COSC 349

Assignment 2

Leon Hoogenraad - 253 8994

1. Deployment

1.1 Webservers

I deployed the front end of my application as two VM's, display and upload. These are automatically created and configured as AWS EC2 instances upon running "vagrant up" in the home directory of the git repository (where the Vagranfile exists).

1.11 Href links

The only thing that is not automated about building the application is the href links at the bottom of the webpages, as well as the redirect link after submitting an answer. I couldn't find any way to set these automatically, so after running vagrant you need to run vagrant ssh vm_name for each VM, and modify the href links in /vagrant/www/display/index.php, /vagrant/www/upload/upload.php, as well as the redirect link at the bottom of /vagrant/www/upload/uploadToDB.php. You only need to modify the display's index file on the display VM, likewise you only need to modify the upload files on the upload VM, although there is no stopping you from modifying all the files on both machines. Nevertheless, the respective links you need to change it to can be found in the AWS EC2 instances console.

1.12 Region & subnet

Our EC2 instances come under the region us-east-1. The availability zone I decided to use comes under us-east-1f. I had to use an alternative method of finding the subnet id of this availability zone, as my awscli wasn't working. I had to look up the documentation for boto3, write out code in a python script to print out all the information of the us-east-1 availability zones, and from there I was able to get the subnet id for us-east-1f.

1.13 Security groups

When the webserver VM's boot, they are automatically assigned to two security groups. One allows for ssh traffic from all addresses (in actual production this would ideally be changed to more specific addresses for security reasons), and the other allows for http traffic from all addresses (This probably should remain the same as we want everyone to be able to access the web application).

1.14 AMI

The ami is the disk image the EC2 instance should use. The one I decided on is a pretty standard ami, to look it up navigate to https://cloud-images.ubuntu.com/locator/ec2/ and enter “ami-0f40c8f97004632f9” in the search bar.

1.2 Database

I chose to use the provided database cloud software on AWS (RDS, relational database software) rather than try deploy a VM that could act as a database server. This involved creating a new security group that allowed for connecting to the database, as well as configuring settings about the database (database name, storage size, etc.).

1.3 Connecting web servers and database

Connecting the webservers to the AWS database was surprisingly easy. All I had to do was modify the php code in the webservers’ webpages where the credentials and address of the database was assigned. Instead of attempting to connect to a database on a local network, the code now connects to the AWS database at the address examanswersdb.craafeyaeyxp.us-east-1.rds.amazonaws.com.

2. Using the application

2.1 Accessing the application

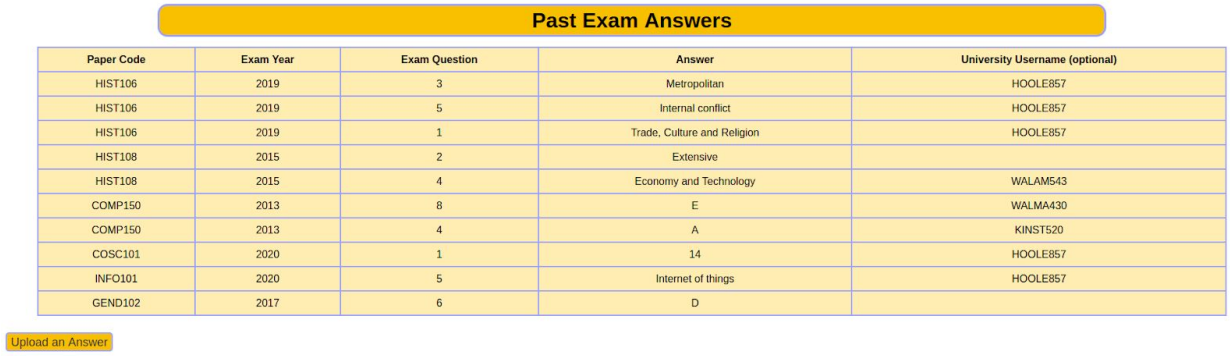
The user can access the application at the address http://ec2-3-235-141-108.compute-1.amazonaws.com/. It should be noted this link may not work once this assignment has been marked.

2.2 Interacting with the application

The function of the application is the same as from assignment one. The application is meant to allow students to upload their answers to past exam questions. They can also view all of the uploaded exam questions on the home page. Upon reaching either page of the application, the user can navigate in between pages using the navigation button at the bottom left of the page.

2.3 Screenshots

2.31 Index before uploading an answer

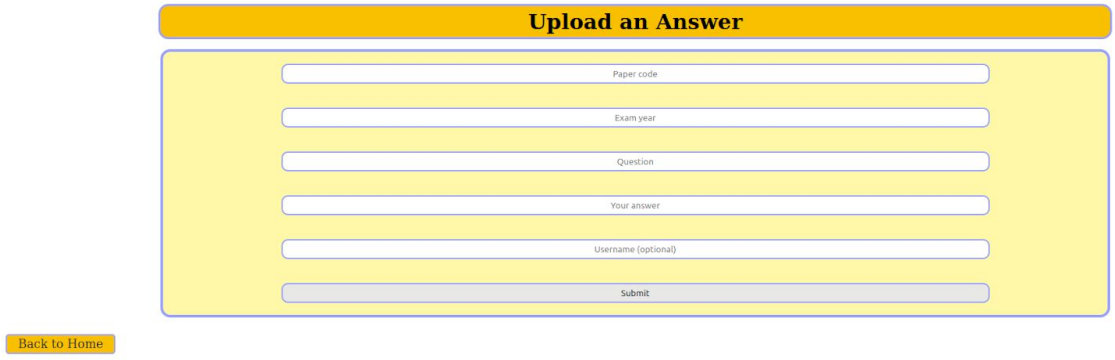


Past Exam Answers

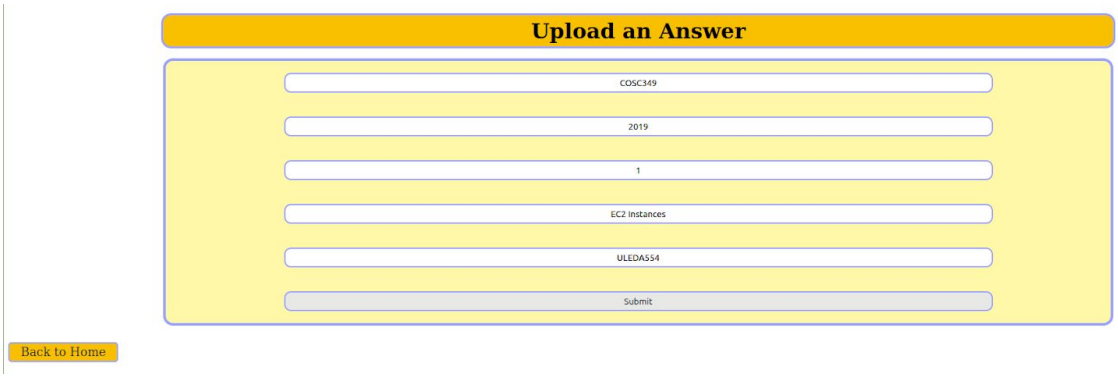
Paper Code	Exam Year	Exam Question	Answer	University Username (optional)
HIST106	2019	3	Metropolitan	HOOLE857
HIST106	2019	5	Internal conflict	HOOLE857
HIST106	2019	1	Trade, Culture and Religion	HOOLE857
HIST108	2015	2	Extensive	
HIST108	2015	4	Economy and Technology	WALAM543
COMP150	2013	8	E	WALMA430
COMP150	2013	4	A	KINST520
COSC101	2020	1	14	HOOLE857
INFO101	2020	5	Internet of things	HOOLE857
GEND102	2017	6	D	

Upload an Answer

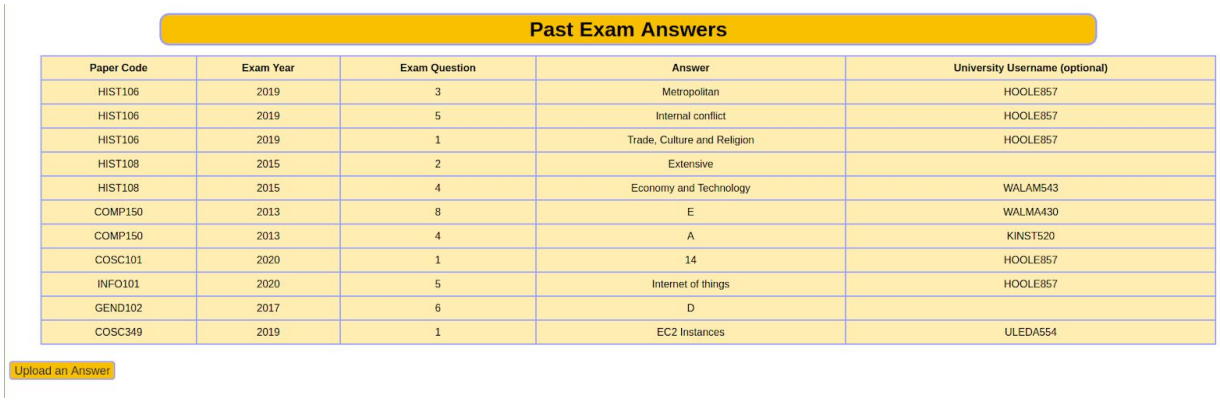
2.32 Upload page with no input



2.33 Upload page with input



2.34 Index page after submitting an answer (new answer at bottom)



Past Exam Answers

Paper Code	Exam Year	Exam Question	Answer	University Username (optional)
HIST106	2019	3	Metropolitan	HOOLE857
HIST106	2019	5	Internal conflict	HOOLE857
HIST106	2019	1	Trade, Culture and Religion	HOOLE857
HIST108	2015	2	Extensive	
HIST108	2015	4	Economy and Technology	WALAM543
COMP150	2013	8	E	WALMA430
COMP150	2013	4	A	KINST520
COSC101	2020	1	14	HOOLE857
INFO101	2020	5	Internet of things	HOOLE857
GEND102	2017	6	D	
COSC349	2019	1	EC2 Instances	ULEDA554

Upload an Answer

3. Cloud service

3.1 AWS RDS Database

I decided to use the provided AWS relational database service, as it was very easy to configure, and to connect the application to. I did have to download a mysql client to connect to the AWS database to set up the schema and insert dummy data, but after that it was all done.

4. Architecture

4.1 Web servers

There are 2 VM's that act as webservers, display and upload.

4.11 Display

The display VM is responsible for pulling data from the AWS database and displaying it neatly on a webpage. It also has a link at the bottom left to the page where the user can upload their own exam answer.

4.12 Upload

The upload VM is responsible for serving a webpage where a user can fill out a form to upload their answer to an exam question. When they submit the form, a php script called uploadToDB.php which sends the user's data to the AWS database, then redirects the user to the index page.