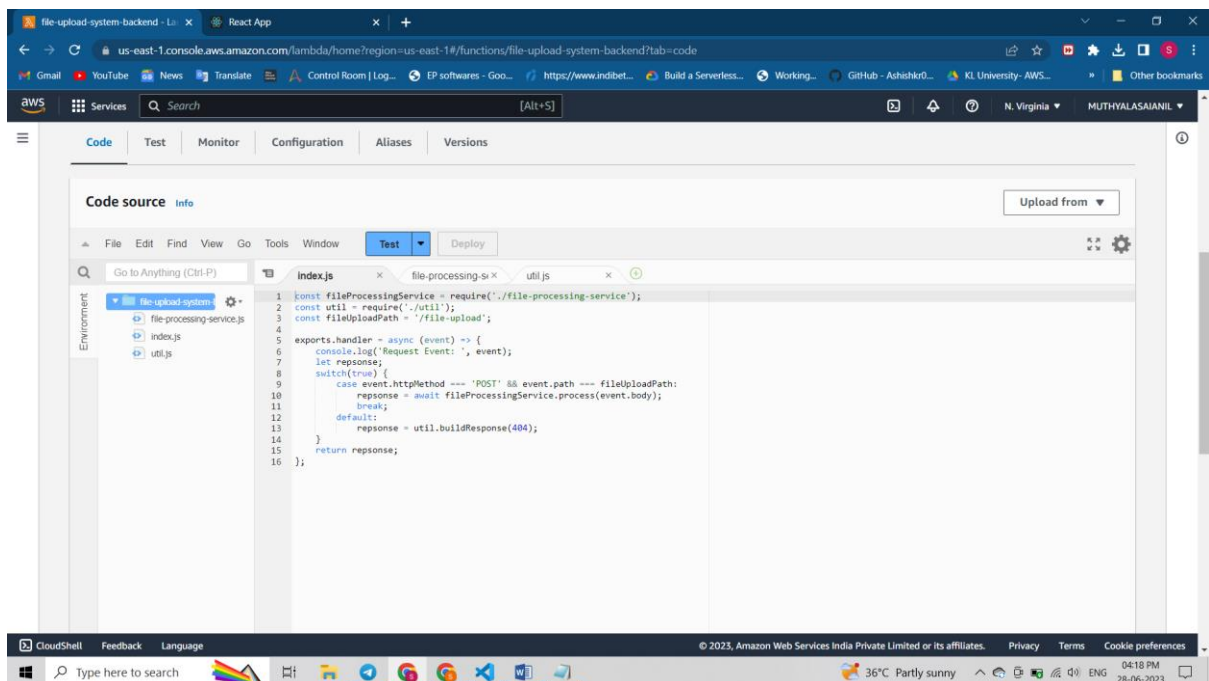


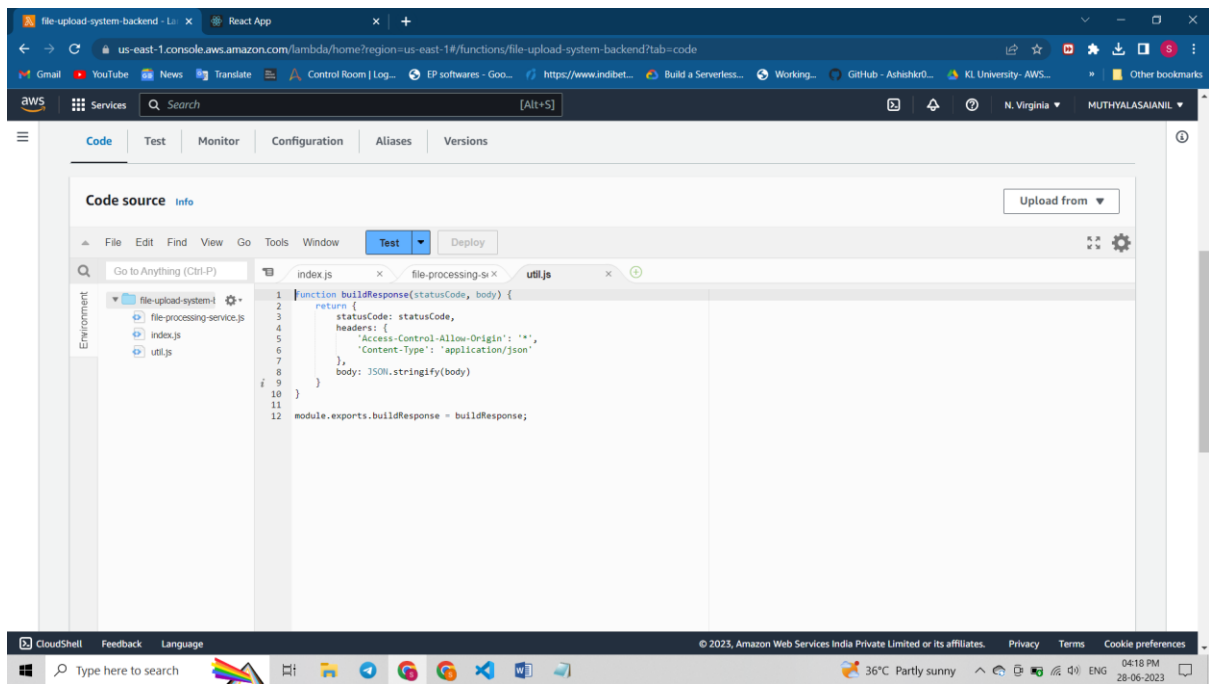
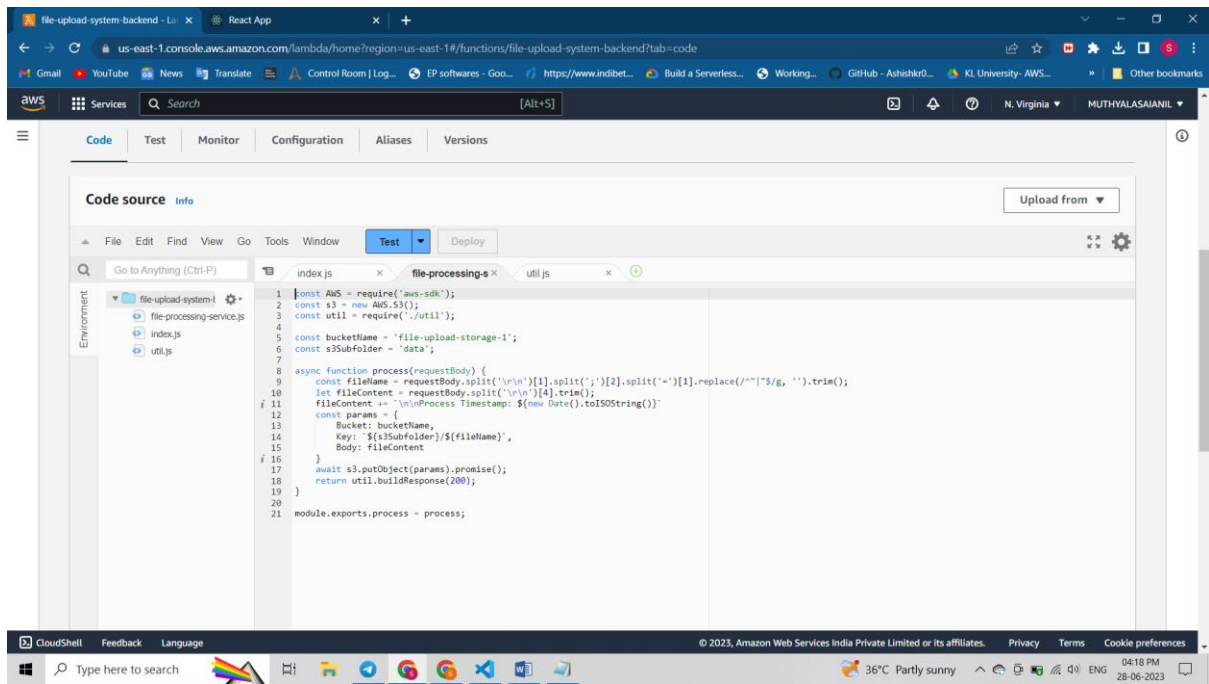
Backend Development:

The backend was implemented by using AWS Lambda and API Gateway.

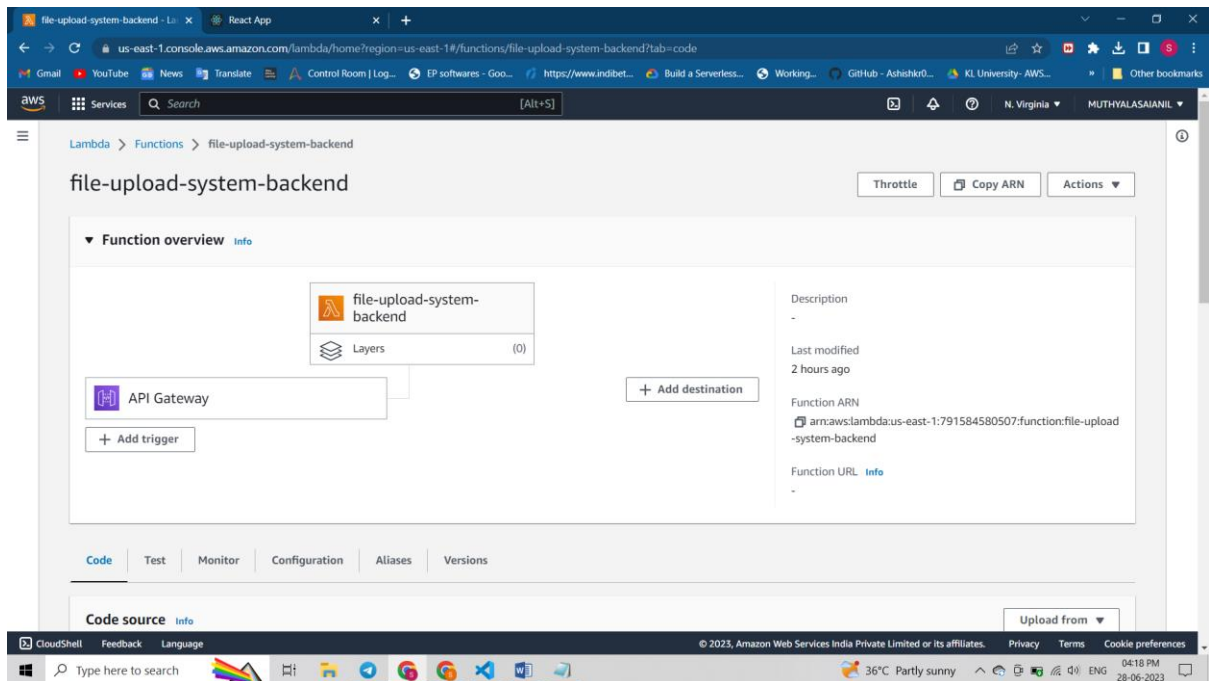


AWS Lambda functions are used to handle API requests for file uploads.

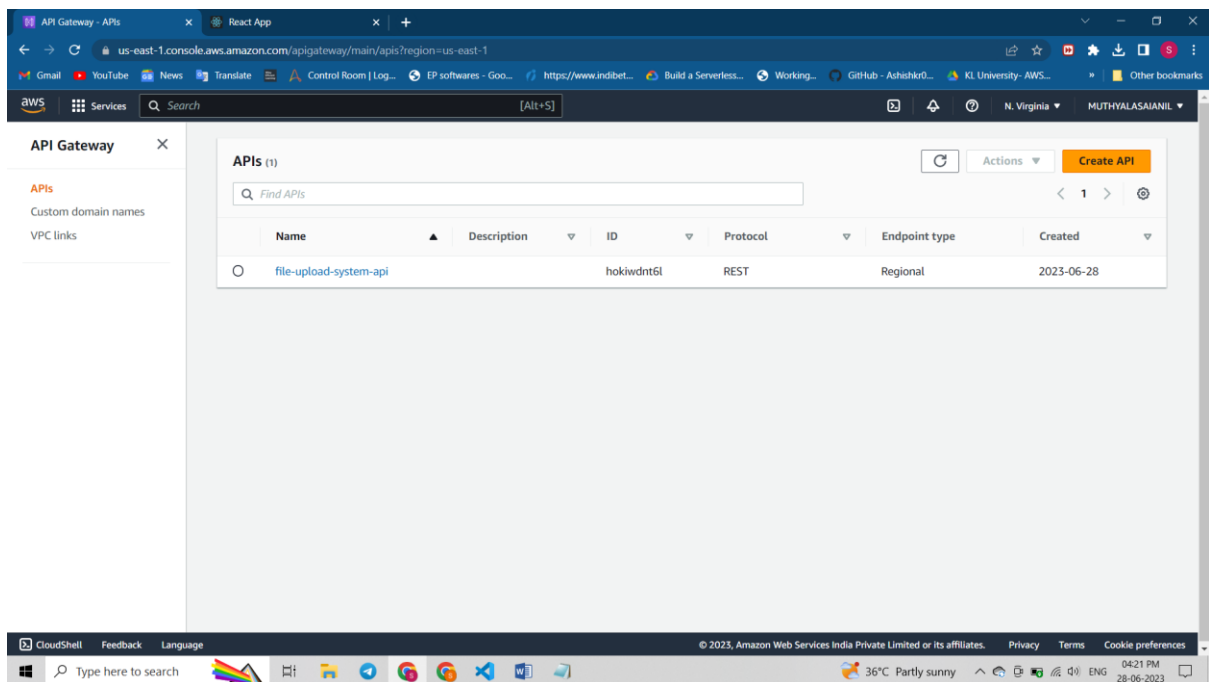




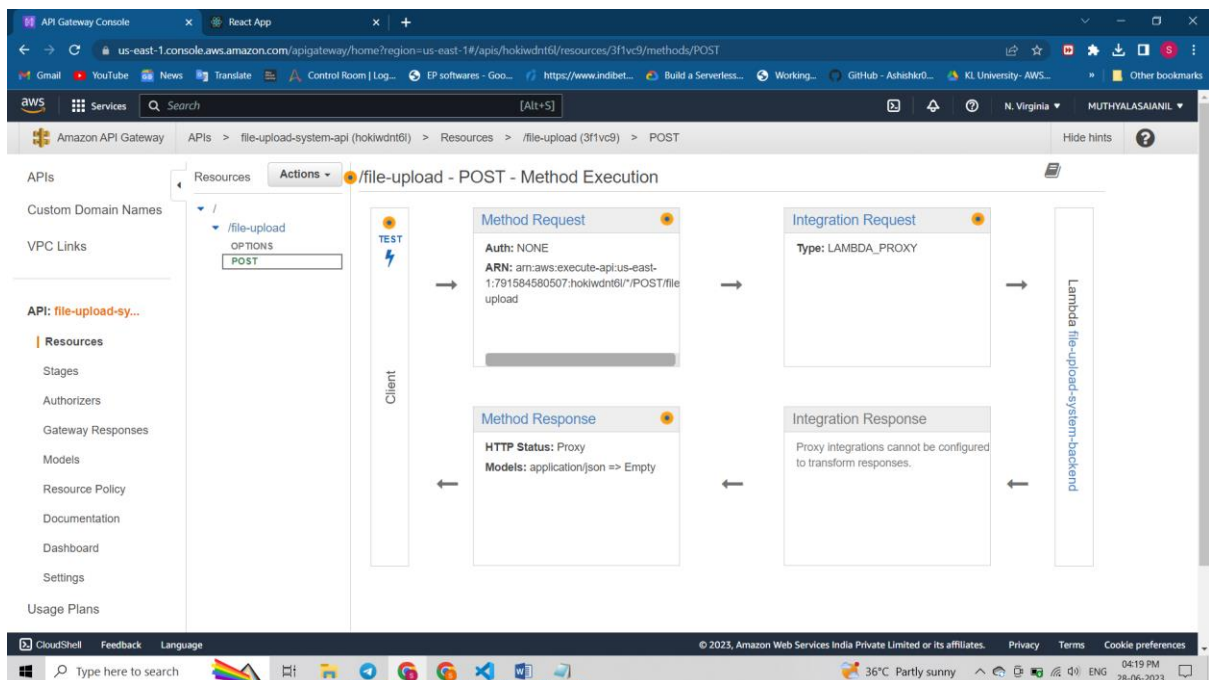
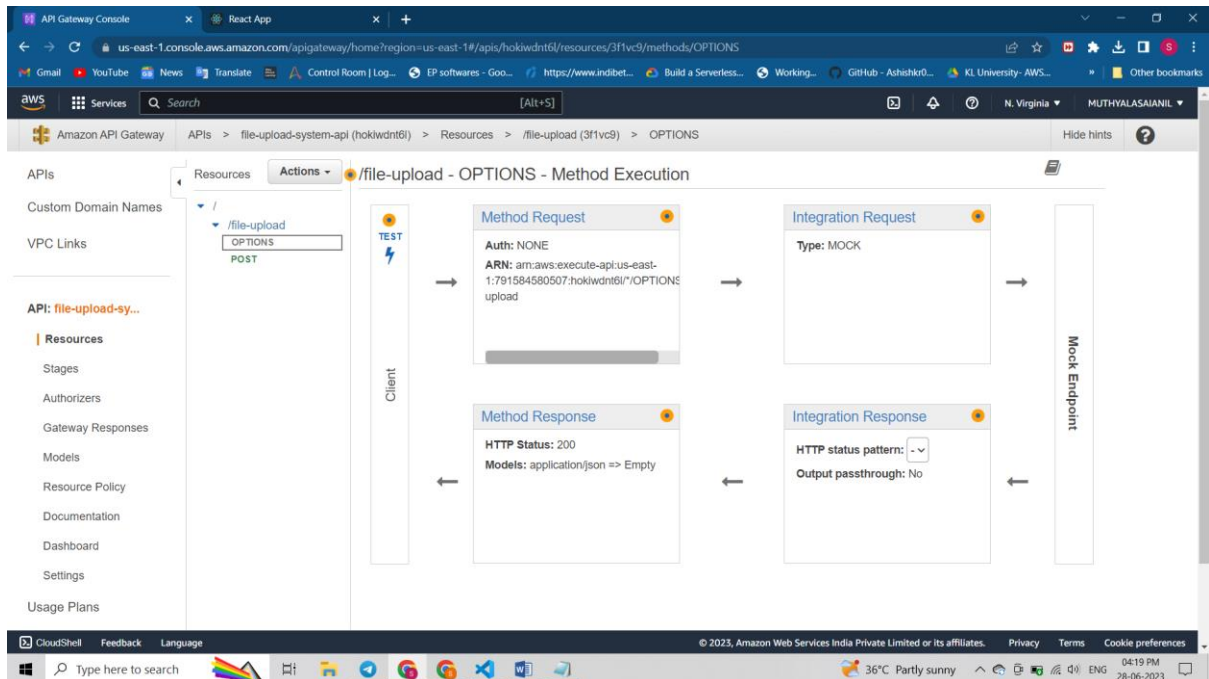
->It shows the connection established between Lambda function and API Gateway



->We created REST API Gateway function.



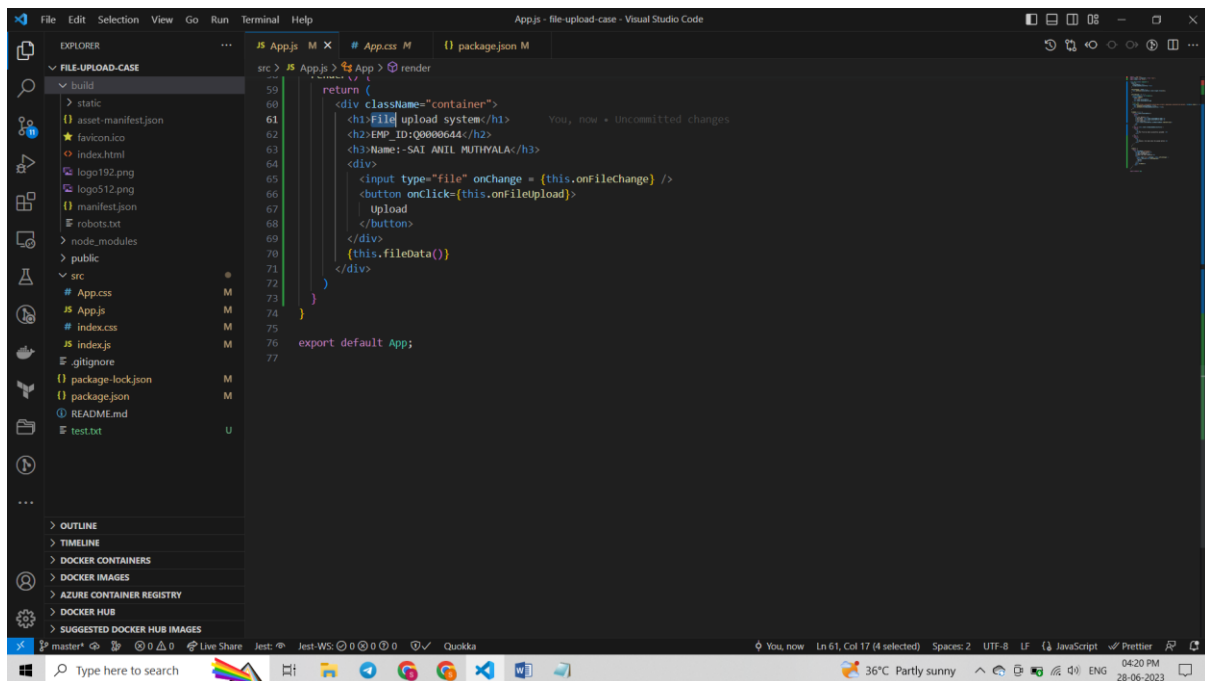
->API Gateway is used to create necessary API endpoints for frontend to communicate with backend.



Frontend using React.js:-

->Front end application was developed using React.js.

->It's a popular JavaScript framework used for building user Interface(UI).



->We need to build React build before uploading files(index.html) into S3 bucket.

