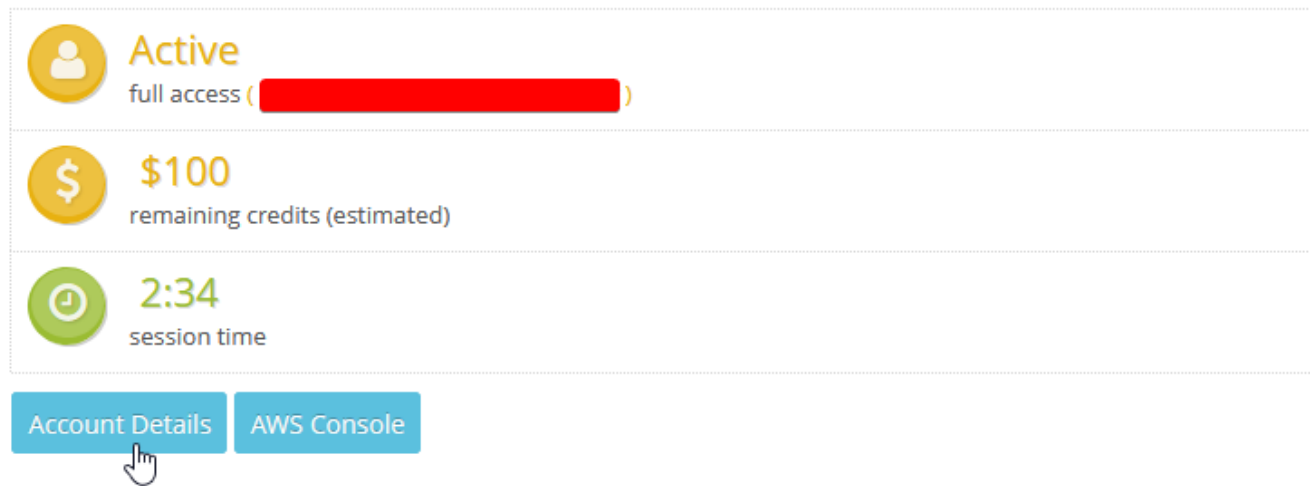


# Setup





## AWS Credentials

1. Login to aws educate account
2. Once on the Vocareum dashboard, click on [Account Details](#)

### Your AWS Account Status

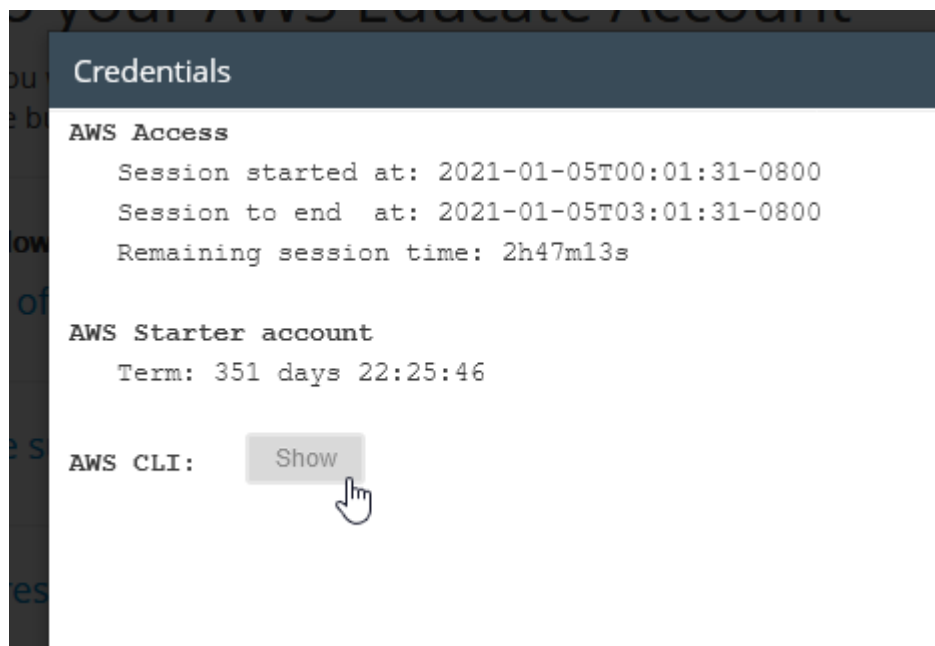


The screenshot shows the 'Your AWS Account Status' dashboard. It features three rows of status information: 1. Account Status: 'Active' with a person icon and 'full access' with a red progress bar. 2. Credits: '\$100' with a dollar sign icon and 'remaining credits (estimated)'. 3. Session Time: '2:34' with a clock icon and 'session time'. Below these rows are two buttons: 'Account Details' (highlighted with a mouse cursor) and 'AWS Console'.

	<b>Active</b> full access (  )
	<b>\$100</b> remaining credits (estimated)
	<b>2:34</b> session time

[Account Details](#) [AWS Console](#)

3. Click on show and copy contents into `~/.aws/credentials` (%UserProfile%\.aws/credentials)
4. Add the following line into the credentials file: `region=us-east-1`. Place this under the [default] config key



The screenshot shows a 'Credentials' window with a dark header. It contains three sections: 'AWS Access' with session start/end times and remaining time; 'AWS Starter account' with a term; and 'AWS CLI:' with a 'Show' button (highlighted with a mouse cursor).

```
Credentials

AWS Access
  Session started at: 2021-01-05T00:01:31-0800
  Session to end   at: 2021-01-05T03:01:31-0800
  Remaining session time: 2h47m13s

AWS Starter account
  Term: 351 days 22:25:46

AWS CLI:  Show
```

## AWS S3 Setup

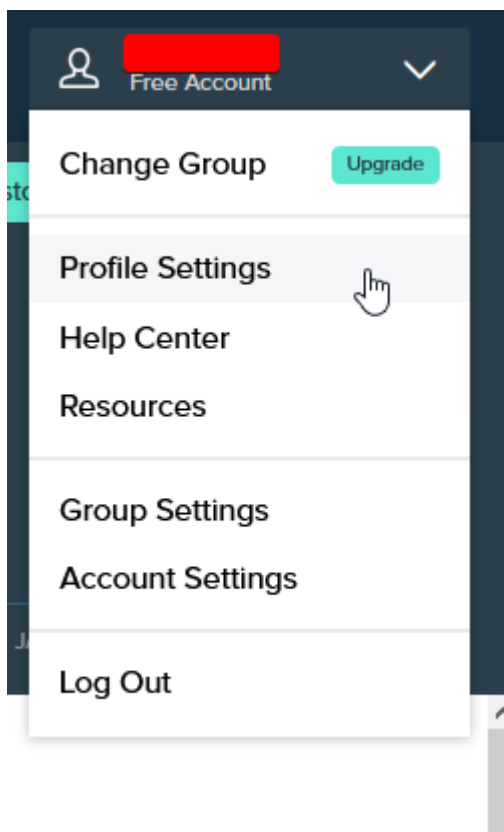
1. Click [AWS Console](#) and navigate over to [S3](#)

2. Create a new bucket in **us-east-1**
3. Turn off **Block all public access** and create the bucket
4. Click on the newly create bucket, go to **Permissions** and scroll down to **Bucket policy** and edit it to:

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicRead",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Resource": "YOUR_BUCKET_ARN_HERE/*"
    }
  ]
}
```

## Bitly API Key

1. Register/login for **Bitly**
2. Click **Profile Settings > Generic Access Token**. Follow the instructions there and create a access token



## Project Configuration

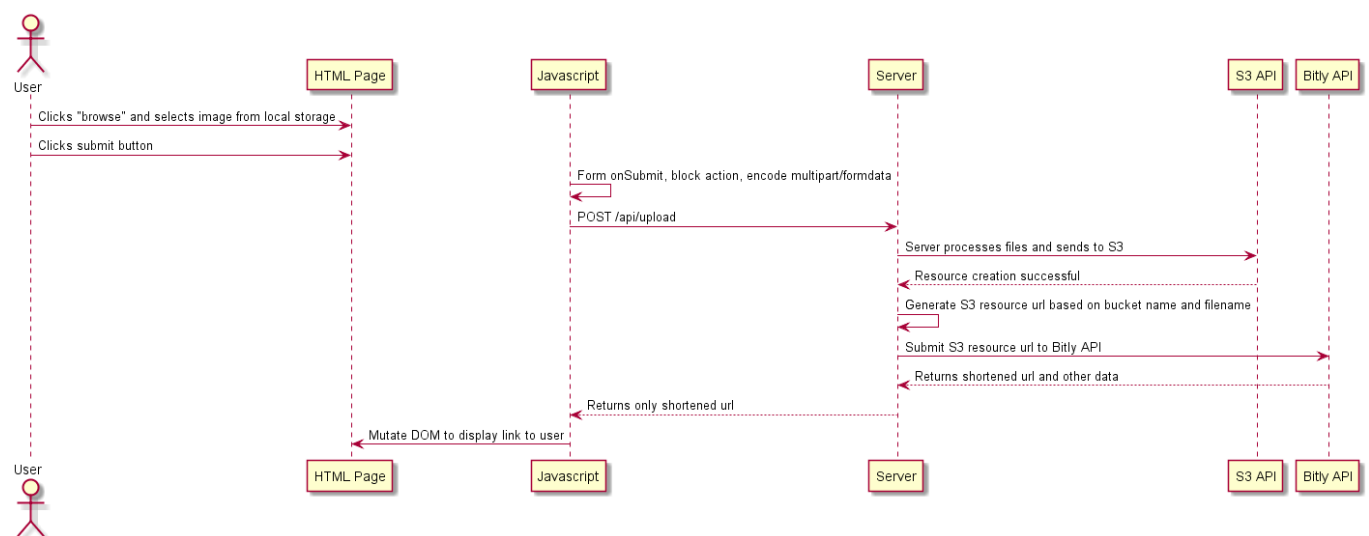
In `solution/task5/task5/` create a file `.env` with keys:

```
BITLY_TOKEN=YOUR_BITLY_TOKEN
AWS_BUCKET=YOUR_AWS_BUCKET_NAME
```

## Usage

Navigate to `/Uploader/Index`, select image and upload it. A download link should appear above the form, errors will be shown in an alert box.

## Sequence Diagram



## API

### Upload Image to S3 and Shorten

`POST /api/Upload`

#### Request

Accepts multipart/form-data with a single file

#### Response

A single string that is a bitly shortened url pointing to the uploaded resource in s3.

Example:

```
https://bit.ly/...
```

## Error

400 Bad Request. If the submitted file is above 2mb or is not an image

## Additional features

---

Verify file is an image and is a reasonable size (server)

```
var file = HttpContext.Current.Request.Files?[0];

if (file != null && file.ContentLength > 0)
{
    string filename = Path.GetFileName(file.FileName);

    if (file.ContentLength > maxSize)
        return BadRequest($"File exceeds size of {maxSize}");

    if (!file.ContentType.StartsWith("image"))
        return BadRequest("Submitted file must be an image");
}
```

Validate file is an image (client)

```
const files = $("#file-input")[0].files;
if (files.length != 1)
{
    alert("Submit a single file");
    e.preventDefault();
    return;
}

if (!files[0].type.startsWith("image/"))
{
    alert("Submitted file must be an image");
    e.preventDefault();
    return;
}
```