**VJTI FORUMS**



**Developed By: -**

Shubh Ganatra-181090060

Shubham Jadhav-181090061

Shubham Agrawal-181090062

Atharva Wadekar-181090073

## What is Django?

Django is an open-source python web framework used for rapid development, pragmatic, maintainable, clean design, and secures websites. ... It takes care of a lot of hassle involved in the web development; enables users to focus on developing components needed for their application.

## Why do you need a framework?

To understand what Django is actually for, we need to take a closer look at the servers. The first thing is that the server needs to know that you want it to serve you a web page.

Imagine a mailbox (port) which is monitored for incoming letters (requests). This is done by a web server. The web server reads the letter and then sends a response with a webpage. But when you want to send something, you need to have some content, and Django is something that helps you create the content.

# ***Apps Used***

# **Blog:**

# What is the use of a blog page?

They can be used for anything that involves communicating or publishing information on the World Wide Web. Common uses include teaching and educational and corporate use. Your blog can be a personal diary, a project collaboration tool, a guide, or any means of communicating and publishing information on the web.

In our project it is used as a platform for asking and answering questions related to academic classwork and homework.

# **Files:**

# Managing files in Django

By default, Django stores files locally, using the **MEDIA\_ROOT** and **MEDIA\_URL** settings. The examples below assume that you’re using these defaults.

However, Django provides ways to write custom file storage systems that allow you to completely customize where and how Django stores files.

In our project we use this app to upload files, pdfs, excel sheets, and important notes to the site server so other users can access and download them.

# **Users:**

# User authentication in Django

The Django authentication system handles both authentication and authorization. Briefly, authentication verifies a user is who they claim to be, and authorization determines what an authenticated user is allowed to do. Here the term authentication is used to refer to both tasks.

Overview

The auth system consists of:

* Users
* Permissions: Binary (yes/no) flags designating whether a user may perform a certain task.
* Groups: A generic way of applying labels and permissions to more than one user.
* A configurable password hashing system
* Forms and view tools for logging in users, or restricting content
* A pluggable backend system

The authentication system in Django aims to be very generic and doesn’t provide some features commonly found in web authentication systems.

Solutions for some of these common problems have been implemented in third-party packages

* Password strength checking
* Throttling of login attempts
* Authentication against third-parties

(for example: OAuth)

* Object-level permissions

# ***Aim***

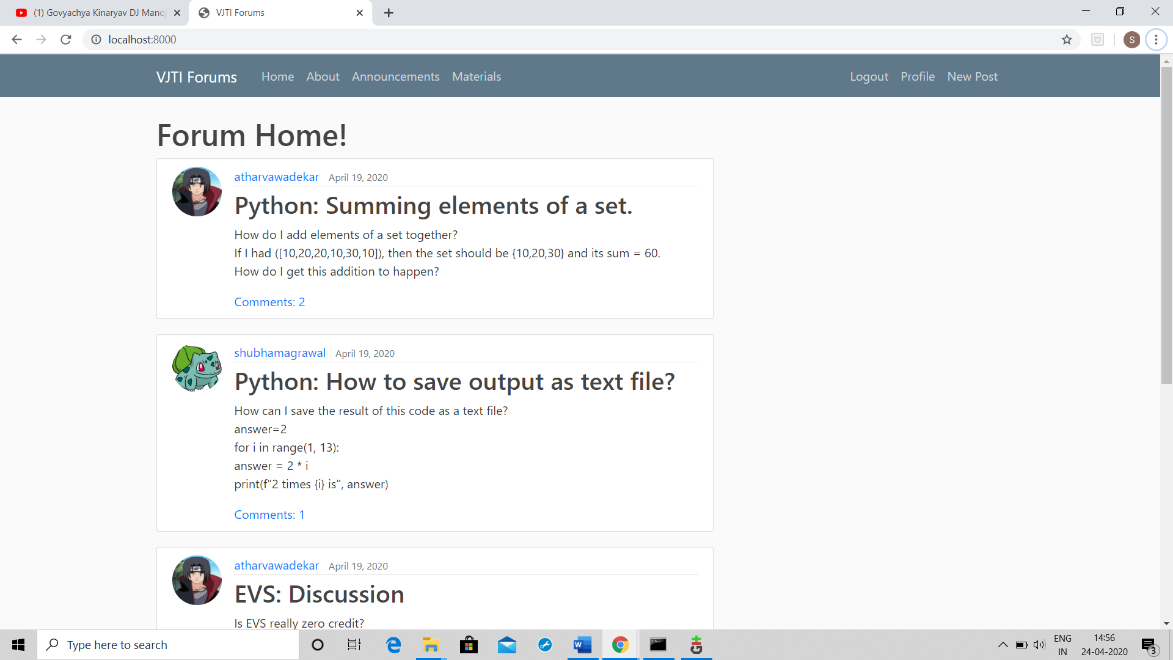
Creating an interface for students to interact and benefit academically.

# ***Features***

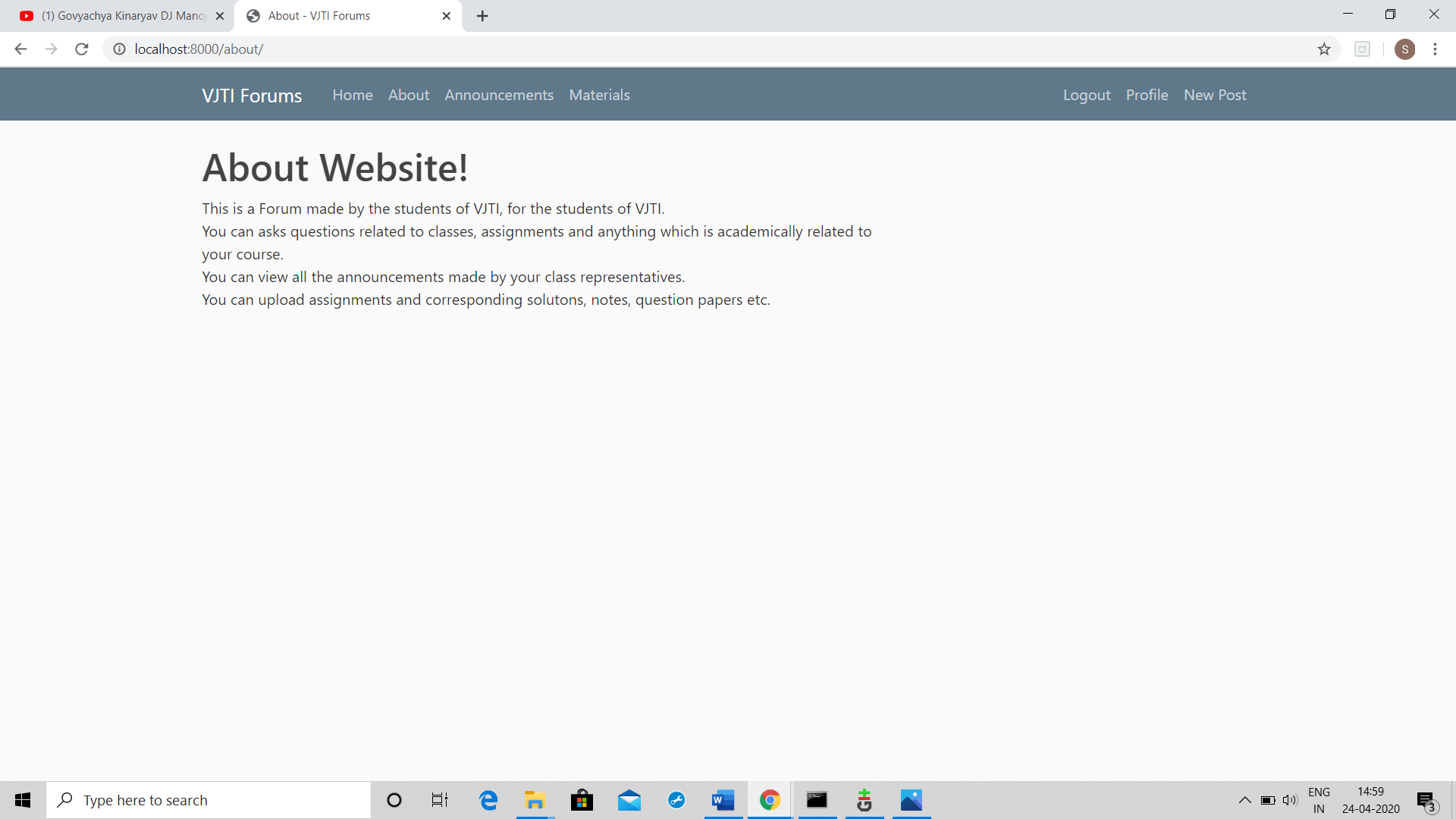
* Posts containing the queries and doubts of student users can be created and are displayed chronologically on the blog wall.
* Posts can also contain general opinions and discussion topics of thee users which they might like the views of others.
* The doubts can be solved by other users in comments of the post.
* New users can be easily registered and verified by the admins.
* Users can upload files related to the academic syllabus or in support off their answers to the servers and others can download these files.
* These posts can be verified by the admins and removed if they are irrelevant.
* For the important announcements the admins can post to the announcement wall which is restricted to be used by only the admins.

# ***Project Snapshots***

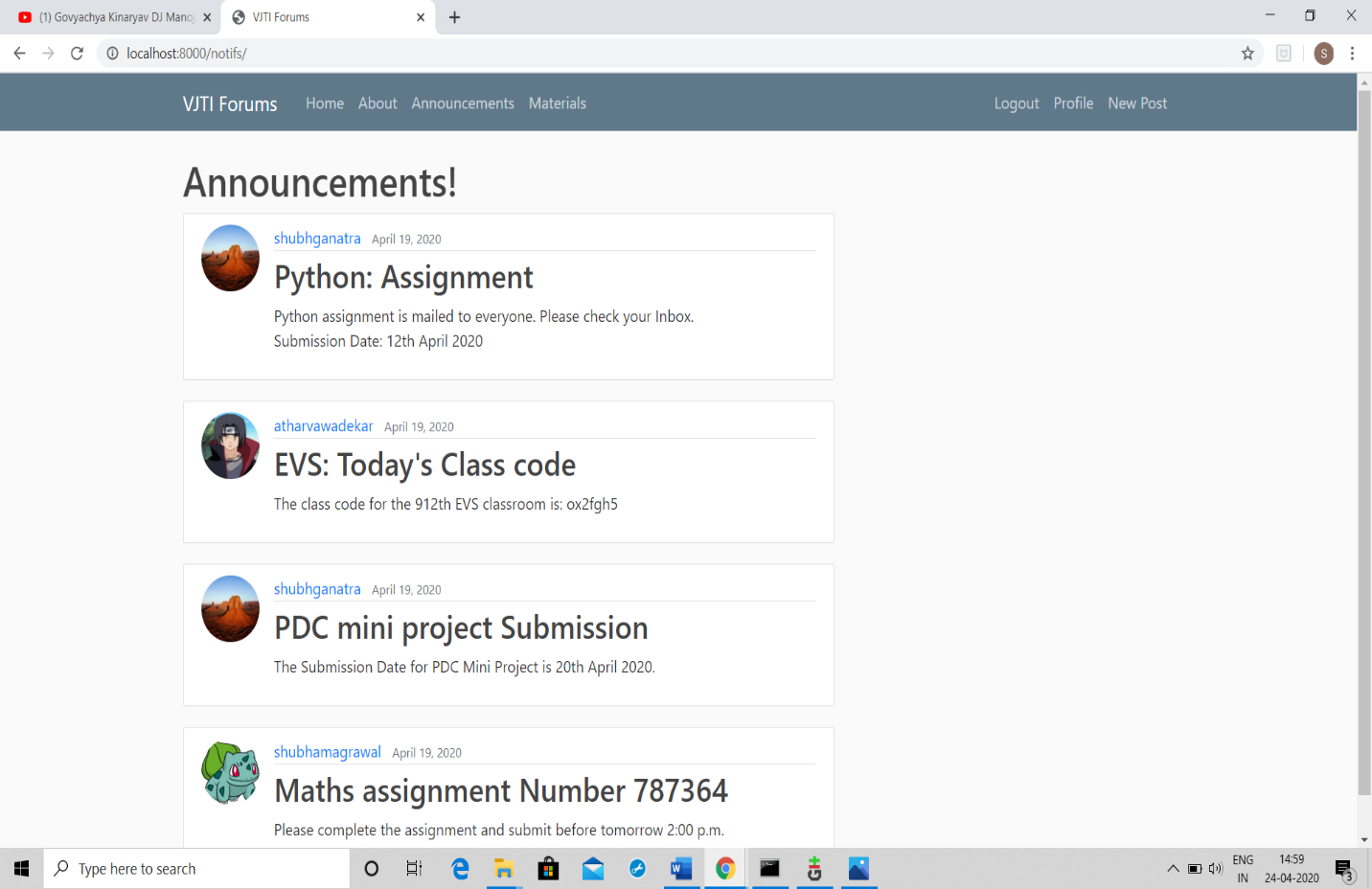
# **Home page**



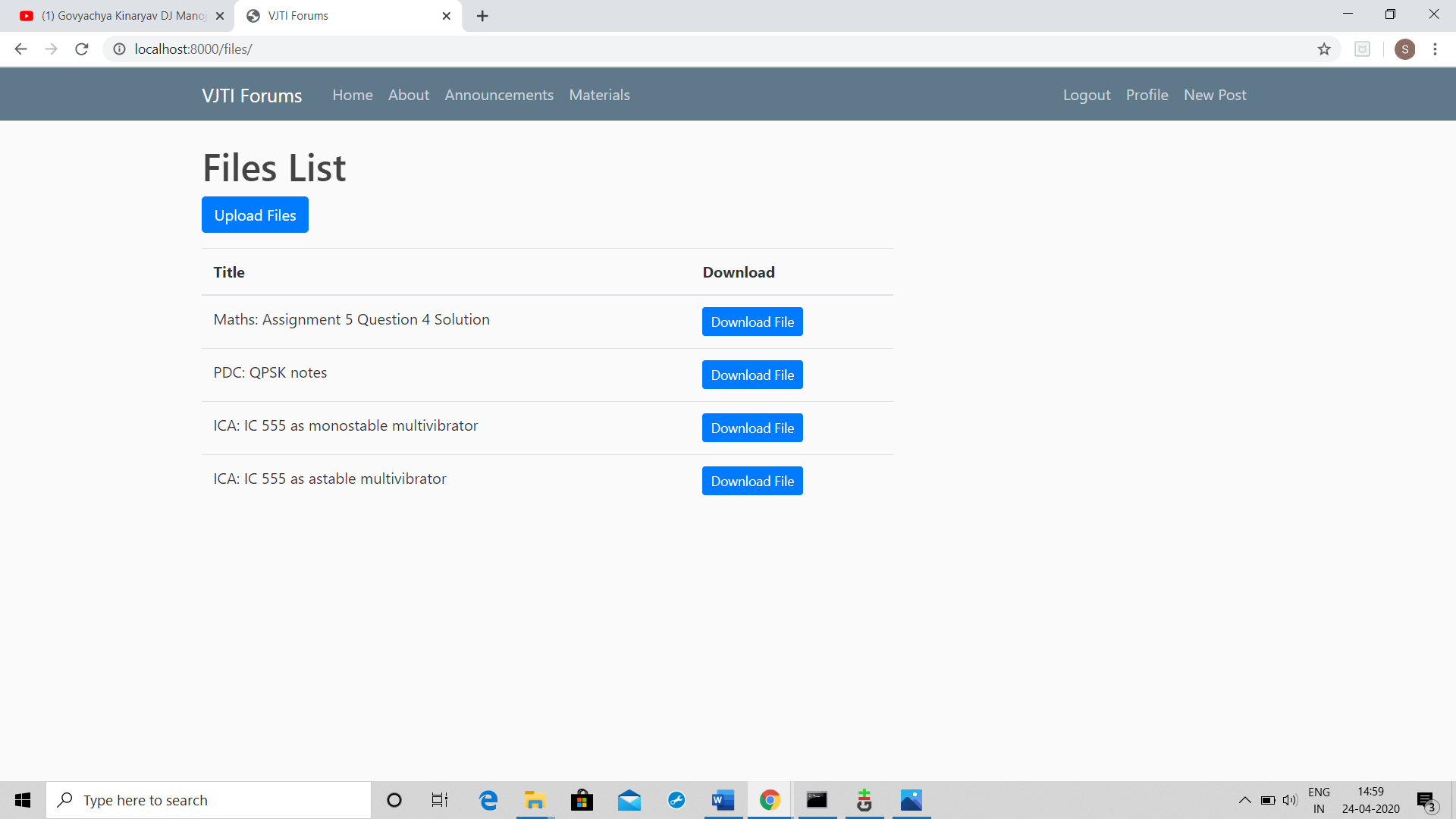
**About page**



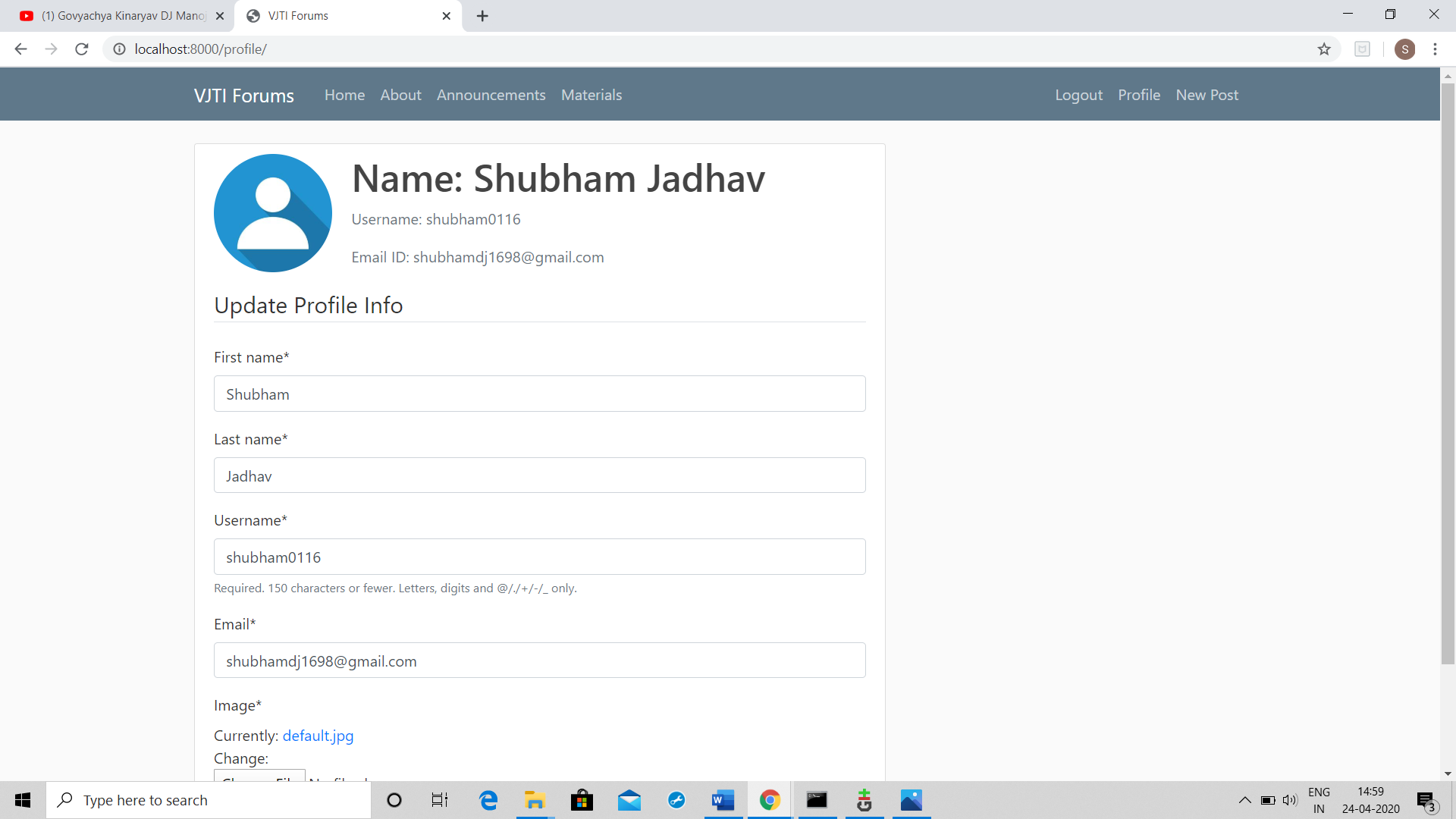
**Announcements**



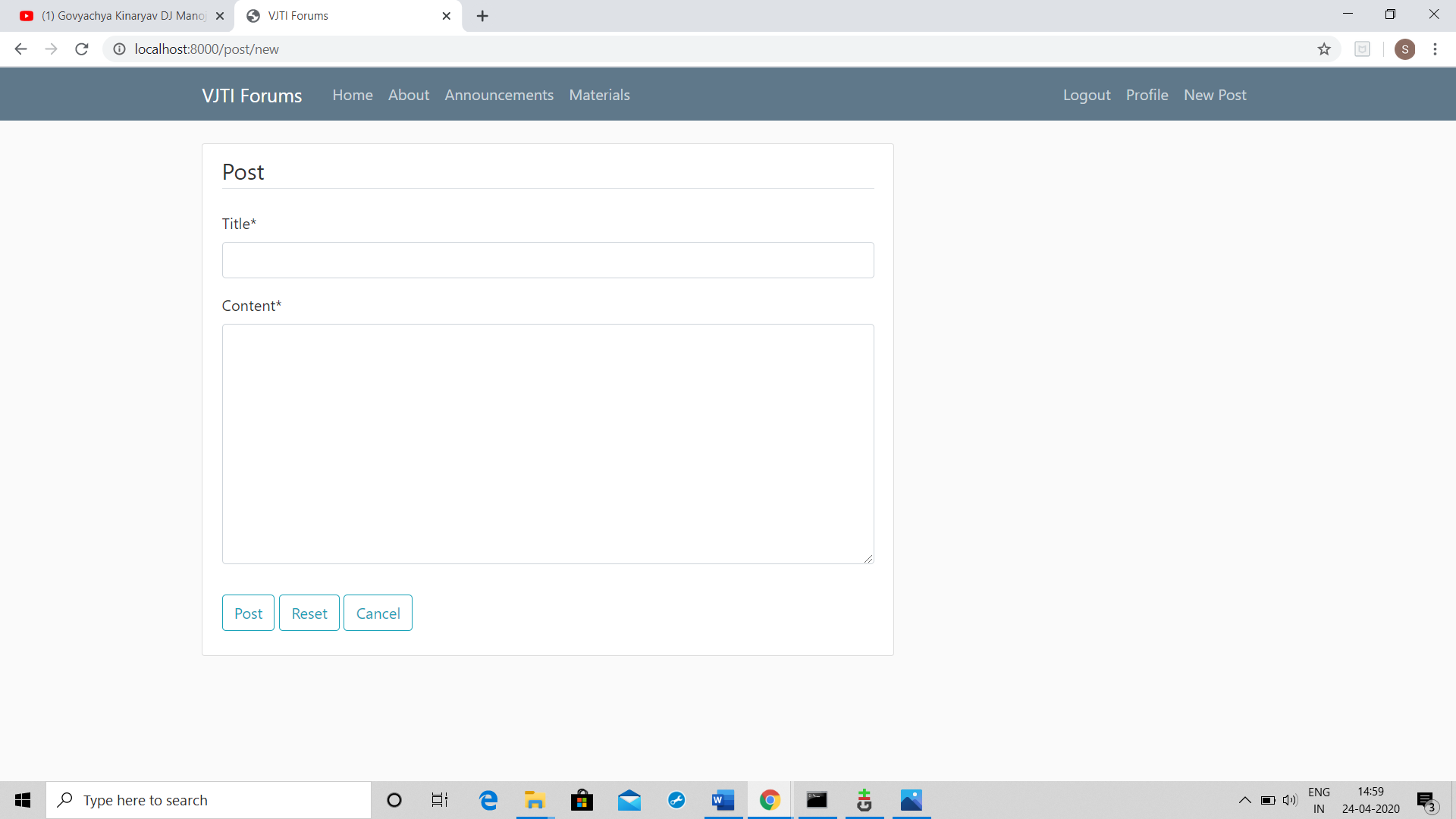
**Materials**



**Profile Page**



**New Post**



**Login Page**

