

# COSC349 Assignment 1

---

Report for Shay Stevens, ID #7196262

## Introduction

My application is a dog booking website, called Doge Rentals. The application is built using three virtual machines (running ubuntu/focal64) . The first VM is a web server that users interact with to make a booking. The second VM is a database server that stores the bookings made by users and admin login details with the use of MySQL. And the third VM is an admin server that hosts the admin page that shows the bookings made by users from the database.

It is necessary to build this application using three VM's due to the massive increase in security it provides. You do not want users to be able to access the database as that will cause massive security problems (data leaks etc.) and leave the app susceptible to malicious people. You also do not want Users to be able to access the admin page because that again can cause the same issues like data leaks. Having separate VM's also makes maintenance and upgrading easier as you can change files for one VM without affecting the other VM's.

## Initial build

Cloning the repository will create the folder COSC349-a1 which has a total size of 59Mb. Cd into the COSC349-a1 directory and run vagrant up will start downloading packages. The initial build takes approximately 6 minutes.

### **Web Server (#VM1): 30.1Mb**

Ubuntu security/updates packages - 24.1Mb

Apache packages - 6Mb

**Database (#VM2):** 56.2Mb

Ubuntu security/updates packages - 24.1Mb

MySQL packages - 32.1Mb

**Admin (#VM3):** 30.1Mb

Ubuntu security/updates packages - 24.1Mb

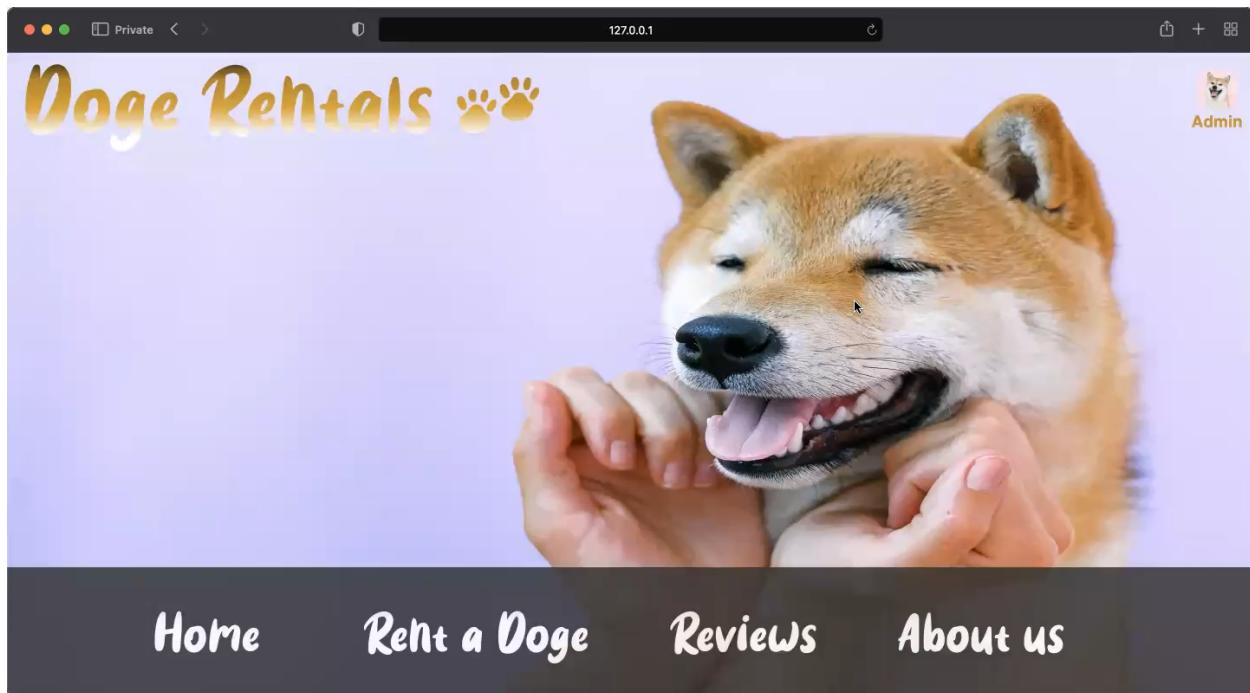
Apache packages - 6Mb

## Redeployment

When running vagrant halt then redeploying the build with vagrant up no extra packages will be downloaded however it will take approximately 2 minutes to run the build process.

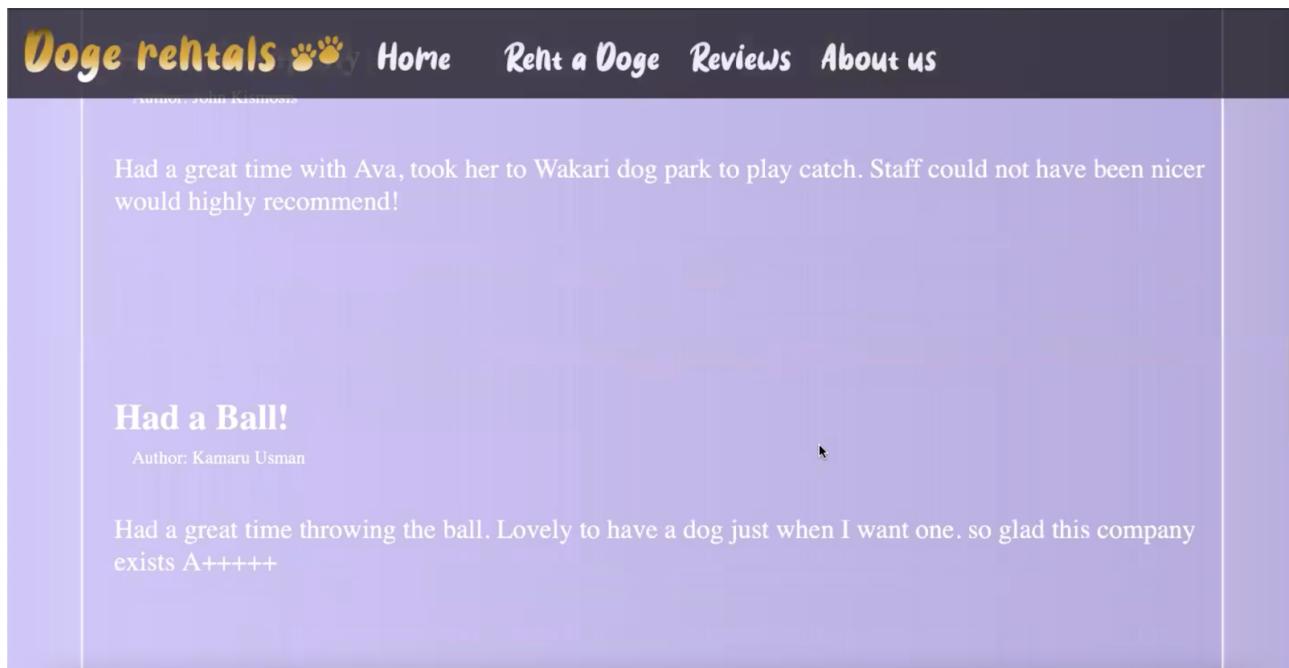
## Recording:

0:00 - 0:04



This is the index page of the web server (#VM1) it is hosted on 127.0.0.1:8080. The admin icon in the top right is used to access the admin page. The user can use the header to go to three different web pages. Rent a Doge, Reviews, or About us.

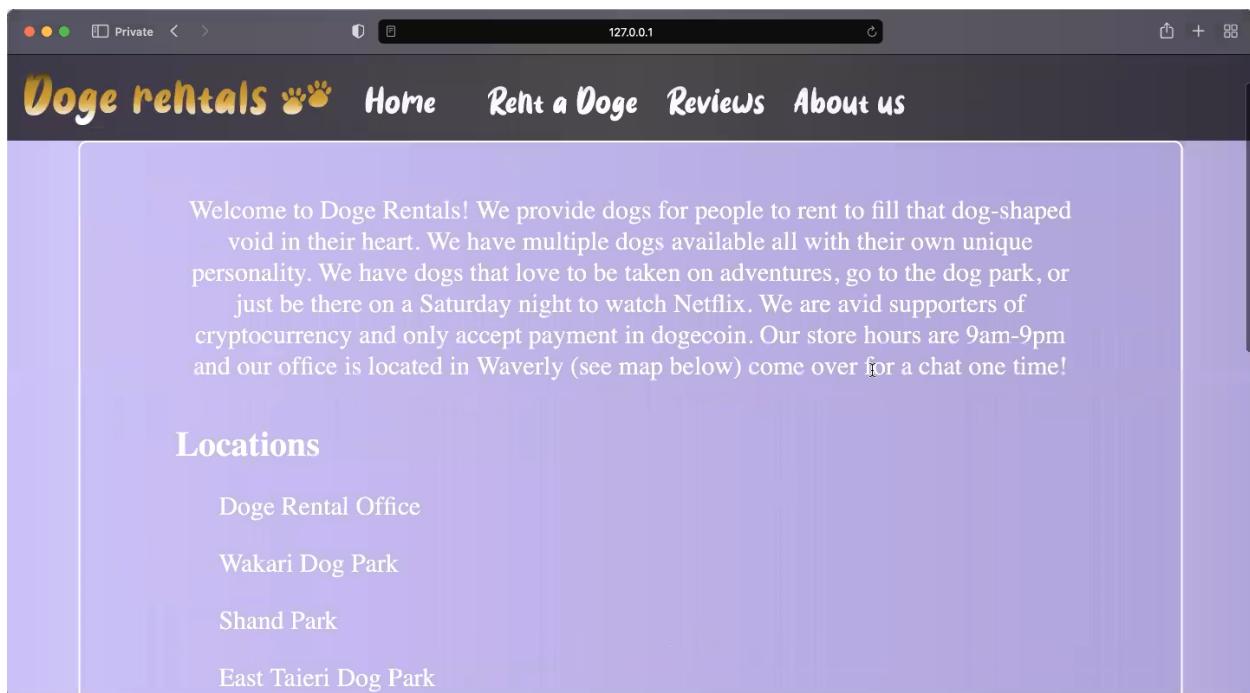
0:05 - 0:08



The screenshot shows a website for 'Doge rentals' with a dark header. The header includes the logo 'Doge rentals' with a paw print icon, and navigation links for 'Home', 'Rent a Doge', 'Reviews', and 'About us'. Below the header, there is a purple content area containing two reviews. The first review is from 'Author: John Kishmire' and says: 'Had a great time with Ava, took her to Wakari dog park to play catch. Staff could not have been nicer would highly recommend!'. The second review is from 'Author: Kamaru Usman' and says: 'Had a great time throwing the ball. Lovely to have a dog just when I want one. so glad this company exists A+++++'. The background of the content area is purple.

This is the reviews page for the website which displays reviews left by other users. The reviews are read from a JSON file.

0:09 - 0:25

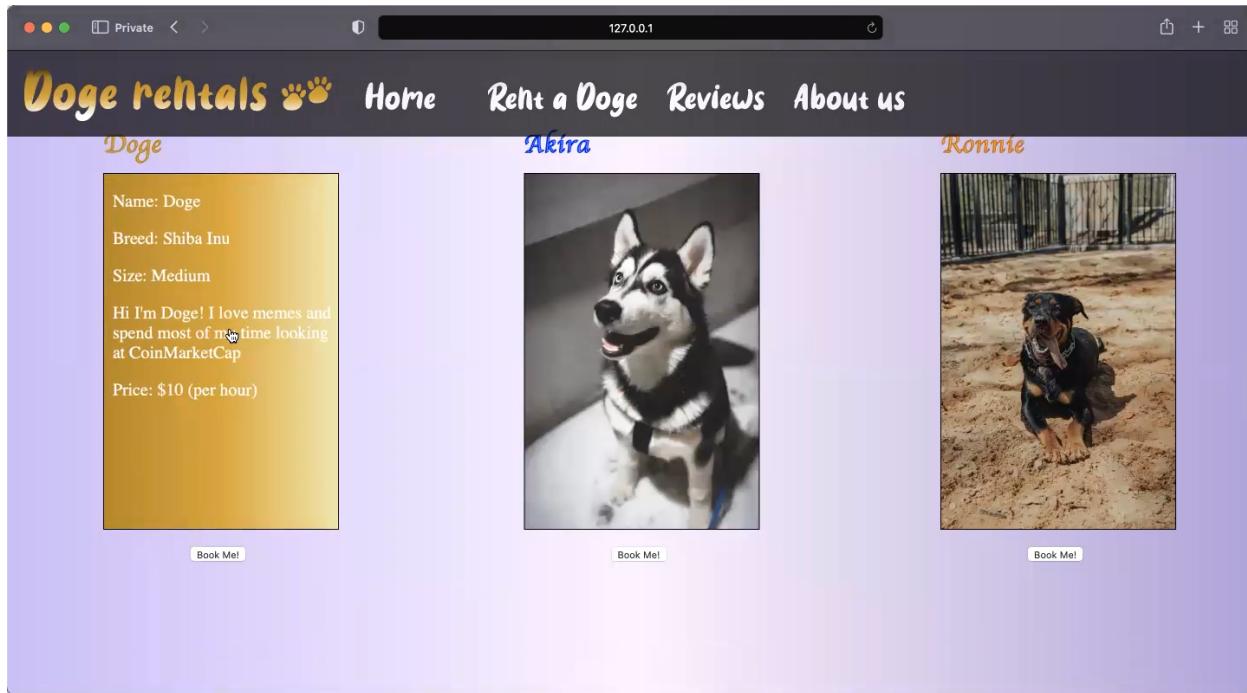


The screenshot shows a website for 'Doge rentals' with a dark header. The header includes the logo 'Doge rentals' with a paw print icon, and navigation links for 'Home', 'Rent a Doge', 'Reviews', and 'About us'. Below the header, there is a purple content area. The first part of the content is a welcome message: 'Welcome to Doge Rentals! We provide dogs for people to rent to fill that dog-shaped void in their heart. We have multiple dogs available all with their own unique personality. We have dogs that love to be taken on adventures, go to the dog park, or just be there on a Saturday night to watch Netflix. We are avid supporters of cryptocurrency and only accept payment in dogecoin. Our store hours are 9am-9pm and our office is located in Waverly (see map below) come over for a chat one time!'. Below this message, there is a section titled 'Locations' with a list of locations: 'Doge Rental Office', 'Wakari Dog Park', 'Shand Park', and 'East Taieri Dog Park'. The background of the content area is purple.

This is the about us page. It contains a nice description about the company. Scrolling down reveals a leaflet map that is used to show key locations. The red marker is the location of the business, Yellow markers are Dog parks, and Blue markers are walking

tracks. The buttons underneath the map can be used to show/hide the tracks/parks and the location names can be clicked to focus the map onto that location. The locations are read from a GEOJSON file.

0:26 - 0:43



This is the Rent a Doge page. You can click on the image of the dog to get a description about the dog names, breed, size etc. Then click again to spin back to the picture of the dog. You can book at maximum three dogs at a time. The dogs chosen are stored in local storage so you can leave the web page and go back and your picks will be remembered.

0:44-1:07

Private  127.0.0.1   

# Doge rentals

[Home](#) [Rent a Doge](#) [Reviews](#) [About us](#)

**Booking Details:**

Please enter your name:

Date:

Time of pickup:

Number of hours:

Please note: You must return dogs by 9pm the same day.

**Payment Details:**

Card type:

Card number:

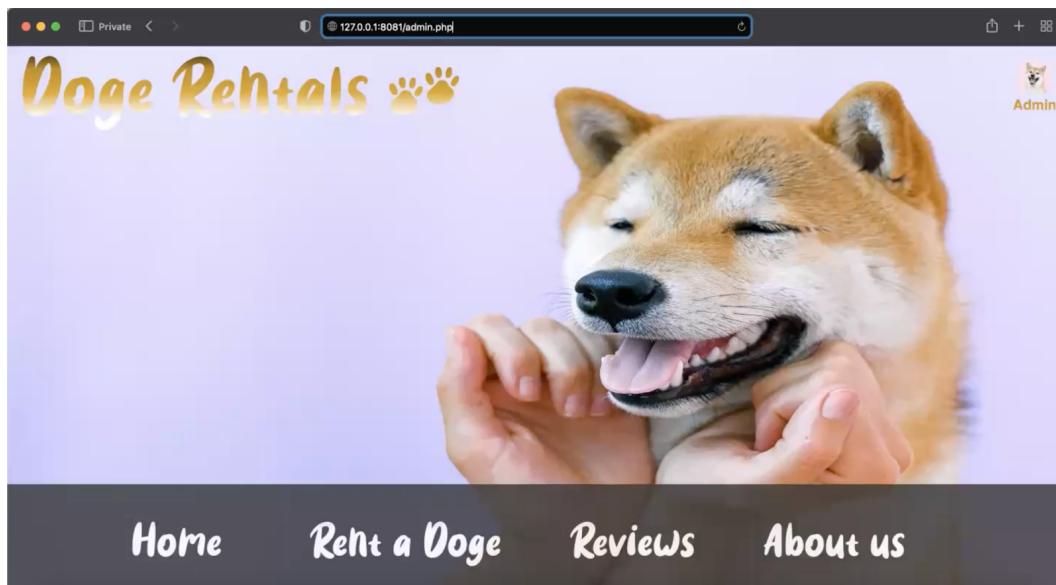
Expiry date:

CVC:

[Cancel Booking](#) [Submit](#)

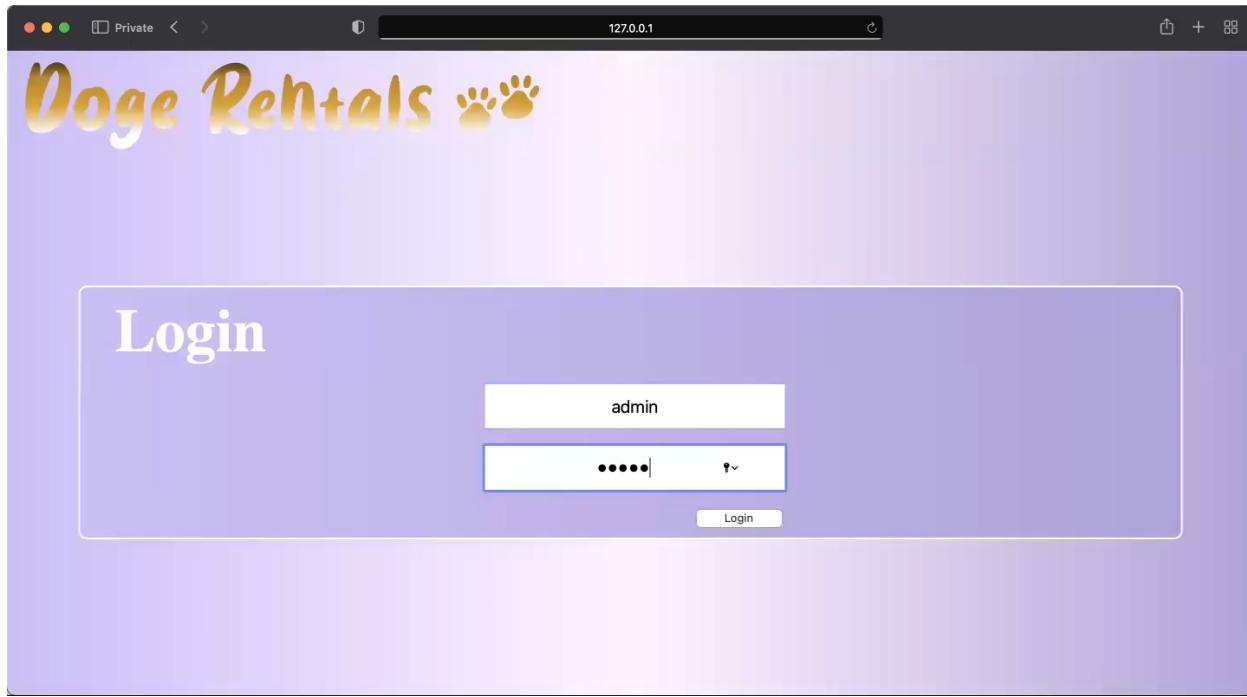
After selecting dogs a form will appear at the bottom of the page. You can use the buttons at the bottom to either cancel the booking, or to submit the form. If you enter details into the form incorrectly error messages will appear at the bottom of the form detailing what you got wrong. When you enter details correctly the form will submit and give you a message saying your booking was successful and insert your booking details into the bookings table in the database server (#VM2).

1:08-1:18



I entered the domain of the admin page 127.0.0.1:8081/admin.php to show that you can not access the admin page without first successfully logging in. I use a session to make sure that the user has logged in to access the admin page.

1:19-1:28

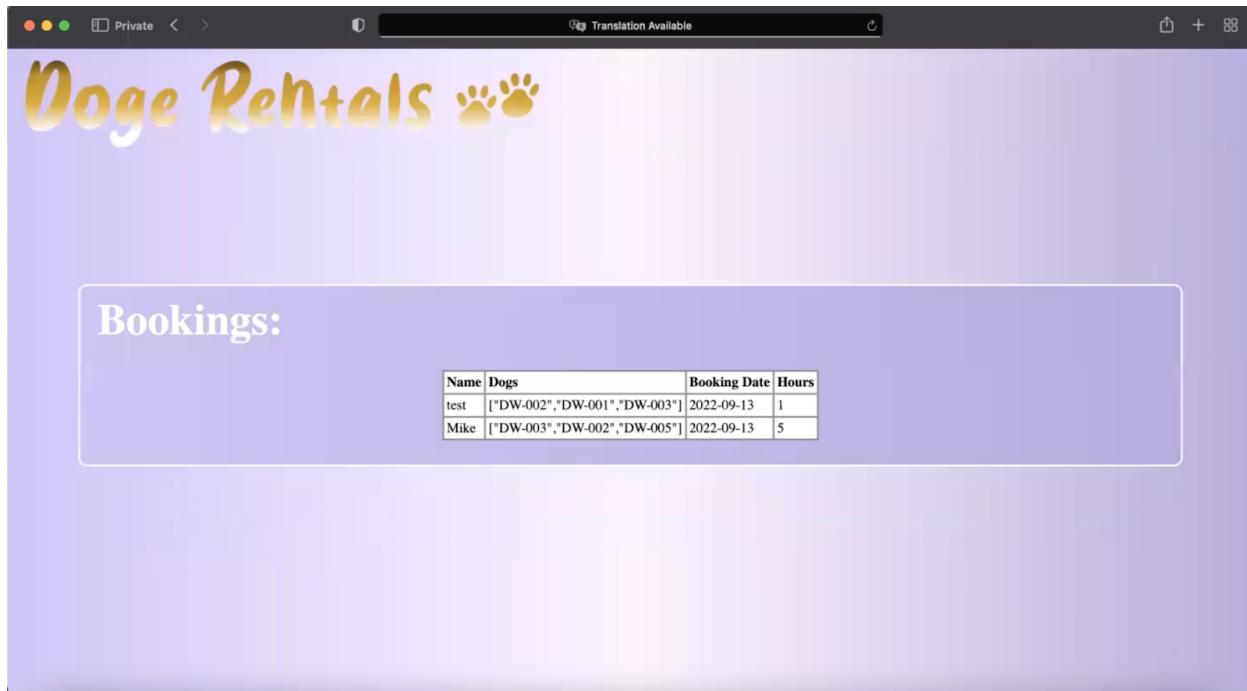


Simple login page to access the admin web page. I first enter details incorrectly to show that if you enter details wrong you will get error messages that appear. When the username and password are entered correctly a session will be created to remember you have logged in allowing you to access the admin page.

Username: "admin"

Password: "admin"

The username and password is stored in the database server (#VM2) in the admin table; the password is encrypted in the database using SHA1 encryption for extra security.



This is the admin page that is hosted by the admin server (#VM3) at 127.0.0.1:8081/admin.php. It reads from the bookings table from the database server (#VM2) and displays the bookings made for the admin to view.

## Changes:

- A developer may want to delete the JSON files used for the Reviews page and read reviews from a database instead and also allow users to enter a review. To do this a developer will just want to edit the files setup-database.sql and create a reviews table to store the reviews. Edit www/reviews.html to change the layout to allow a user to enter a review and change the file to a php file to read reviews from the database and display them. And delete reviews.js as it will no longer be needed. The developer will then need to save changes and run vagrant up in order to rebuild the application.
- A developer may want to change the admin.php page. Specifically they may want to add the option to allow an admin to edit the database or to delete entries in the database. To make these changes a developer will only need to change admin/admin.php as it will just be queries to the table. They will not need to

rebuild the application; they could just save the changes and the changes will appear.