Project Report

CST2120 – Coursework 3

Name: Rodrigo Ferreira

Student Number: M00736217

**1. Introduction**

The goal of this project is to serve public indexed information in a simple and honest manner.

**2.1 Design**

**Design purpose**: Simple page to serve online academic publications in pdf format.

**Design:**

app.js - main, connect to database and serve papers to the page

public - files :: pdf papers; stylesheets :: css sheet

views - ejs files (index, partials/header, partials/footer)

**Db:**

admin - papers\_setup :: compress/decompress and setup papers to the correct folders; crud\_db :: insert, delete, update and select from database menu (bash); admin\_db :: admin database management controls

database - bash scripts manage and configure database

**Dev:**

admin - notes :: Linux commands, GNU utilities and database commands; tests :: frontend and backend tests, backend use run-test script to test or reset database

**2.2. Implementation**

**Back-end features**

Database automation.

Test and run database functionality with CLI bash scripts running shell and GNU utilities.

Load pdf plus csv file with information as described in contributing/template with a simple bash script into the database with a few keystrokes.

Run-test tests database functionality with a bash script.

Compress and decompress pdf for portability using 7z.

**Front-end interaction**

Single page using EJS

Table of contents - papers in pdf format open in browser or download

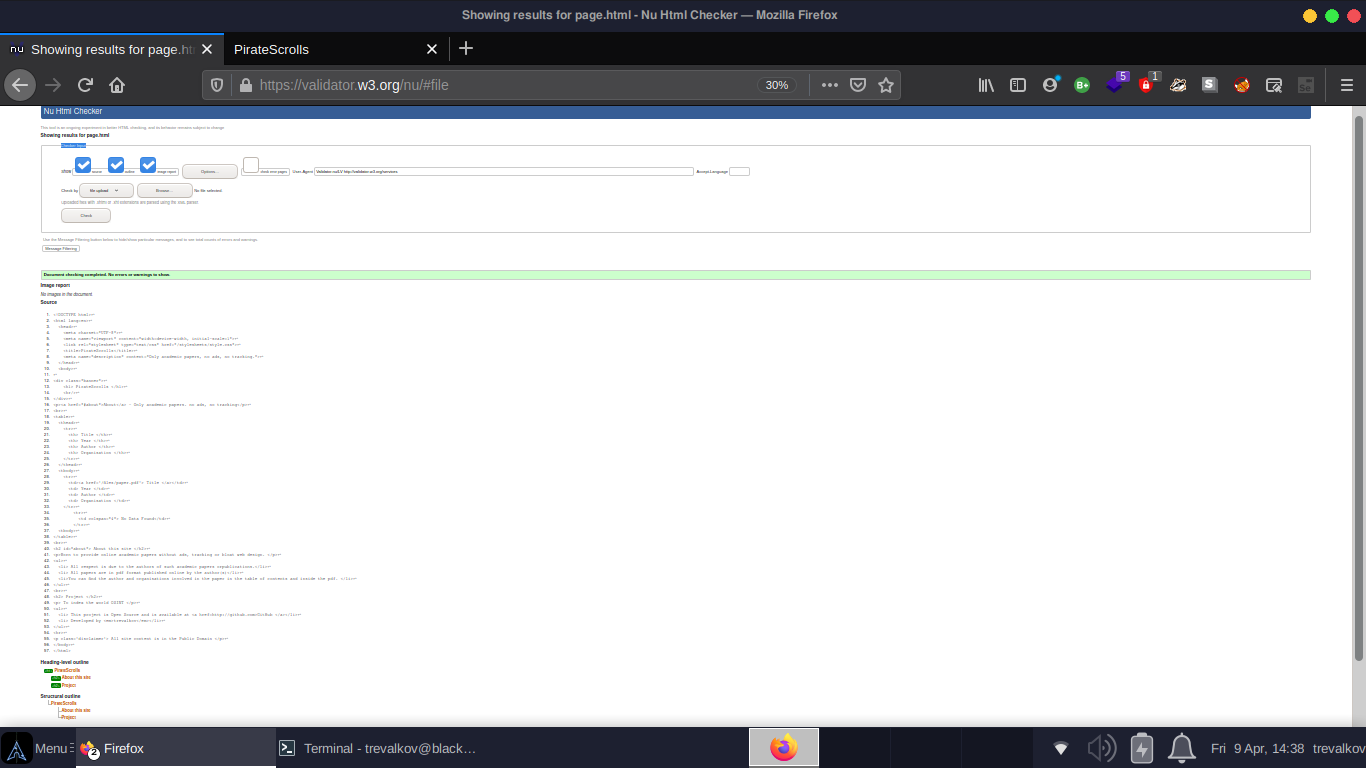
About - about project page

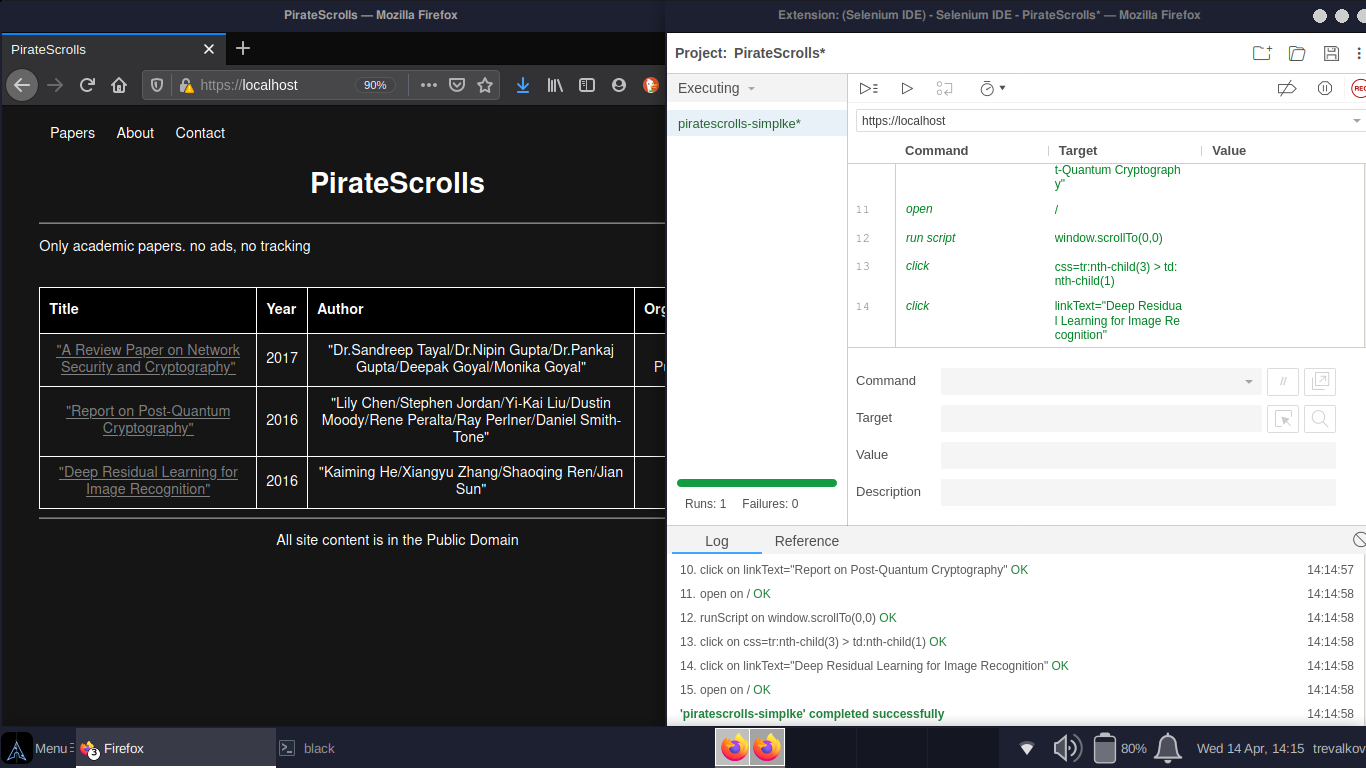
GitHub - link to the project

**3. Testing**

**Application**

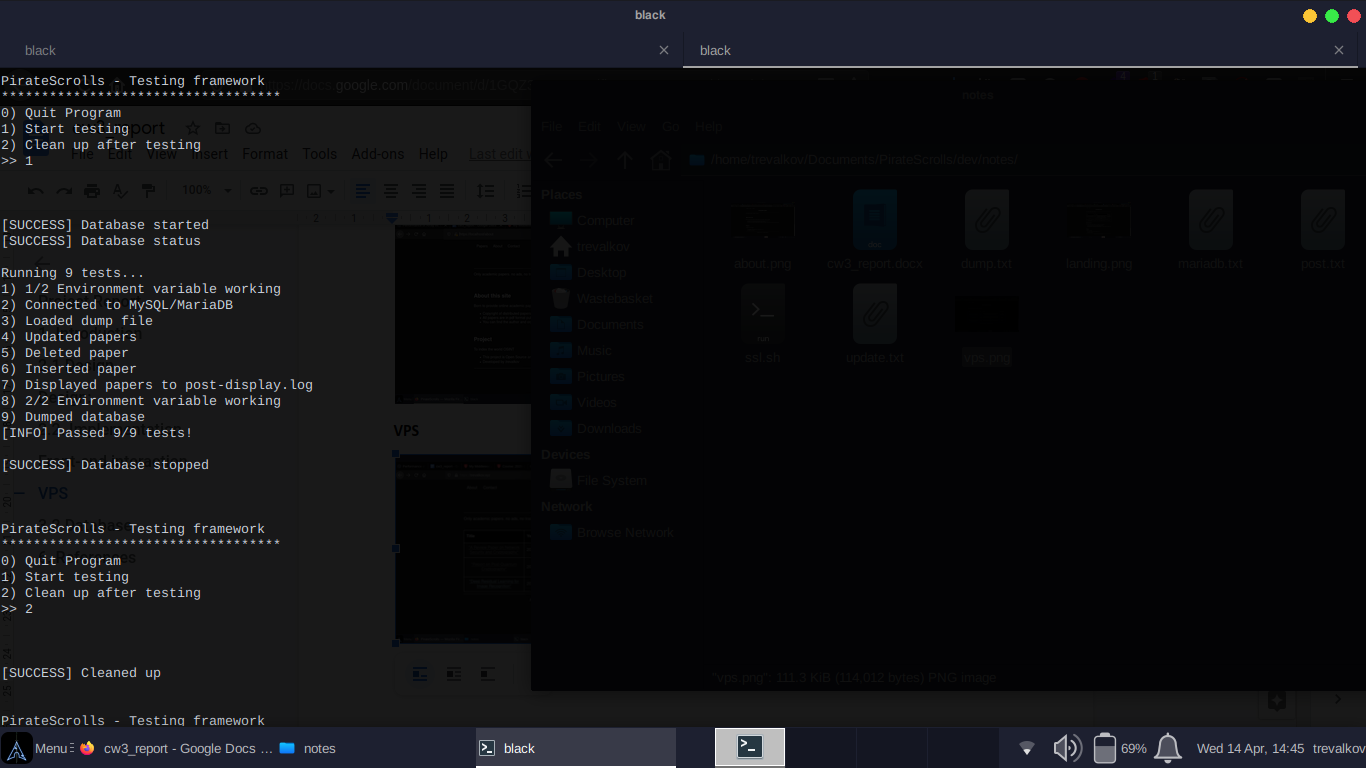
Testing expected DOM interaction successful

****

****

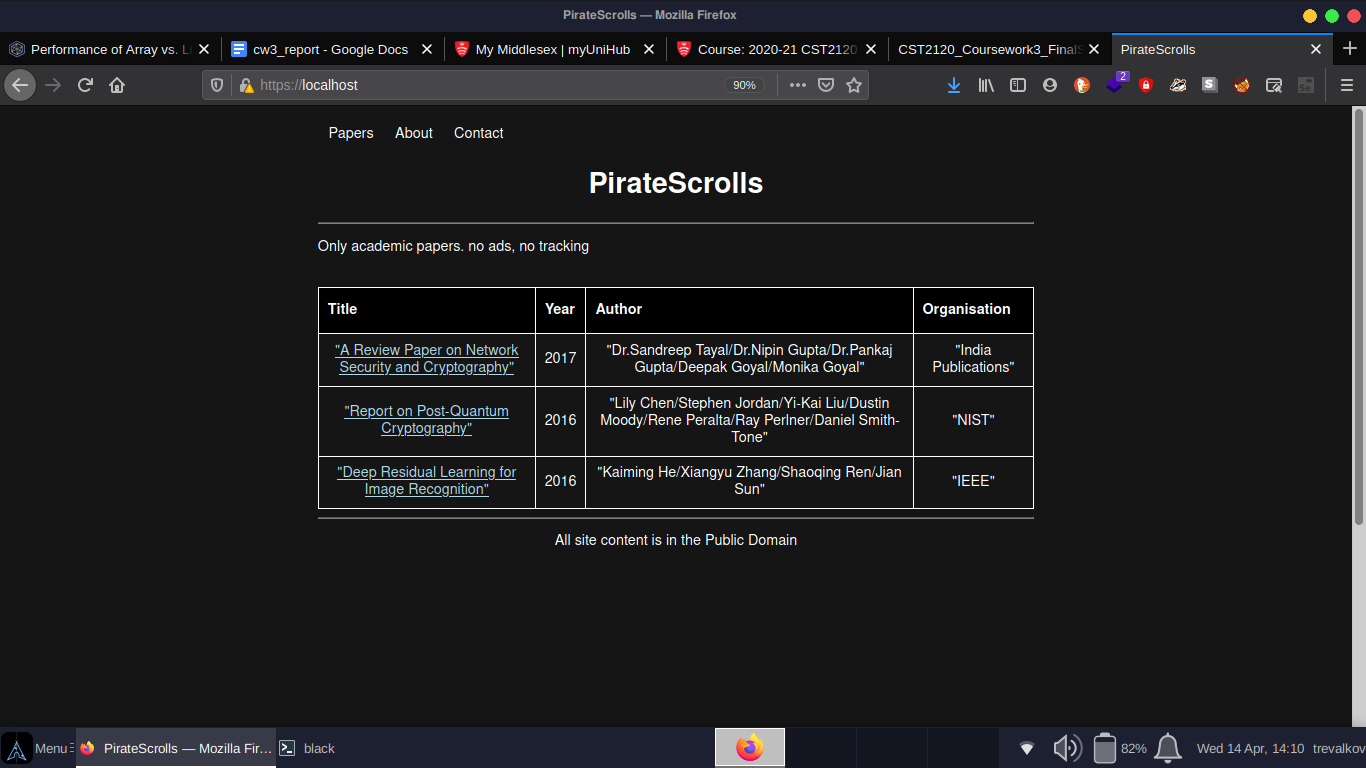
**Database**

Testing database interaction and management, if this script passes all test the database is configured correctly

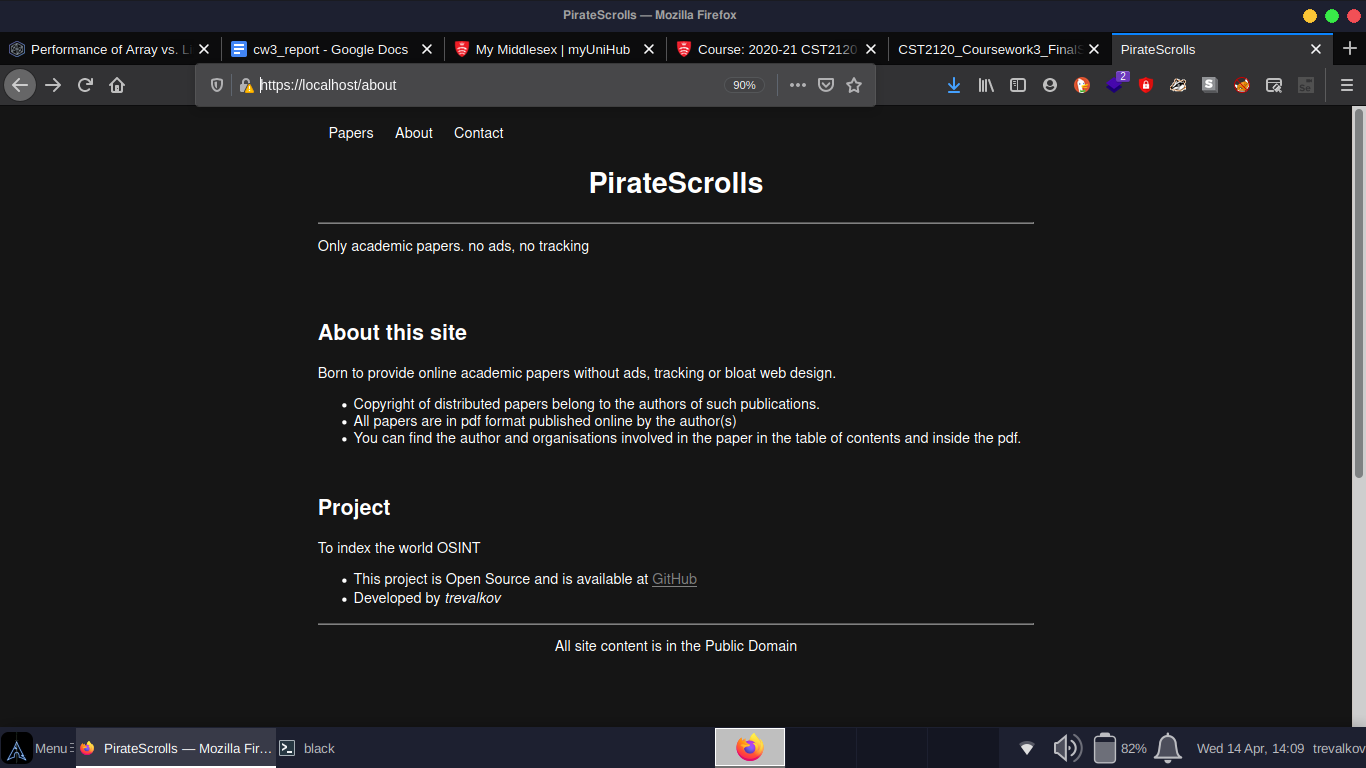
****

**4.1 Application**

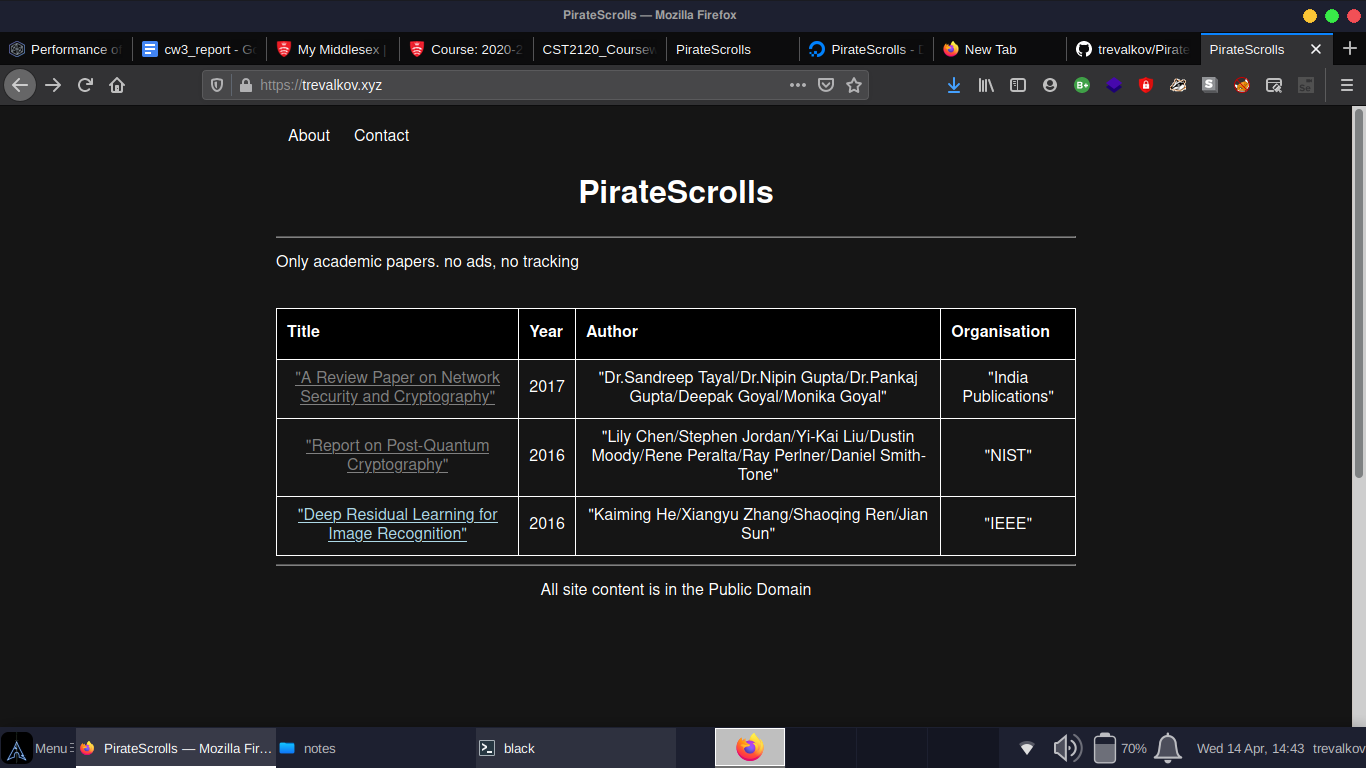
**Landing**

****

**About**

****

**VPS**

****

**4.2 Database**

-- Tables

:: papers(5)

:: :: not null int auto\_increment primary\_key $id

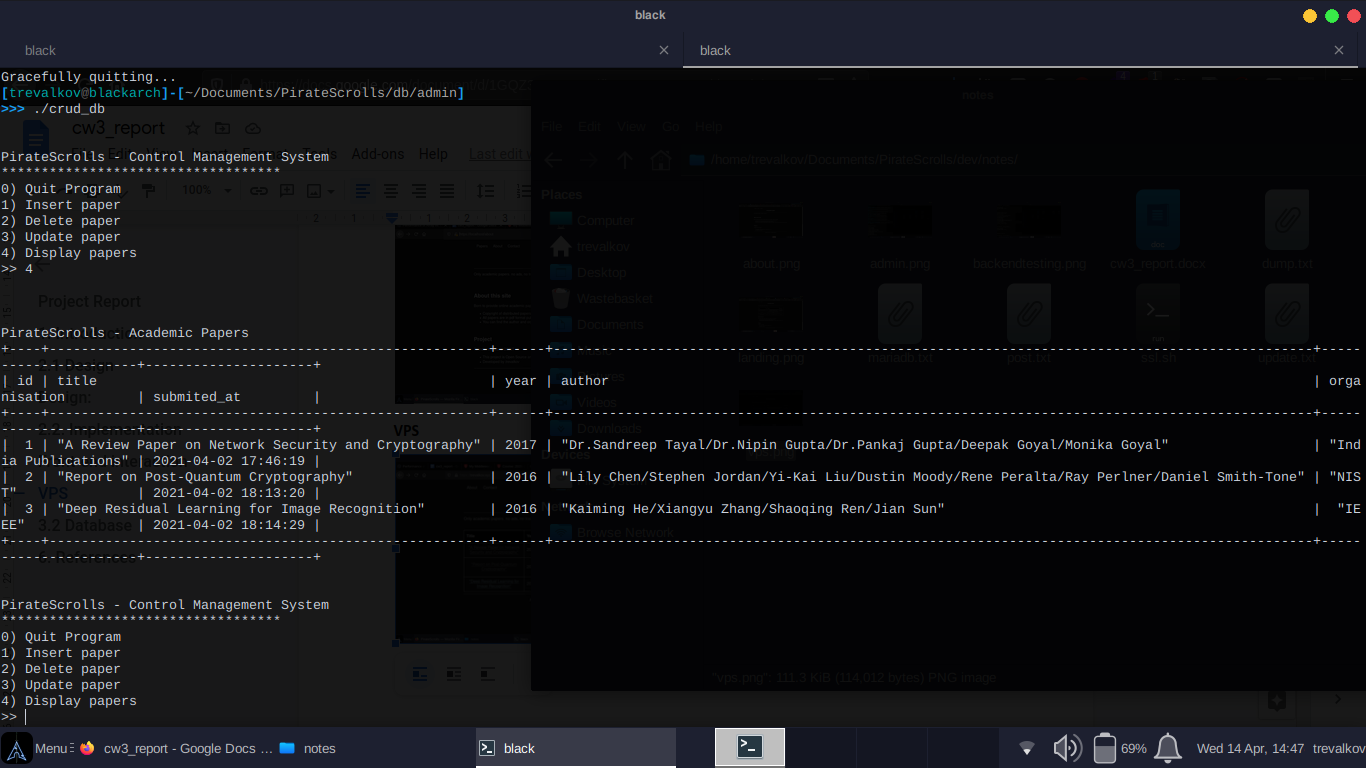
:: :: not null varchar(256) $title

:: :: not null int

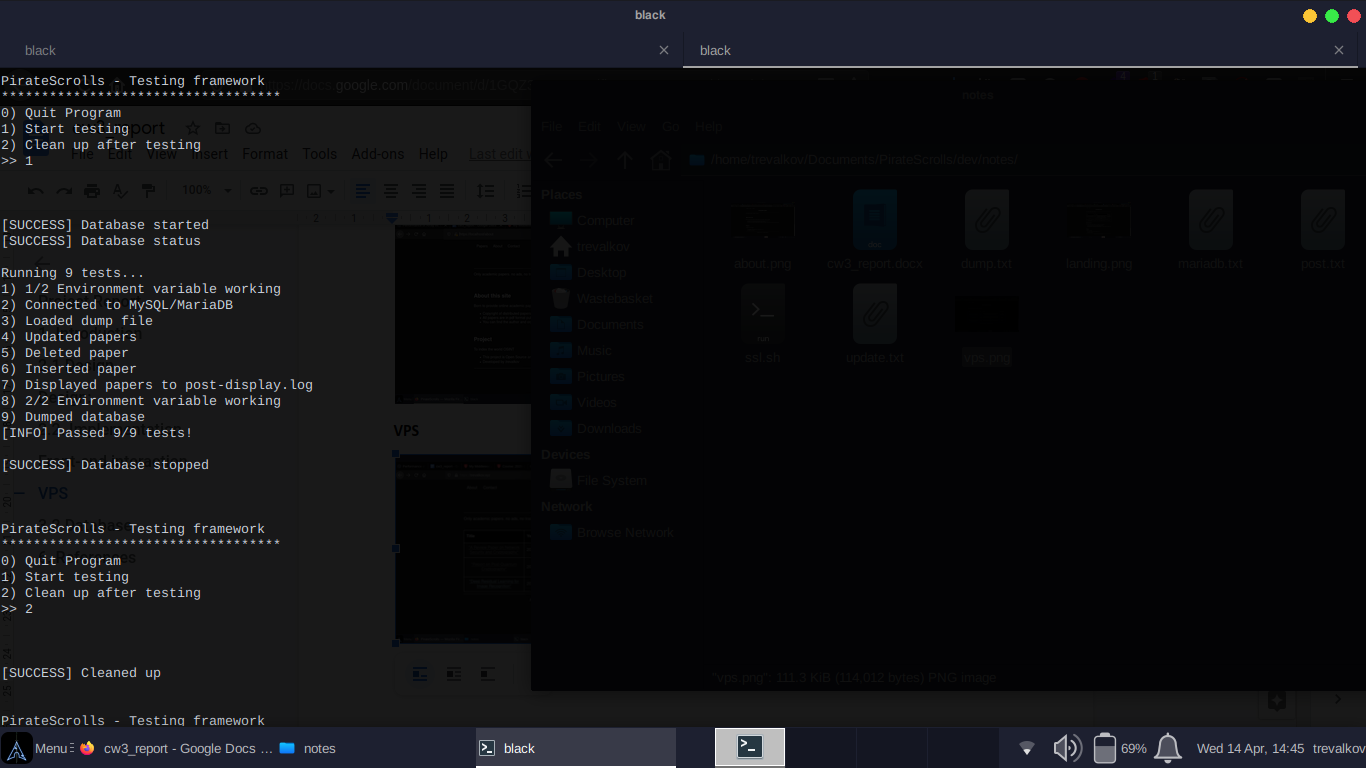
:: :: not null varchar(256) $author

:: :: not null varchar(128) $organisation

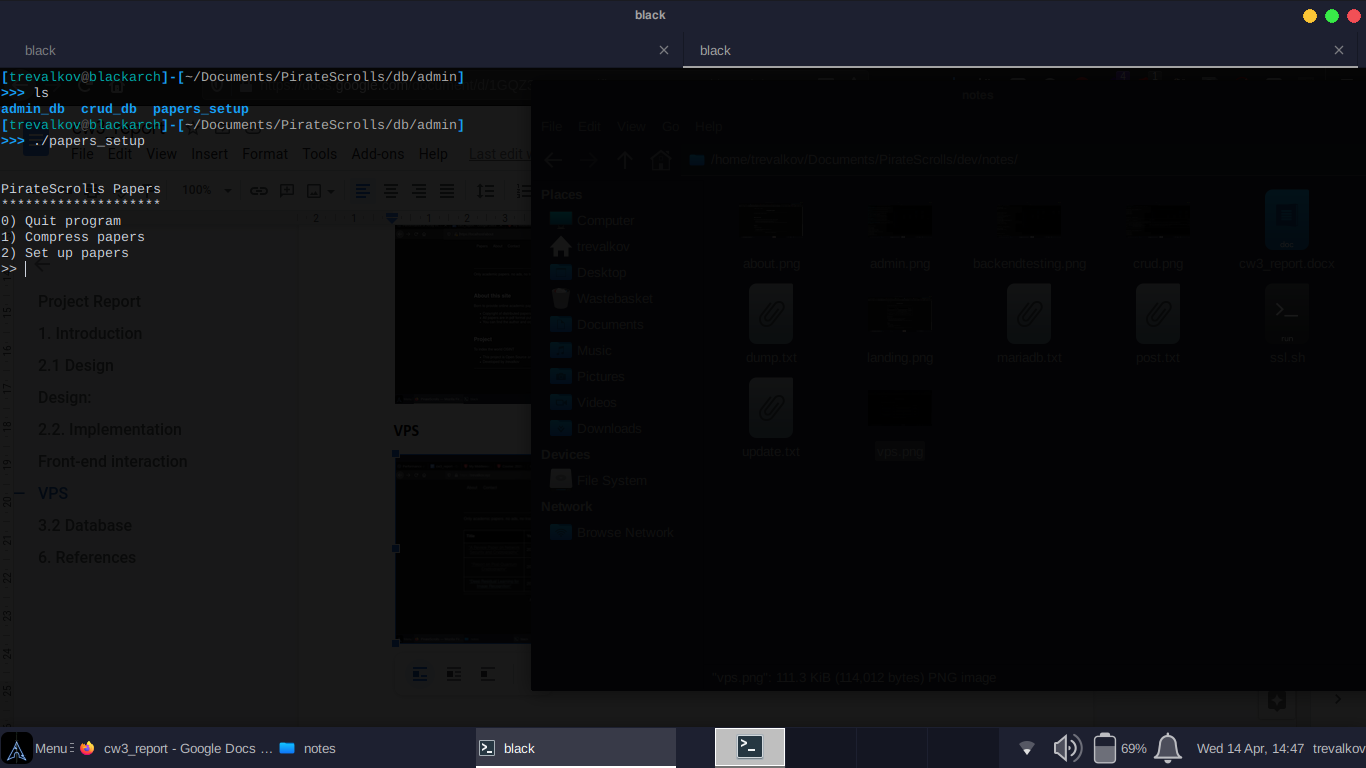
**Admin Control**



**Crud CMS**



**Setup papers**



**6. Project Description**

* For this project I decided to build a web server for educational content. This website was designed with a minimalistic approach thus avoiding web bloating rendering the using EJS to respect the REST framework
* To serve its purpose it doesn’t require tracking or ads either, making the application itself not too hard to develop, I have tried to think about some extra features I could add as an Express middleware but it didn’t seem necessary, after noticing that I decided to invest some more time into the backend and database.
* Simply rendering the page with the papers and database data selected from the main table on both http and https on ports 80 and 443 with a self signed certificate (I have tested it on a VPS with a Let’sEncrypt certificate as shown in the last screenshot)
* Under db/database there is a CRUD app to control the database and simply add new papers with their respective data (title, year, author, organisation) with just a simple keystrokes after moving the pdf paper to the ‘contributing’ folder and creating the CSV with the data inside the same folder. An admin control script to easily manage the database in most Linux distros and automated database testing with another script as well. db/admin holds the admin and crud script as well as a script to compress and decompress and setup the papers to serve on the website as pdf files can become heavy. There is a dump backup available at db/database/dump/PirateScrolls1.sql with just the template values I used in the screenshot.
* To quickly start the server start MySQL, create database PirateScrolls, load dump into PirateScrolls, go to dev/notes and use the ssl.sh to get self signed certificates, then go to dev/tests/backend and run the run-test script to test the database. Then go to app and with root permissions run ‘node app.js’ to be able to start the server on ports 80 and 443.

**7. References**

- <https://github.com/>

- online lectures/labs

- Q&A sessions

- <https://www.youtube.com/>

-<https://odysee.com/>

- “Privacy Considerations of Online Behavioural Tracking” [ENISA, 2012]