

How to Run the project given in the Assignment

- For Development of the API's Django Framework is used.
- Open The Assignment Folder with any IDE
- Create Virtual Environment (Python Version == 3.13.0)
- Navigate to Python Task directory
`cd url_shortener`
- Install the requirements for the project from requirements.txt file
`pip install -r requirements.txt`
- Make Migrations. Use Below commands.
`python3 manage.py makemigrations`
`python3 manage.py migrate`
- Now run the project with the below command
`python3 manage.py runserver`
- If all the steps are correctly followed the project will be up and running.
- To test the API's follow this link
<https://documenter.getpostman.com/view/29643947/2sAYQamrMu>
This link contains all the API's with json request preloaded
- Change your environment url according to your system.
- Please note when the shortened url is generated like this
"shortened_url": "https://short.ly/a24a5b",
Use a24a5b in the next api for redirecting or viewing the analytics which will be further explained in api documentation in the above link or below pages.

For any queries feel free to contact
niteshmauryac7r.nm@gmail.com
9820187742

URL Shortener API Documentation

Base URL:

All API endpoints are available at: <https://yourdomain.com>

1. POST /shorten

Create a shortened URL

- **Description:** This endpoint takes a long URL and creates a shortened URL. Optionally, you can set an expiration time (in hours) and provide an optional password for protection.
- **Request:**

Headers:

- Content-Type: application/json

Body:

```
{  
  "url": "https://www.example.com",  
  "expiry": 24,  
  "password": "yourpassword"  
}
```

- url (string, required): The original long URL to shorten.
- expiry (integer, optional): The expiration time of the shortened URL in hours. Default is 24 hours.
- password (string, optional): An optional password for protecting the shortened URL.

- **Response:**

- **Success (200 OK):**

```
{  
  "shortened_url": "https://short.ly/abc123",  
  "expiry": "2025-01-20T12:00:00Z"  
}
```

- shortened_url (string): The generated shortened URL.
- expiry (string): The expiration timestamp of the shortened URL.

- **Error (400 Bad Request):**

```
{  
  "error": "Invalid URL format."  
}
```

Possible Errors:

- Invalid URL format: The provided URL is invalid.
 - Missing 'url' parameter: The url field is missing in the request body.
-

2. GET /{short_url}

Redirect to the original URL

- **Description:** This endpoint redirects the user to the original long URL corresponding to the shortened URL. If the URL has expired, the request will return an error.
- **Request:**
 - **URL Parameter:**
 - short_url (string, required): The shortened URL identifier (e.g., abc123).

- **Response:**

- **Success** (301 Moved Permanently): The user will be redirected to the original URL (e.g., <https://www.example.com>).
- **Error** (404 Not Found):

```
{
  "error": "This URL has expired."
}
```

Possible Errors:

- This URL has expired: The shortened URL has expired and cannot be accessed.
- **Error** (403 Forbidden):

```
{
  "error": "Invalid password."
}
```

Possible Errors:

- Invalid password: The password provided (if any) does not match the stored password for the shortened URL.

3. GET /analytics/{short_url}

Retrieve analytics data for a shortened URL

- **Description:** This endpoint provides the analytics for a shortened URL, including the number of accesses and the IP addresses that accessed the URL, along with timestamps.
- **Request:**
 - **URL Parameter:**
 - `short_url` (string, required): The shortened URL identifier (e.g., `abc123`).
- **Response:**
 - **Success** (200 OK):

```
{
  "original_url": "https://www.example.com",
  "access_count": 5,
  "logs": [
    {
      "timestamp": "2025-01-19T12:00:00Z",
      "ip_address": "192.168.1.1"
    },
    {
      "timestamp": "2025-01-19T12:10:00Z",
      "ip_address": "192.168.1.2"
    }
  ]
}
```

- `original_url` (string): The original long URL.
- `access_count` (integer): The total number of times the shortened URL has been accessed.
- `logs` (array): List of access logs, with each log containing:
 - `timestamp` (string): The timestamp of the access.
 - `ip_address` (string): The IP address of the user who accessed the URL.

- **Error (404 Not Found):**

```
{  
  "error": "This URL has expired."  
}
```

Possible Errors:

- This URL has expired: The shortened URL has expired and no analytics can be fetched.

Error Handling

The API returns the following standard error responses:

- **400 Bad Request:** Invalid or missing parameters.
- **403 Forbidden:** Invalid password for protected URL.
- **404 Not Found:** URL does not exist or has expired.
- **500 Internal Server Error:** Unexpected server-side error.