WEB-NOTES

A PROJECT BASED LEARNING

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

BY

K. S. V RAVI SASTRY	**N.V.S.R SRUJAN**
(20331A0588)	**(20331A05C6)**
K. SARATH KUMAR	**I. DEENA**
(20331A0584)	**(20331A0573)**

Under the Supervision of

ARUNA KUMARI

Assistant Professor

MANSAS

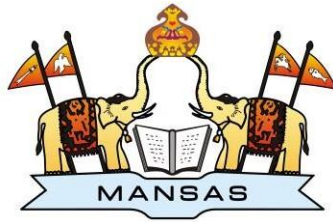**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

MVGR COLLEGE OF ENGINEERING (Autonomous)

VIZIANAGARAM-535005, AP (INDIA)

(Accredited by NBA, NAAC, and Permanently Affiliated to Jawaharlal Nehru Technological University Kakinada)

2022-2023

CERTIFICATE

This is to certify that the project report entitled "**WEB-NOTES**" is being submitted by **K.S.V RAVI SASTRY, N.V.S.R SRUJAN, K.SARATH KUMAR, I.DEENA** bearing registered numbers **20331A0588, 20331A05C6, 20331A0584, 20331A0573** respectively, in partial fulfilment for the award of the degree of "**Bachelor of Technology" in Computer Science and Engineering** is a record of bona fide work done by them under my supervision during the academic year 2022-2023.

Dr P Ravi Kiran Varma   **P.RAMA SANTOSH NAIDU**
Head of the Department   **Assistant Professor**
Dept. of CSE   **Dept. of CSE**

ABSTRACT

Taking a note is the process of capturing information in a source or event. This usually takes type of writing, jotting, wording, scanning, labelling and outline. Many apps have their own unique features, storage, editing, and sharing capabilities, and so on for the user to see which note-taking app will work best for their needs but every time when they work on pcs they have to take their phone and note it down and also website have a better compatibility and require less money than mobile applications. The purpose of this website is to simplify the note taking process on any device.

TABLE OF CONTENTS

WEB-NOTES

1. INTRODUCTION

The Web-Notes is a basic text editor that you can use to create simple documents. The most common use for Web-Notes is to view or edit text(.txt) files online. Every time we work on pcs, we must take our phone and note it down and websites have better compatibility and require less money than mobile applications. The purpose of this website is to simplify the note taking process on any device.

Why Our Application?

- ✓ Simple Web Application, easy to use and very easy to deploy locally
- ✓ Written in simple python. Even a beginner Python developer can contribute to this
- ✓ REST API for retrieving data easily

2. REQUIREMENTS

SOFTWARE /HARDWARE REQUIREMENTS

SOFTWARE:

1. Python

2. PyCharm

3. Flask

4. Other Modules Needed:

      ✓ Flask-RESTful == 0.3.6

      ✓ Jinja2==2.9.6

      ✓ WTForms == 2.1

      ✓ Flask-WTF==0.14.2

HARDWARE:

1. PROCESSOR: Intel Core(I3 processor)

2. RAM:4GB

3. HARD DISK:500GB

3. IMPLEMENTATION

It is a web-based application built with Flask and Python to store notes and add tags to notes.

Flask Framework

✓ Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application.

✓ Flask is a small and powerful web framework for Python. It's easy to learn and simple to use, enabling you to build your web app in a short amount of time.

✓ Flask is cost and time effective and it suits any kind of project irrespective of the size of the project.

✓ Flask is cross-platformed(supports windows, Linux, mac os) and it is compatible with most major databases and highly scalable.

✓ Flask follows DRY(Don't Repeat Yourself) Philosophy i.e. reusability of code is possible and it is highly secure and one doesn't need to worry about the common security concerns.

✓ Flask has its own infrastructure i.e. it has pretty much everything, from a web server and a templating engine to an Object Relational Mapper (ORM), which allows the framework to use different databases and switch between them within one project.

Major Roles in the Application:

1. **Admin**
2. **User**

WEB-NOTES

Functions of the Roles:

    1. Admin:
- ✓ After login you can see the users registered

- ✓ Can View, Update, Delete User.

- ✓ Can see the last login of the User.

    2. User:
- ✓ The User can create an account by providing basic details.

- ✓ Account can be created by No Approval Required by Admin and Can log in After Signup

- ✓ The User can create a note and add a tag to the note.

- ✓ The User can also add new tags for grouping.

- ✓ The User can edit the already taken note

- ✓ The User can also write Markdown files

- ✓ At the Time of writing the User can see the live preview of their note

- ✓ The User can change their password and can also see their last login time.

WEB-NOTES

Tech Stack:

A. Frontend: HTML, CSS and JavaScript

B. Backend:

    1. Web Framework: Django and Django Rest

    2. Programming Language: Python

    3. Database: SQLite

    4. Deployment Server: Replit.com

C. Testing: Flask (Unit Test Module)

4. DATABASE FILE

```
use notes;
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_Z
ERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';


-- ---------------------------------------------------
-- Schema notes
-- ---------------------------------------------------

-- ---------------------------------------------------
-- Schema notes
-- ---------------------------------------------------
CREATE SCHEMA IF NOT EXISTS `notes` DEFAULT CHARACTER SET utf8 ;
-- ---------------------------------------------------
-- Schema notes
-- ---------------------------------------------------
USE `notes` ;

-- ---------------------------------------------------
-- Table `notes`.`users`
-- ---------------------------------------------------
CREATE TABLE IF NOT EXISTS `notes`.`users` (
 `id` INT NOT NULL,
 `registered_at` TIMESTAMP NULL,
 `last_login` TIMESTAMP NULL,
 `username` VARCHAR(255) NULL,
 `password` VARCHAR(200) NULL,
 `email` VARCHAR(200) NULL,
 PRIMARY KEY (`id`))
ENGINE = InnoDB;



-- ---------------------------------------------------
-- Table `notes`.`notes`
-- ---------------------------------------------------
CREATE TABLE IF NOT EXISTS `notes`.`notes` (
 `id` INT NOT NULL,
 `created` VARCHAR(45) NULL,
 `updated` TIMESTAMP NULL,
 `note_title` VARCHAR(45) NULL,
 `note` TEXT(1000) NULL,
 `note_markdown` TEXT(1000) NULL,
 `tags` VARCHAR(200) NULL,
 `users_id` INT NOT NULL,
 PRIMARY KEY (`id`, `users_id`),
```

```
 INDEX `fk_notes_users1_idx` (`users_id` ASC) VISIBLE,
 CONSTRAINT `fk_notes_users1`
 FOREIGN KEY (`users_id`)
 REFERENCES `notes`.`users` (`id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION) CREATE TABLE `notes` (
  `id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  `created` TIMESTAMP NOT NULL DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now',
'localtime')),
  `updated` TIMESTAMP NOT NULL DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now',
'localtime')),
  `note_title` VARCHAR(255),
  `note` TEXT,
  `note_markdown` TEXT,
  `tags` VARCHAR(200),
  `user_id` INTEGER,
  FOREIGN KEY(user_id) REFERENCES users(id)
);

CREATE TRIGGER `triggerDate` AFTER UPDATE ON `notes`
BEGIN
  update `notes` SET `updated` = (strftime('%Y-%m-%d %H:%M:%S', 'now', 'localtime')) WHERE id =
NEW.id;
END;

CREATE TABLE `users` (
  `id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  `registered_at` TIMESTAMP NOT NULL DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now',
'localtime')),
  `last_login` TIMESTAMP NOT NULL DEFAULT (strftime('%Y-%m-%d %H:%M:%S', 'now',
'localtime')),
  `username` VARCHAR(255),
  `password` VARCHAR(200),
  `email` VARCHAR(200)
);

CREATE TRIGGER `triggerUserLogin` AFTER UPDATE ON `users`
BEGIN
  update `users` SET `last_login` = (strftime('%Y-%m-%d %H:%M:%S', 'now', 'localtime')) WHERE id
= NEW.id;
END;

CREATE TABLE `tags` (
  `id` INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
  `tag` TEXT,
  `user_id` INTEGER,
  FOREIGN KEY(user_id) REFERENCES users(id)
);
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `notes`.`tags`
-- -----------------------------------------------------
```

```
CREATE TABLE IF NOT EXISTS `notes`.`tags` (
 `id` INT NOT NULL,
 `tagscol` TEXT(100) NULL,
 `users_id` INT NOT NULL,
 PRIMARY KEY (`id`, `users_id`),
 INDEX `fk_tags_users_idx` (`users_id` ASC) VISIBLE,
 CONSTRAINT `fk_tags_users`
 FOREIGN KEY (`users_id`)
 REFERENCES `notes`.`users` (`id`)
 ON DELETE NO ACTION
 ON UPDATE NO ACTION)
ENGINE = InnoDB;


SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

5. OUTPUTS
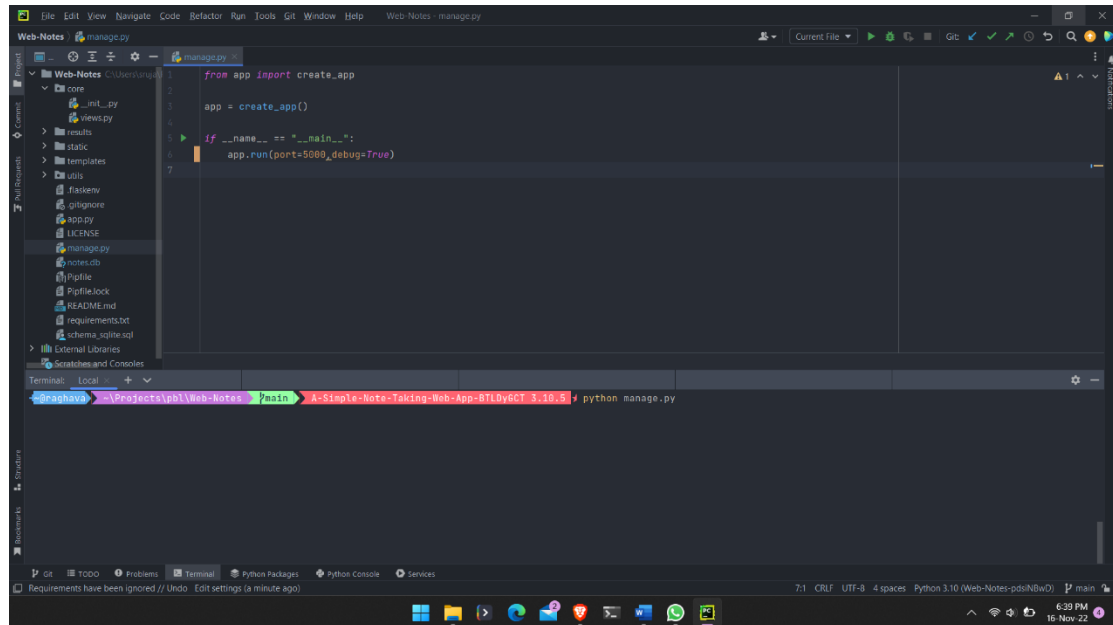
python manage.py – run this command in terminal to direct to the web page view.
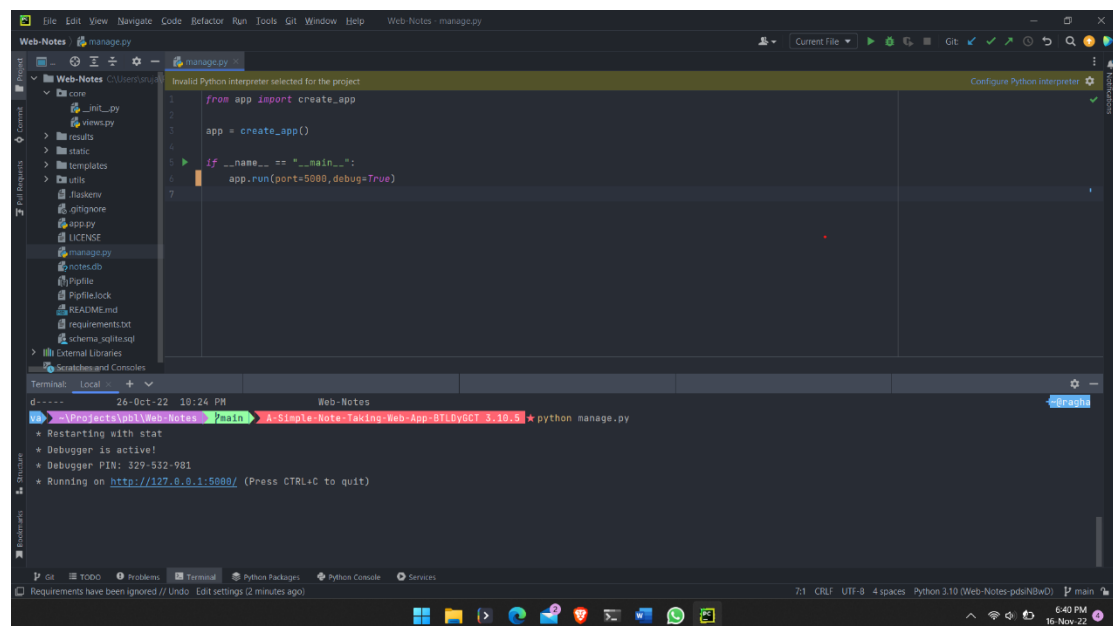
When you click on **http://127.0.0.1:8000/** shown in the above figure , the web browser will be opened.

WEB-NOTES

HOME PAGE

This page shows the basic interface of our website and the display of the content what the user wants to access.
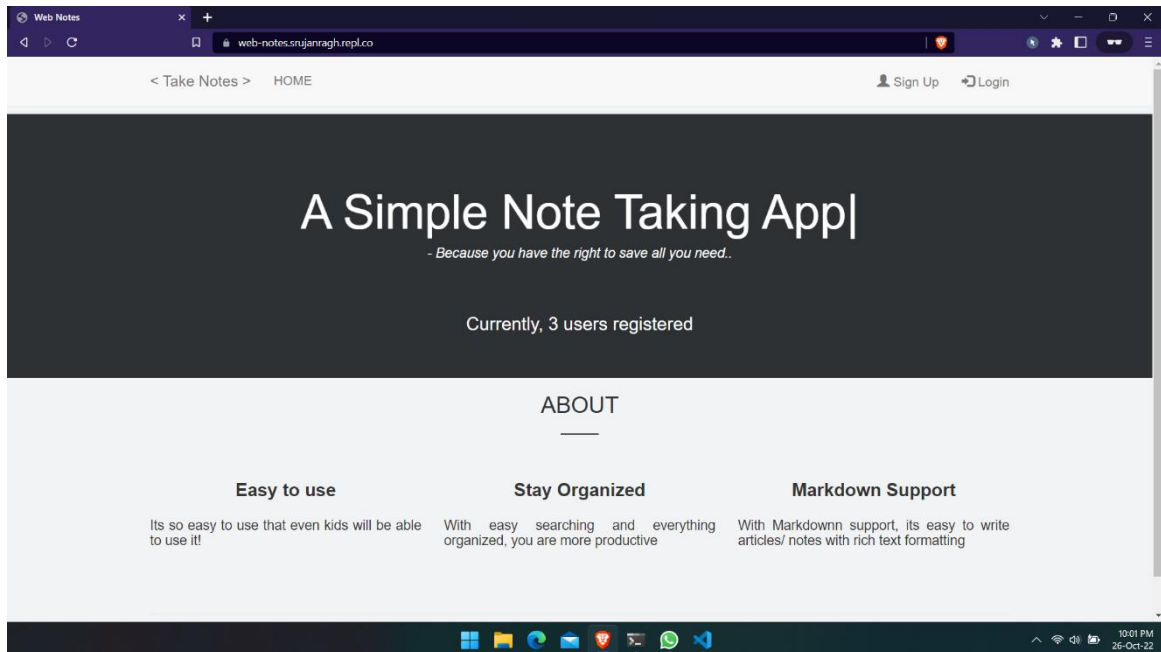
USER HOME PAGE

On this page, the user can search for the required blood group and select the place where the recipient needs the blood to be delivered /donated.
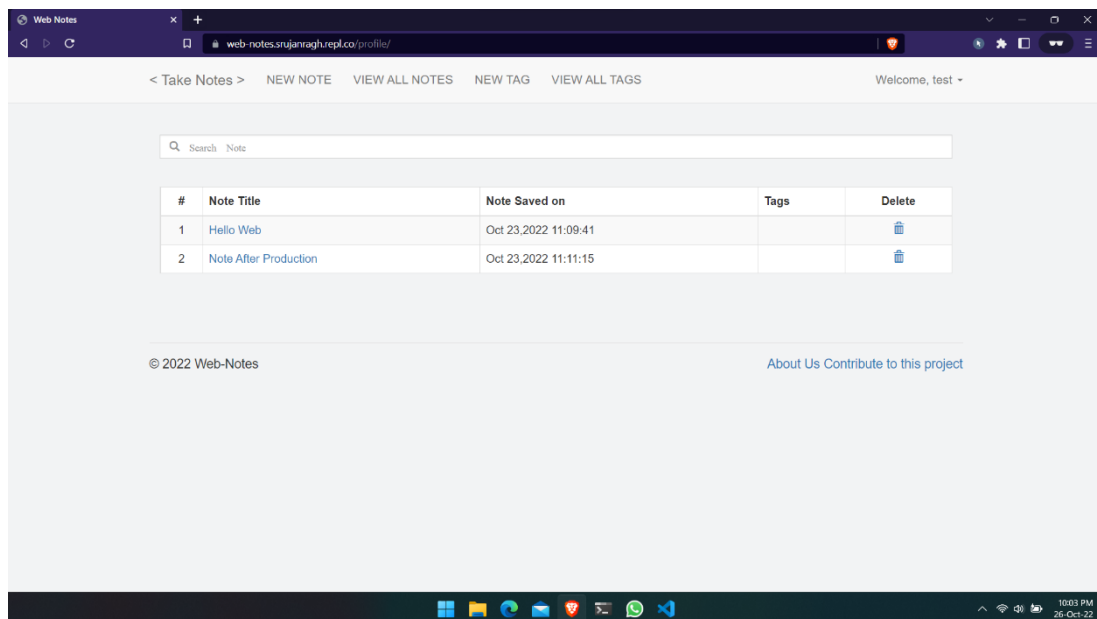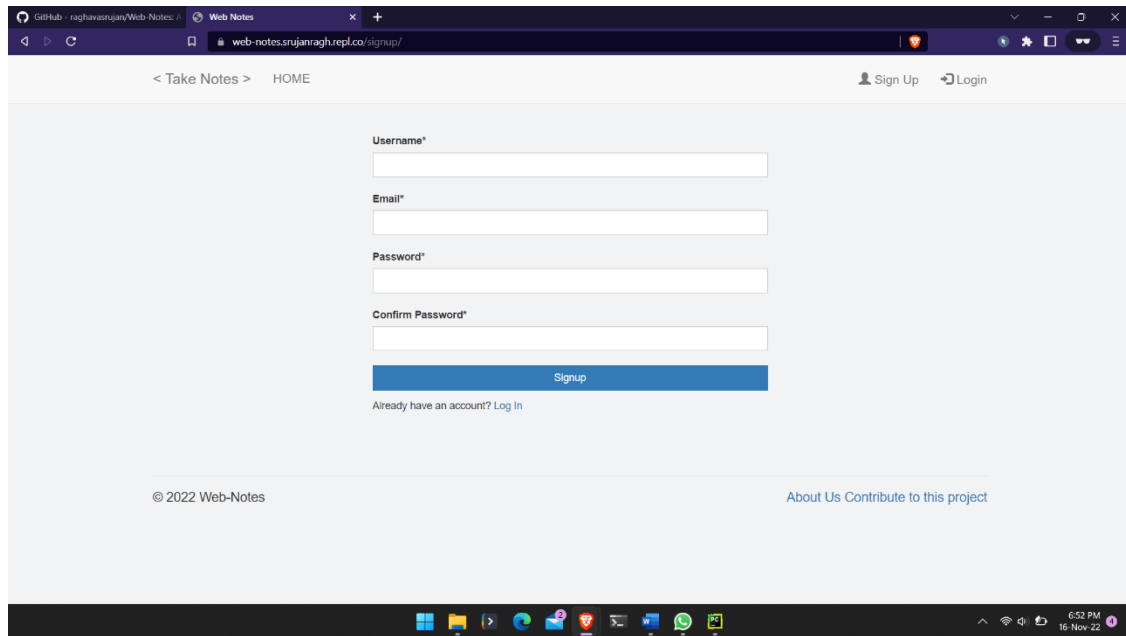
WEB-NOTES

USER REGISTRATION PAGE

On this page, a user can register to this website. The user needs to give details like
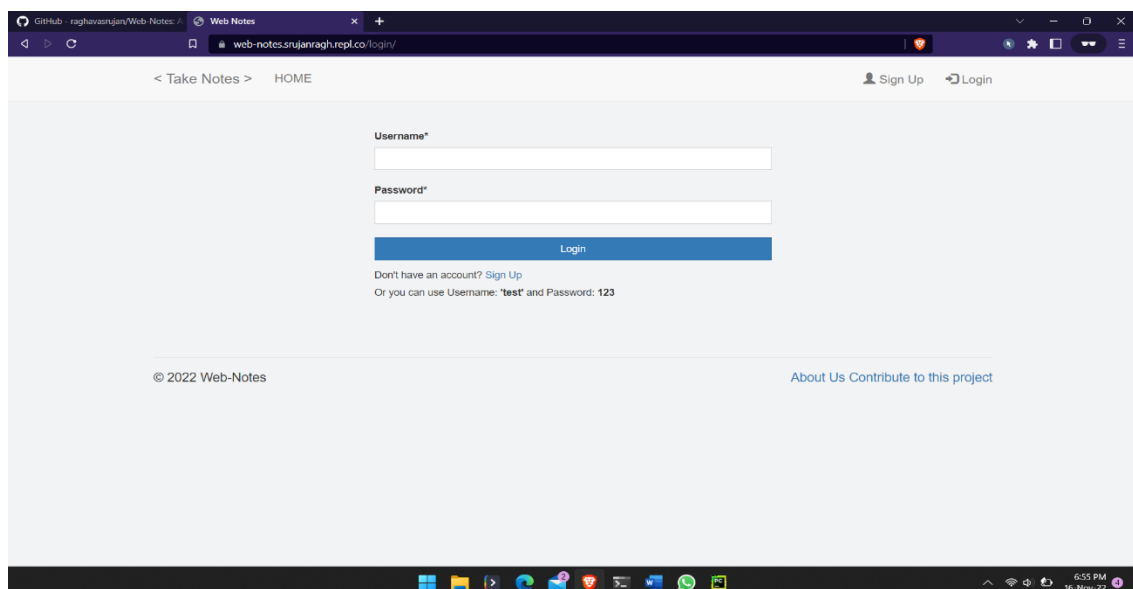
1. Username
2. Email
3. Password
4. Confirm Password

LOGIN PAGE

This page can be accessed only by the admin and the admin is the one who accesses all the details of the registered users. This page shows how many members accessed and how many members registered the website.

WEB-NOTES

NEW NOTE PAGE

This page consists of adding your note and adding tag

NEW TAG PAGE

This page can add new tags to the user account.

WEB-NOTES

DETAILS ABOUT A USER

On this page, we can see the details about a logged user.
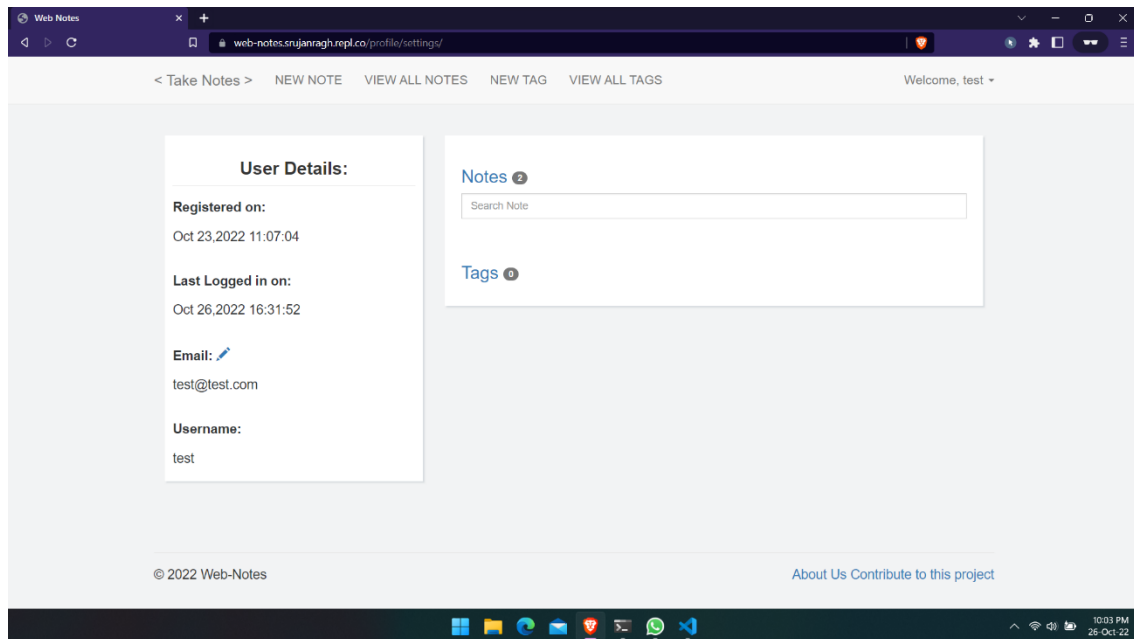
6. CONCLUSION

We successfully implemented our project which helps us to create and edit notes in online web-based application. There are many applications like notepad, notepad++ in windows and mac but we cannot access them wherever we need and There are no tag-like options available in the above specified applications.Our project helps the users to access the notes anywhere from the website which is authentic and secure.

7. PROJECT OUTCOME

- ✓ From this project, we have learnt how to improve our presentation skills, how to make an effective presentation and even learnt how to present our thoughts and implementation through this.

- ✓ From this project, team management skills improvised a lot, each one of us enthusiastically took each part of theirs and worked on it. We learnt how coordination can help make a bigger problem into a smaller problem and each person was the leader of the team in their way.

- ✓ From this project, communication skills have got better and how to represent ourselves in front of a crowd improved, communicating among ourselves also helped in making our work easier.

- ✓ From this project, we have learnt a new software called Flask framework which is used to implement web applications using a python programming language.

- ✓ Finally, this project helped us in learning new skills, helped us to think out of the box, improve our thoughts towards society and how to take up a problem from our daily life and put some work on it to give a minimal solution which would positively benefit the society.

WEB-NOTES

REFERENCES

1. https://keep.google.com/
2. https://simplenote.com/blog/