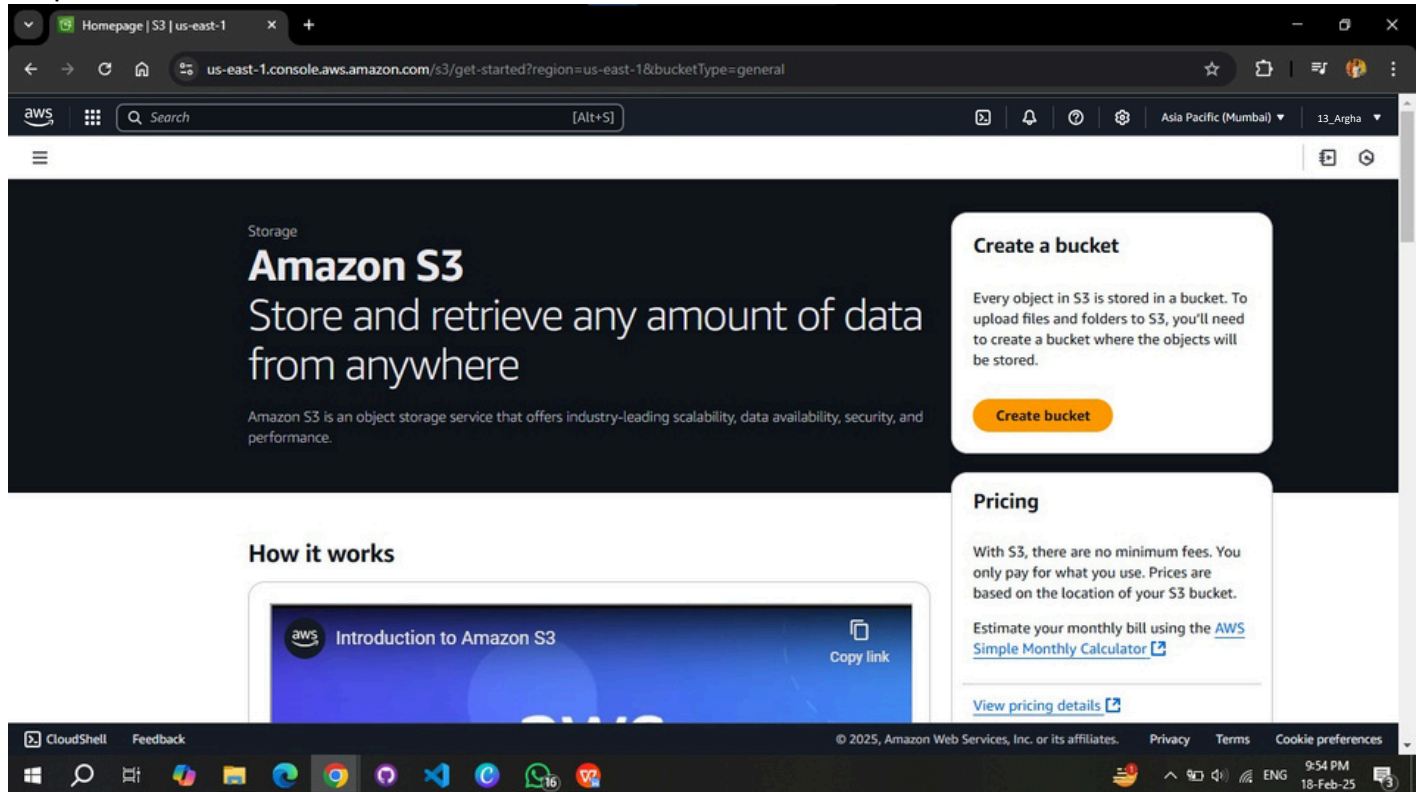


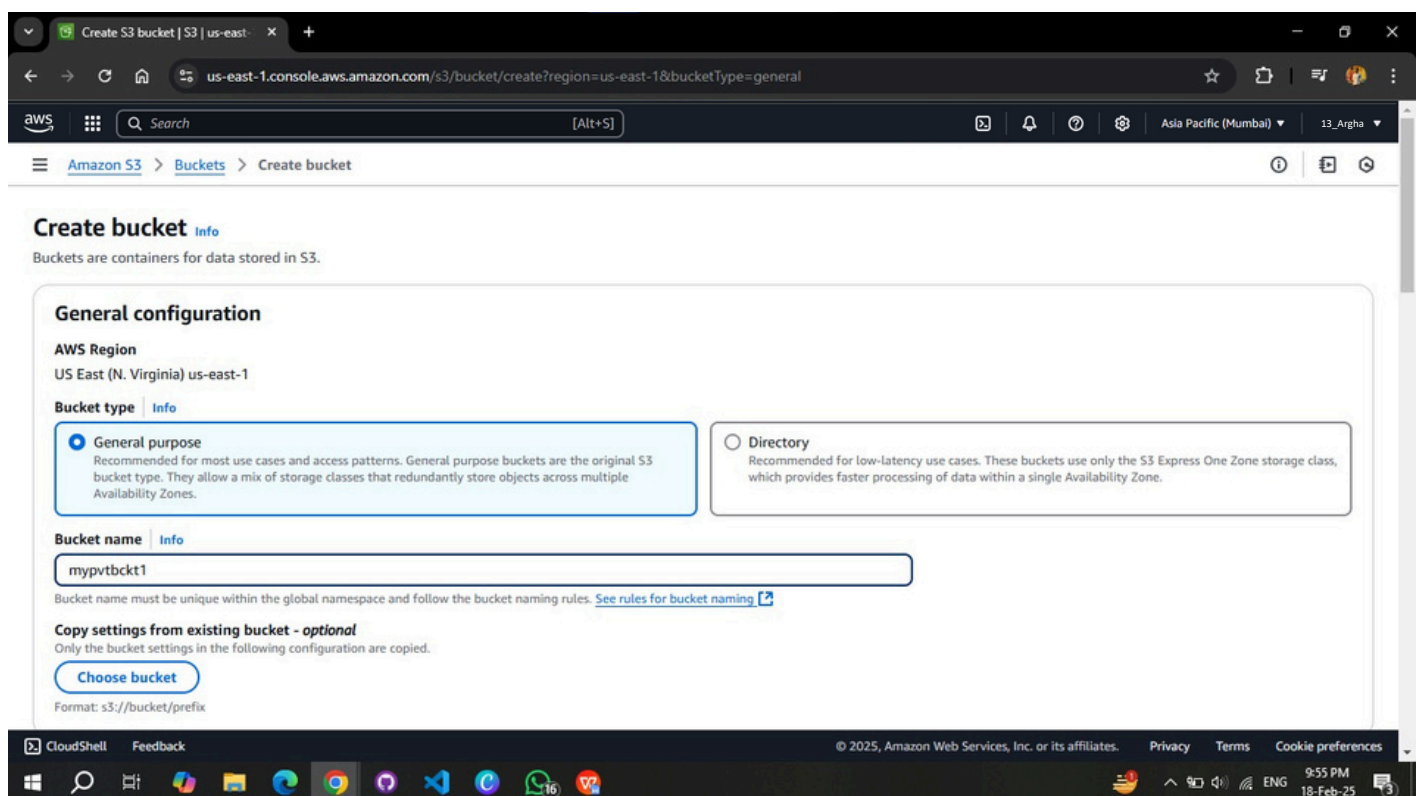
#### Assignment : 4

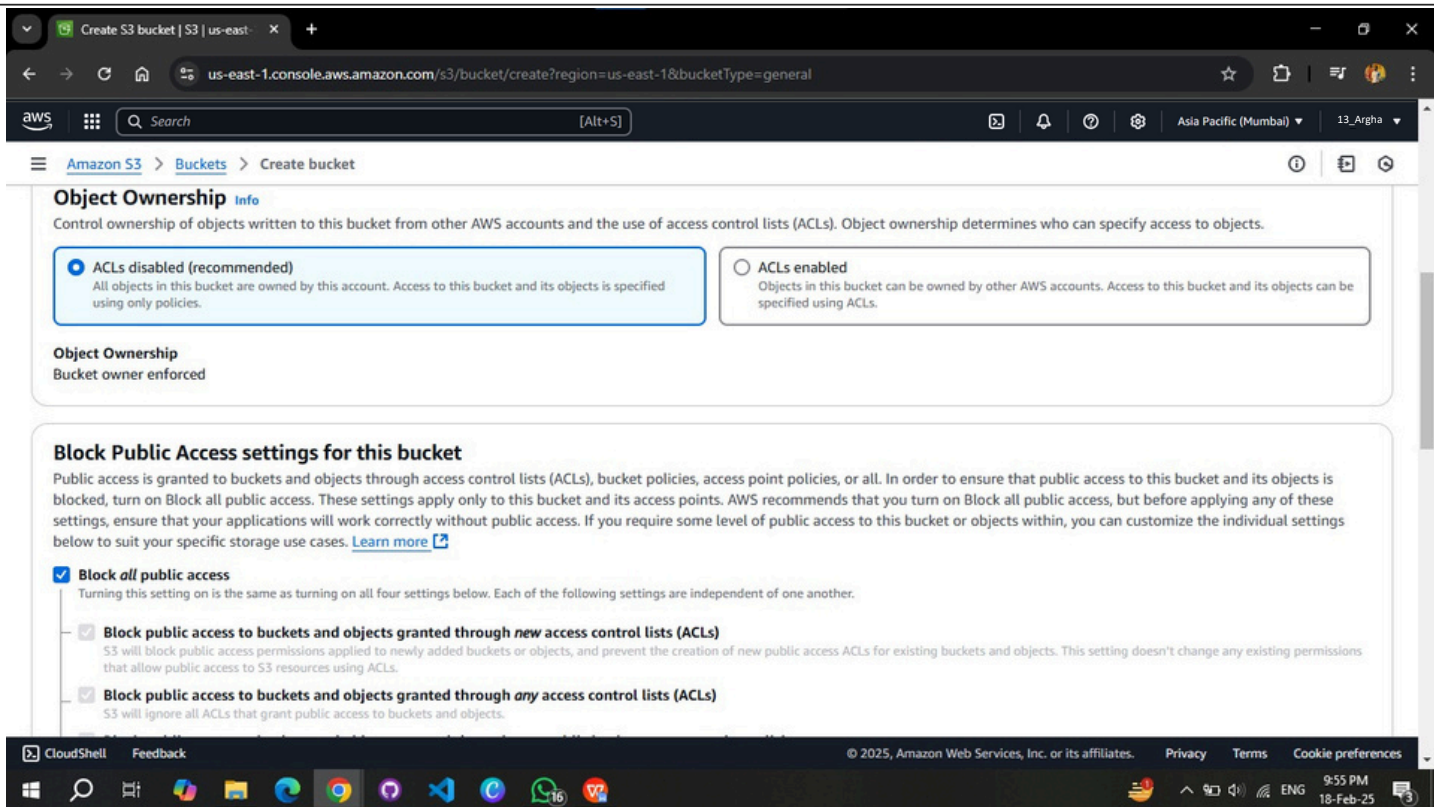
Create a private bucket in AWS. Upload a file and check by presigned URL whether you can access the file or not.

#### Step 1: Visit Amazon S3 and click on “Create Bucket”

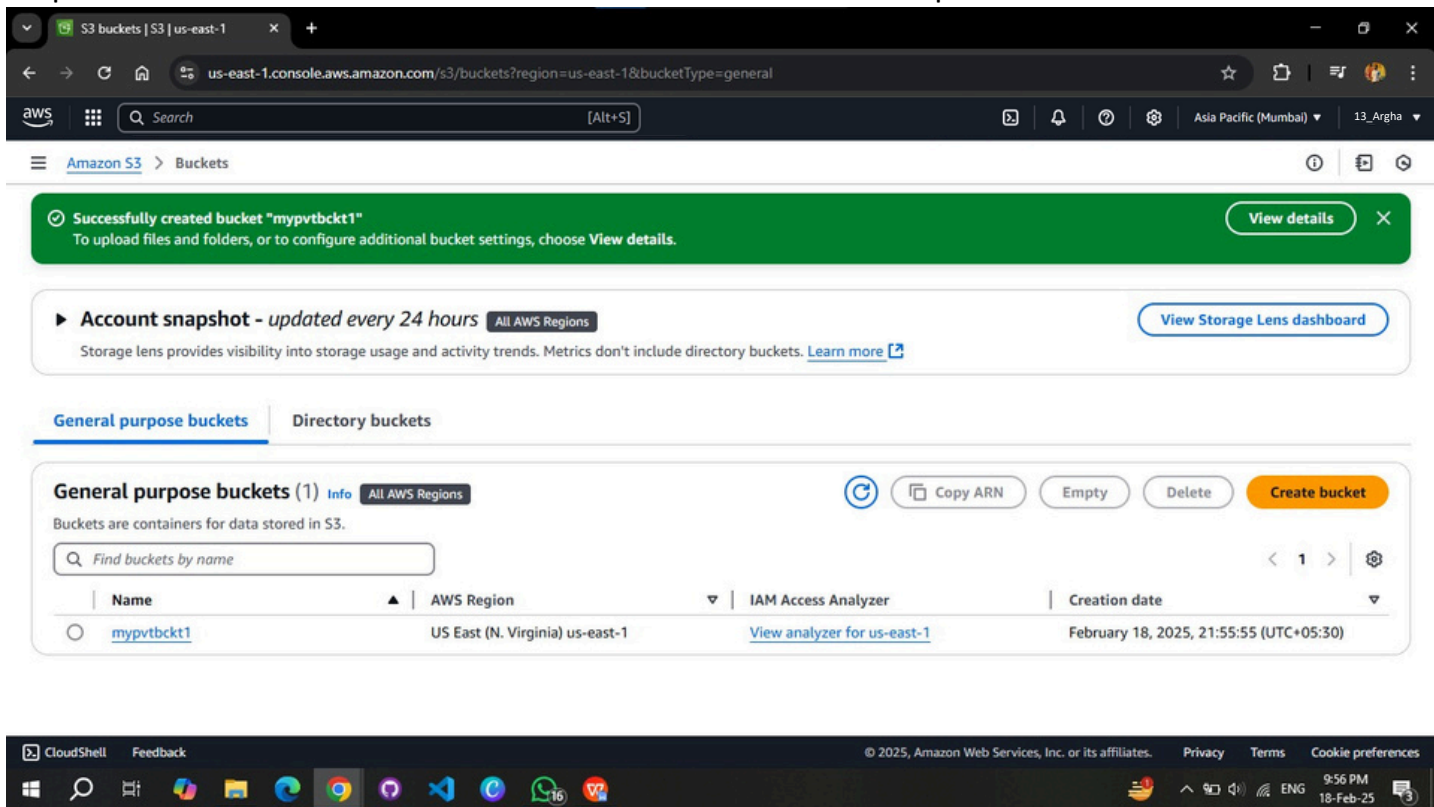


Step 2: In the Create bucket section, firstly name the bucket and keep ‘Block all public access’ box checked to keep the bucket private. Leave all other options as default and click on Create Bucket at the end of the screen.





Step 3: After creation of the bucket click on the bucket name to open it.



Step 4: Click on 'Upload' button. And then click on 'Add files' button and add a file.

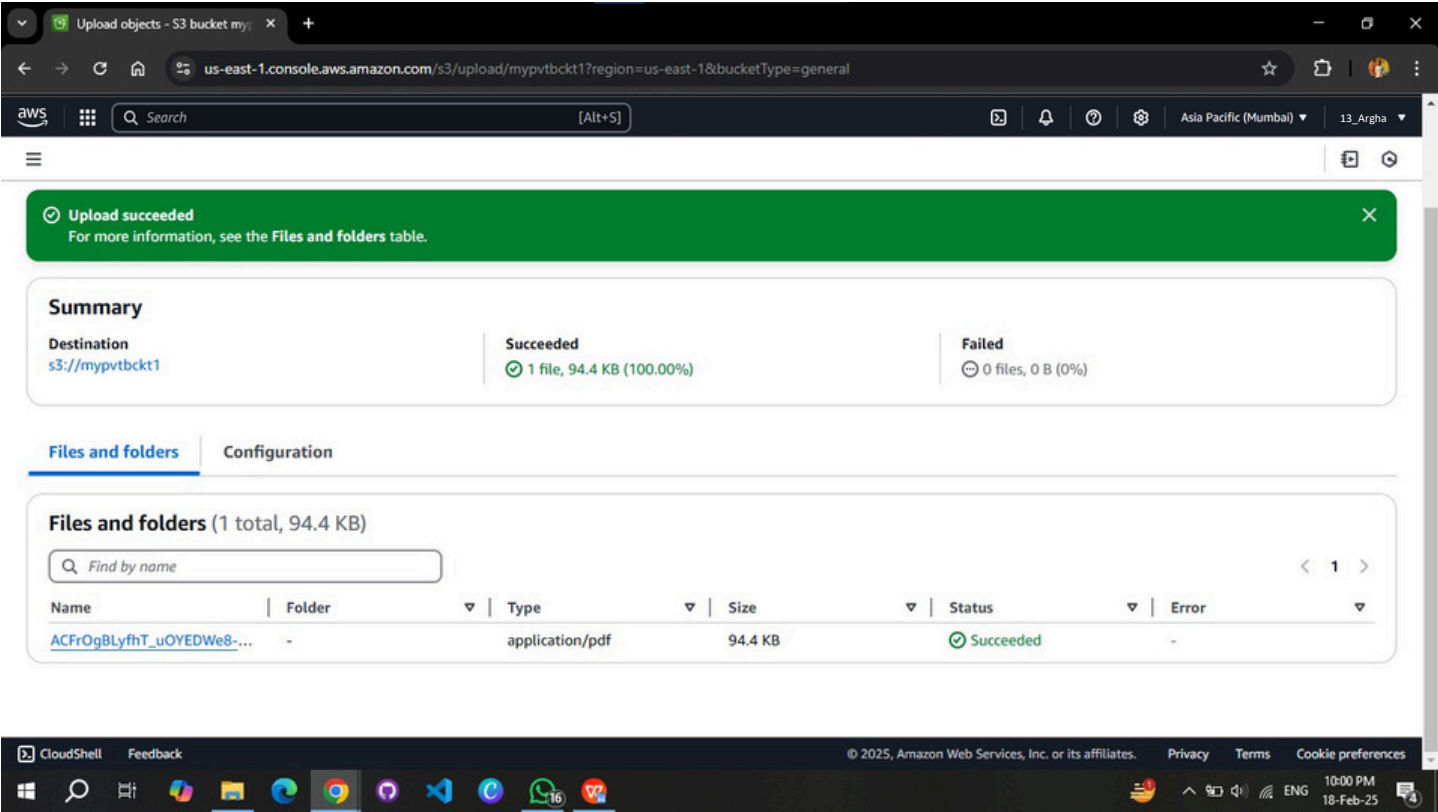
The screenshot shows the AWS S3 console's 'Upload' page for a bucket named 'mypvtbckt1'. The page has a breadcrumb trail: 'Amazon S3 > Buckets > mypvtbckt1 > Upload'. Below the breadcrumb, there's an 'Upload' section with an 'Info' icon. It contains a message: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDKs or Amazon S3 REST API. [Learn more](#)'. Below this is a dashed box with the text: 'Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.' Underneath is a section titled 'Files and folders (0)' with a 'Remove' button and 'Add files' and 'Add folder' buttons. A message says: 'All files and folders in this table will be uploaded.' There's a search bar with 'Find by name' and a table with columns: Name, Folder, Type, and Size. The table is empty, with a message: 'No files or folders. You have not chosen any files or folders to upload.' Below the table is a 'Destination' section with an 'Info' icon, showing the destination as 's3://mypvtbckt1'. The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock showing 9:59 PM on 18-Feb-25.

Step 5: After adding the file, click on 'Upload' to upload the selected file.

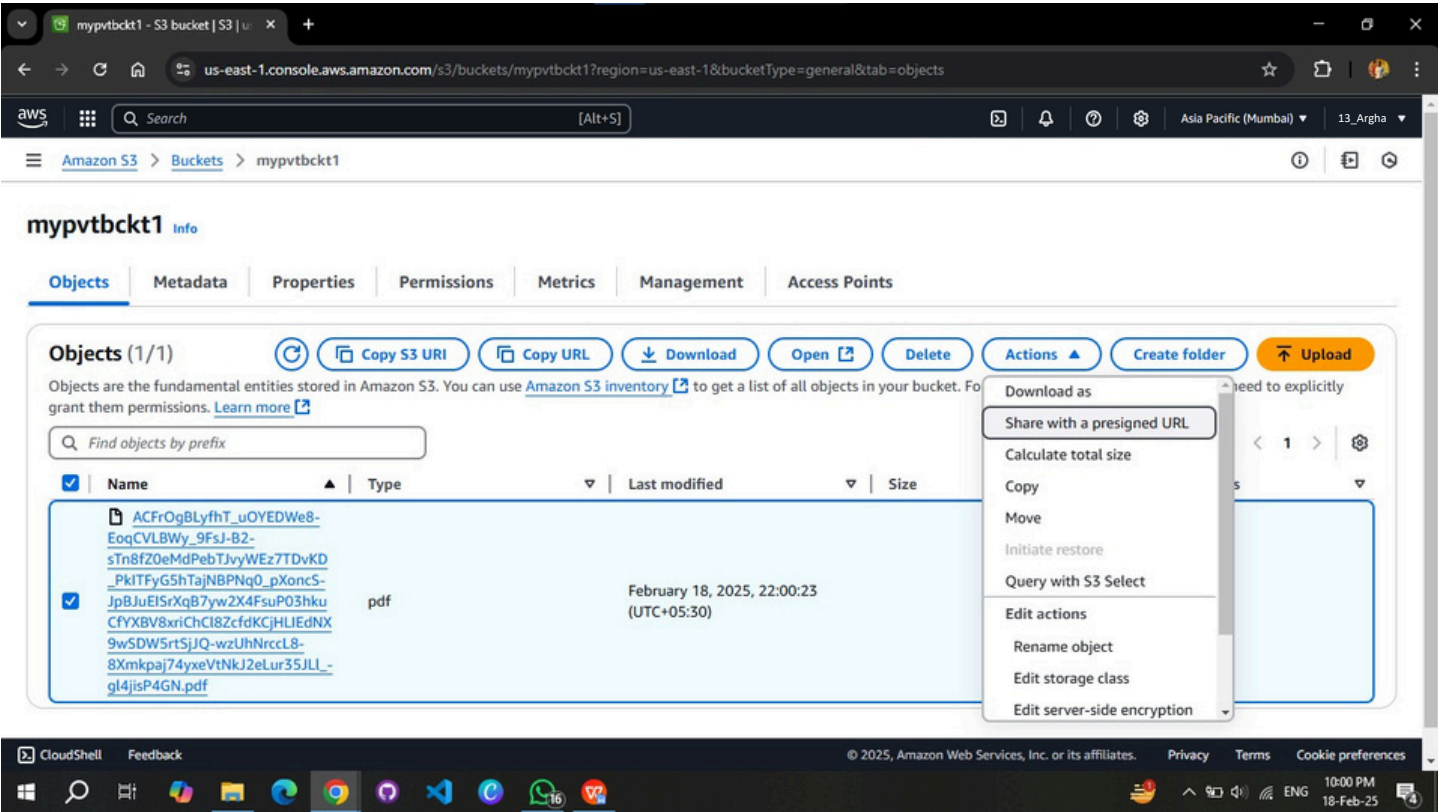
This screenshot shows the same AWS S3 console 'Upload' page, but now a file has been added. The 'Files and folders' section now shows '(1 total, 94.4 KB)'. The 'Add files' and 'Add folder' buttons are still present. The table now contains one entry: a file named 'ACFrOgBLyfhT\_uOYEDWe8-EoqCVLBWY...' with a type of 'application/pdf' and a size of '94.4 KB'. The 'Destination' section remains the same, showing 's3://mypvtbckt1'. The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock showing 10:00 PM on 18-Feb-25.



Step 6: Click on close after the upload is successful.



Step 7: Select the uploaded file and in the Actions bar, select 'Share with a presigned URL'



Step 8: Set the Time limit upto which we want our file to be accessible and click on 'Create presigned URL'. And the copy it.

The screenshot shows the AWS Management Console for the 'mypvtbckt1' bucket. A green notification banner at the top states: "A presigned URL for 'ACFrOgBlyfT\_uOYEDWe8-EoqCVLBWY\_9FsJ-B2-sTn8fZ0eMdPebTjvyWEz7TDvKD\_PkiTFyG5hTajNBPnQ0\_pXoncS-JpBjuEISrXqB7yw2X4FsuP03hkuCFYXBV8xriChCl8ZcfdKJHLIEdNX9wSDW5rt5JlQ-wzUhNrccL8-8Xmkpaj74yxeVtNkJ2eLur35JLL-gl4jisP4GN.pdf' has been created and copied to your clipboard." A 'Copy presigned URL' button is visible in the notification. Below the notification, the 'Presigned URL copied' status is confirmed. The 'Objects' tab is selected, showing a list of objects. The object 'ACFrOgBlyfT\_uOYEDWe8-EoqCVLBWY\_9FsJ-B2-sTn8fZ0eMdPebTjvyWEz7TDvKD\_PkiTFyG5hTajNBPnQ0\_pXoncS-JpBjuEISrXqB7yw2X4FsuP03hkuCFYXBV8xriChCl8ZcfdKJHLIEdNX.pdf' is highlighted. The console's top navigation bar shows 'Amazon S3 > Buckets > mypvtbckt1'. The bottom taskbar shows various application icons and the system clock at 10:01 PM on 18-Feb-25.

Step 9: Paste the copied URL in a new window or browser to check if the file is accessible or not within the time limit.

The screenshot shows a web browser window displaying a document. The address bar shows the presigned URL: 'mypvtbckt1.s3.us-east-1.amazonaws.com/ACFrOgBlyfT\_uOYEDWe8-EoqCVLBWY\_9FsJ-B2-sTn8fZ0eMdPebTjvyWEz7TDvKD\_PkiTFyG5hTajNBPnQ0\_pXoncS-JpBjuEISrXqB7yw2X4FsuP03hkuCFYXBV8xriChCl8ZcfdKJHLIEdNX.pdf'. The document content includes the header 'MCKV Institute of Engineering' and '243 G. T. Road (N), Liluah, Howrah - 711204'. Below this, the subject is 'Software Development and IT Operations Lab' and the code is 'PC-CS-694'. The stream is 'Credit: 1' and the year and semester are 'A.Y. 2024-2025'. A table with 4 columns (Experiment No., Name of the Experiment, Marks Obtained, Signature of the Class Teacher) and 11 rows is displayed. The first row is for '1. Create an account in AWS and configure a budget.' The second row is for '2. Create MFA for authentication.' The third row is for '3. Create IAM user and give full access to S3.' The fourth row is for '4. Create a private bucket in AWS. Upload a file and check by presigned URL whether you can access the file or not.' The fifth row is for '5. Create a public Bucket in AWS. Upload a file and give the necessary permission to check whether the file URL is working.' The sixth row is for '6. Upload a static website on S3.' The seventh row is for '7. Hosting a website on EC2.' The eighth row is for '8. Deploy a project from a local machine to GitHub and vice versa.' The ninth row is for '9. Deploy a project from GitHub to EC2.' The tenth row is for '10. Deploy a project from GitHub to EC2 by creating a new security group and user data.' The eleventh row is for '11. Build scaling plans in AWS that balance the'. The browser's bottom taskbar shows various application icons and the system clock at 10:02 PM on 18-Feb-25.

Experiment No.	Name of the Experiment	Marks Obtained	Signature of the Class Teacher
1.	Create an account in AWS and configure a budget.		
2.	Create MFA for authentication.		
3.	Create IAM user and give full access to S3.		
4.	Create a private bucket in AWS. Upload a file and check by presigned URL whether you can access the file or not.		
5.	Create a public Bucket in AWS. Upload a file and give the necessary permission to check whether the file URL is working.		
6.	Upload a static website on S3.		
7.	Hosting a website on EC2.		
8.	Deploy a project from a local machine to GitHub and vice versa.		
9.	Deploy a project from GitHub to EC2.		
10.	Deploy a project from GitHub to EC2 by creating a new security group and user data.		
11.	Build scaling plans in AWS that balance the		

Step 10: After the set time limit, the file is inaccessible, so we've successfully created a private bucket.

