Digital Commons @ LMU and LLS

Loyola Marymount University and Loyola Law School
Digital Commons at Loyola Marymount University and Loyola Law School

Honors Thesis

Honors Program

5-3-2017

Parkview Pet Hospital: Veterinary Practice Management System Development

Ian T. Wiedenman
Loyola Marymount University, ian.wiedenman@gmail.com

LMU | LA
University Honors
Program

Parkview Pet Hospital: Veterinary Practice Management System Development

A thesis submitted in partial satisfaction

of the requirements of the University Honors Program

of Loyola Marymount University

by

Ian Wiedenman

May 3rd, 2017

Introduction

Background

We are Ian Wiedenmann and Taylor Kozakar, AIMS Seniors. Prior to the beginning of this project, we both had an interest in healthcare IT systems, as well as database and server development and deployment in a professional environment. As such, our project for Parkview Pet Hospital was an intriguing and stimulating project.

Parkview Pet Hospital is a small but full-service animal hospital that deals with routine checkup for dogs and cats as well as emergency cases. Parkview Pet hospital is a comfortable, kid-friendly, and calm environment. They were established in 2009 and currently have one veterinarian, Dr. Gamble, and 3-4 technicians on full-time staff fulfilling a number of roles, including administration, technical expertise, nursing, and various other defined roles.

The Primary Problem Statement

How does Parkview Pet Hospital maintain an operational, enterprise RDBMS with accompanying front-end, while keeping costs at a minimum and maintaining ease of use?

Exploration of the Problem Statement and Enterprise Requirements

Parkview Pet Hospital is continuously searching for new deployments in their information technology architecture. As such, the organization needs a centralized system for recording and tracking key business processes and the data/metadata they create, as well as storing data regarding the company's inventory, employees, clients, client’s pets, and critical organizational actions. The problem, as it stands, is that the hospital lacks a robust system for storage of data, as well as templates for data entry, and reporting services that reliably produce the properly formatted data that serves both employees and clients alike. Parkview Pet Hospital has used many practice management softwares in the past but the cons often outweigh the pros.

Therefore, there is a need for an easy-to-use, low-cost/free, open-source database system for operational enterprise data storage, as well as a dependable website-hosting system and accompanying database. Finally, there is a need for a well-designed, modern website that allows for easier customer interaction, as well as a portal for staff to quickly execute queries on patient and client records.

The Development Process

Initial Steps

In the early process, we utilized a combination of Kimball data discovery techniques and traditional approaches to data discovery/requirements gathering. We interviewed users of the data and systems to determine needs and gaps, as well as questions regarding the general nature of the enterprise, as well as their future plans and what they could need in the future. As such, this information was ascertained through: phone/in-person interviews with primary physician and hospital administrator, in-person discussion with IT staff member, brief discussions with clients regarding the website interface, and analysis of current, high-cost system (FileMaker).

Early Database Development Phase

Using the information we gathered through the interviews, we first constructed an ER model that best fit the needs of the organization. See the attached ER diagram.

After settling on a general schema structure, we began developing the necessary fields and data types required for accurately storing organizational and client data. See the attached field names.

Initially, we used these table structures and field name delineations to build out a an instance of MySQL Server (on an InnoDB database engine) hosted locally on our MacBook PRO OSX laptop. MySQL Workbench was the RDBMS software utilized to build out and manage this database, as well as input sample data. Accessing the localhost instance of MySQL Server at 127.0.0.1 (localhost) with UNIX socket port of 3306 was the primary means of linking the two pieces of software. This was implemented with the SQL language, on top of the MySQL Workbench platform.

Early Web Development Phase

Since this project requires a front-end for end-users, we designed a sitemap that delineated all of the pages of our site in a hierarchical formation. See attached sitemap.

Next, we used the sitemap as a guide to create a storyboard for the website which shows a rough outline of what the pages will look like as well as their functionality. See attached storyboard slides.

After developing the blueprint materials for the web development phase, we used a bootstrap template by the name of AdminLTE to kickstart out front-end development. Having a nicely formatted wireframe with built-in grid systems allowed us to streamline our backend development and fit our code into the front-end site, and helped us concentrate more of our time and effort into database development, PHP programming, SQL script deployment, and architectural configuration.

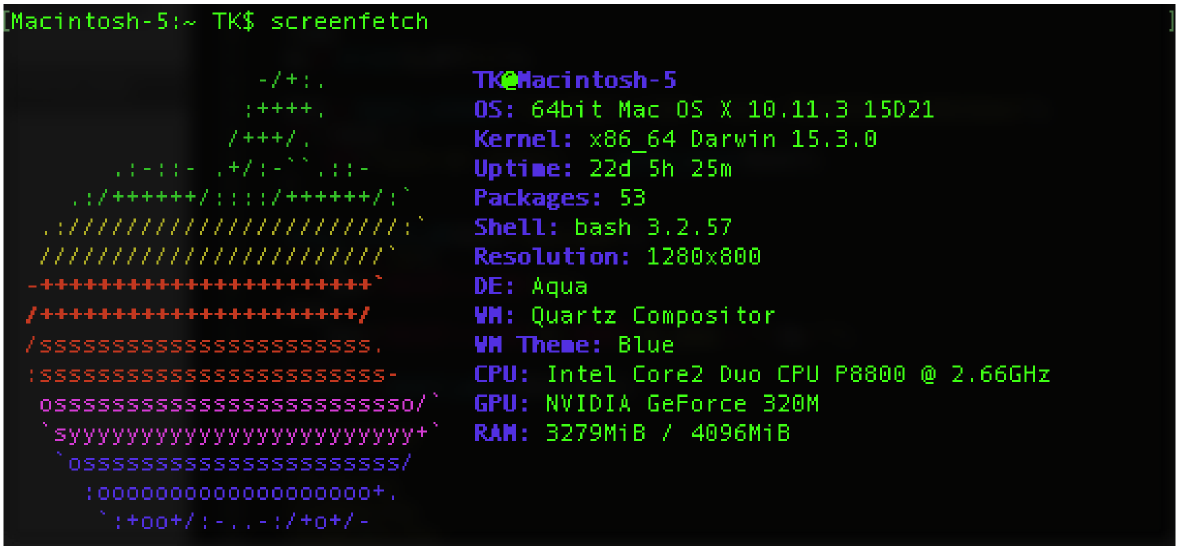
This phase was implemented with HTML/CSS, Javascript, jQuery, PHP, and the Admin LTE bootstrap templates.

The MAMP Phase

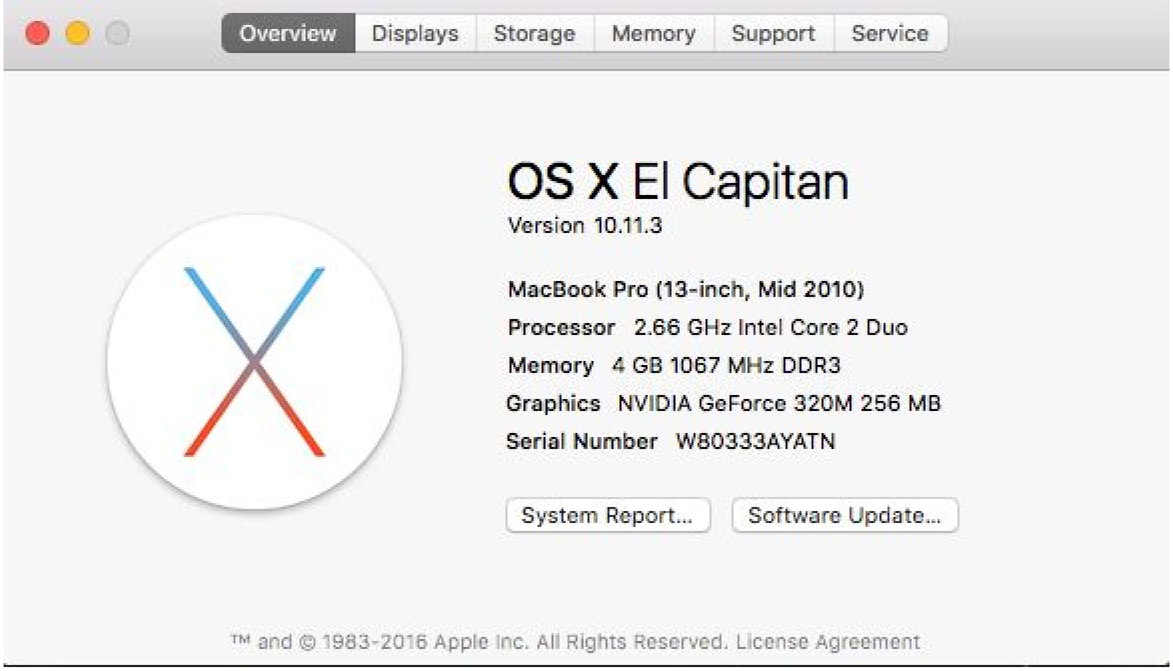
At this point, we had a working instance of MySQL Server and a functioning website, but no adequate way of having the two platforms interact or exchange data. As such, we realized we needed to implement some form of webserver software (most likely Apache), as well as extensive PHP code to facilitate interaction. Instead of writing a large amount of middleware in C and cranking out Unix BASH scripts, we decided to install and use MAMP Pro, a webserver virtualization platform that integrates Apache Webserver, MySQL Server, PHP/Python/Perl, and the Mac OSX operating system environment. See below for configuration screenshots that demonstrate the necessary settings for running a virtualized server environment.

Prototype Implementation

Hardware/OS configurations:



```
[Macintosh-5:~ TK$ screenfetch                                                       ]

                 -/+:.          TK@Macintosh-5
                :++++.          OS: 64bit Mac OS X 10.11.3 15D21
               /+++/.           Kernel: x86_64 Darwin 15.3.0
       .:-::- .+/:-``.::-       Uptime: 22d 5h 25m
    .:/++++++/::::/++++++/:`    Packages: 53
  .:///////////////////////:`   Shell: bash 3.2.57
  ////////////////////////`     Resolution: 1280x800
 -+++++++++++++++++++++++`      DE: Aqua
 /++++++++++++++++++++++/       WM: Quartz Compositor
 /sssssssssssssssssssssss.      WM Theme: Blue
 :ssssssssssssssssssssssss-     CPU: Intel Core2 Duo CPU P8800 @ 2.66GHz
  osssssssssssssssssssssssso/`  GPU: NVIDIA GeForce 320M
  `syyyyyyyyyyyyyyyyyyyyyyyy+`  RAM: 3279MiB / 4096MiB
   `ossssssssssssssssssssss/
     :ooooooooooooooooooo+.
      `:+oo+/:-..-:/+o+/-
```



OS X El Capitan

Version 10.11.3

MacBook Pro (13-inch, Mid 2010)

Processor 2.66 GHz Intel Core 2 Duo

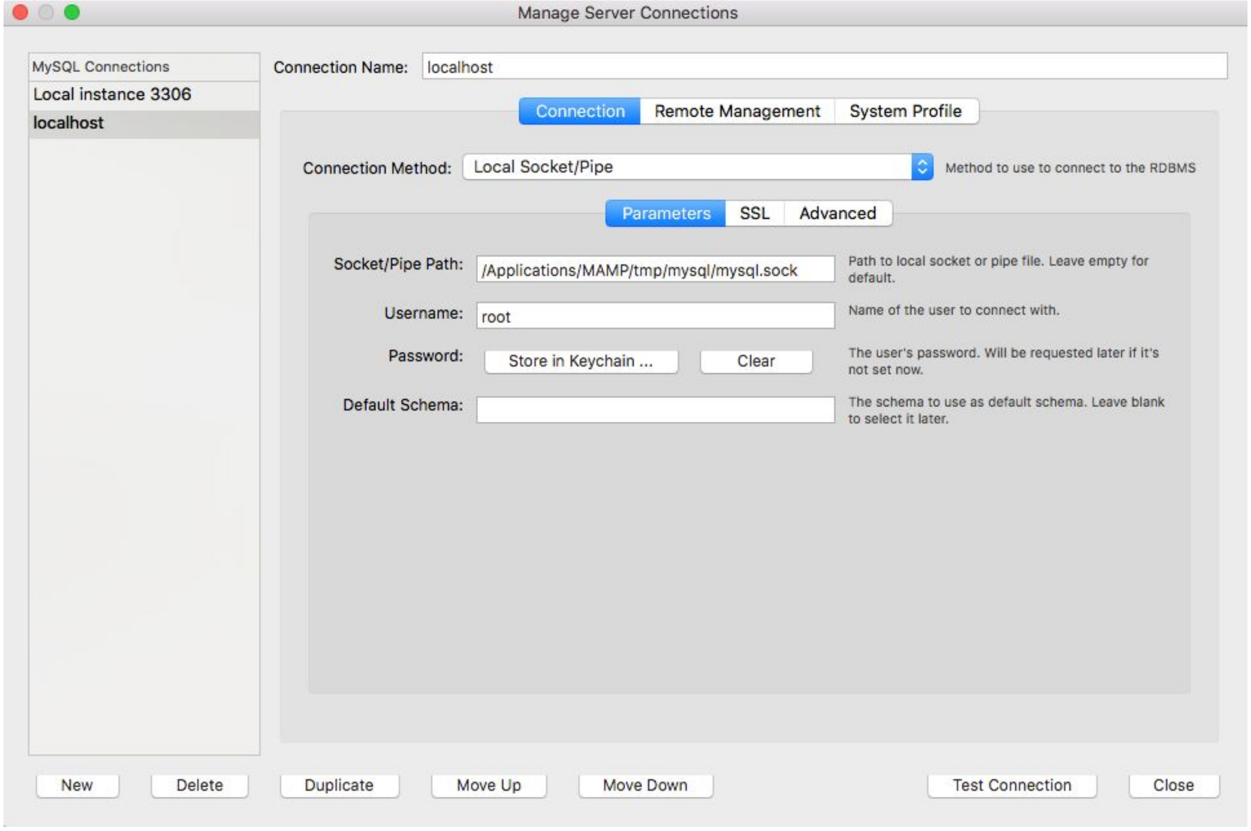
Memory 4 GB 1067 MHz DDR3

Graphics NVIDIA GeForce 320M 256 MB

Serial Number W80333AYATN

™ and © 1983-2016 Apple Inc. All Rights Reserved. License Agreement

MySQL Workbench: Connection



Connection Name: localhost

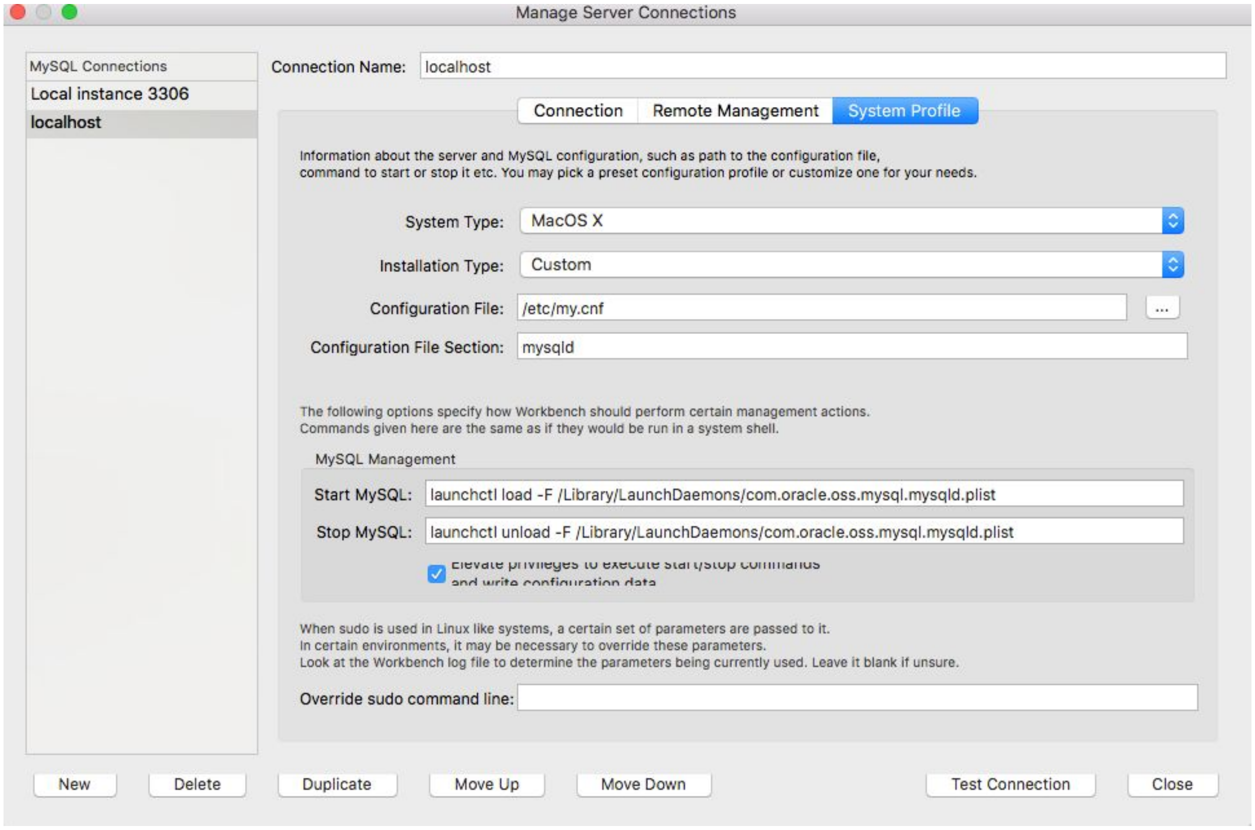
Connection Method: Local Socket/Pipe — Method to use to connect to the RDBMS

Socket/Pipe Path: /Applications/MAMP/tmp/mysql/mysql.sock — Path to local socket or pipe file. Leave empty for default.

Username: root — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set now.

Default Schema: — The schema to use as default schema. Leave blank to select it later.



Connection Name: localhost

Information about the server and MySQL configuration, such as path to the configuration file, command to start or stop it etc. You may pick a preset configuration profile or customize one for your needs.

System Type: MacOS X

Installation Type: Custom

Configuration File: /etc/my.cnf

Configuration File Section: mysqld

The following options specify how Workbench should perform certain management actions. Commands given here are the same as if they would be run in a system shell.

MySQL Management

Start MySQL: launchctl load -F /Library/LaunchDaemons/com.oracle.oss.mysql.mysqld.plist

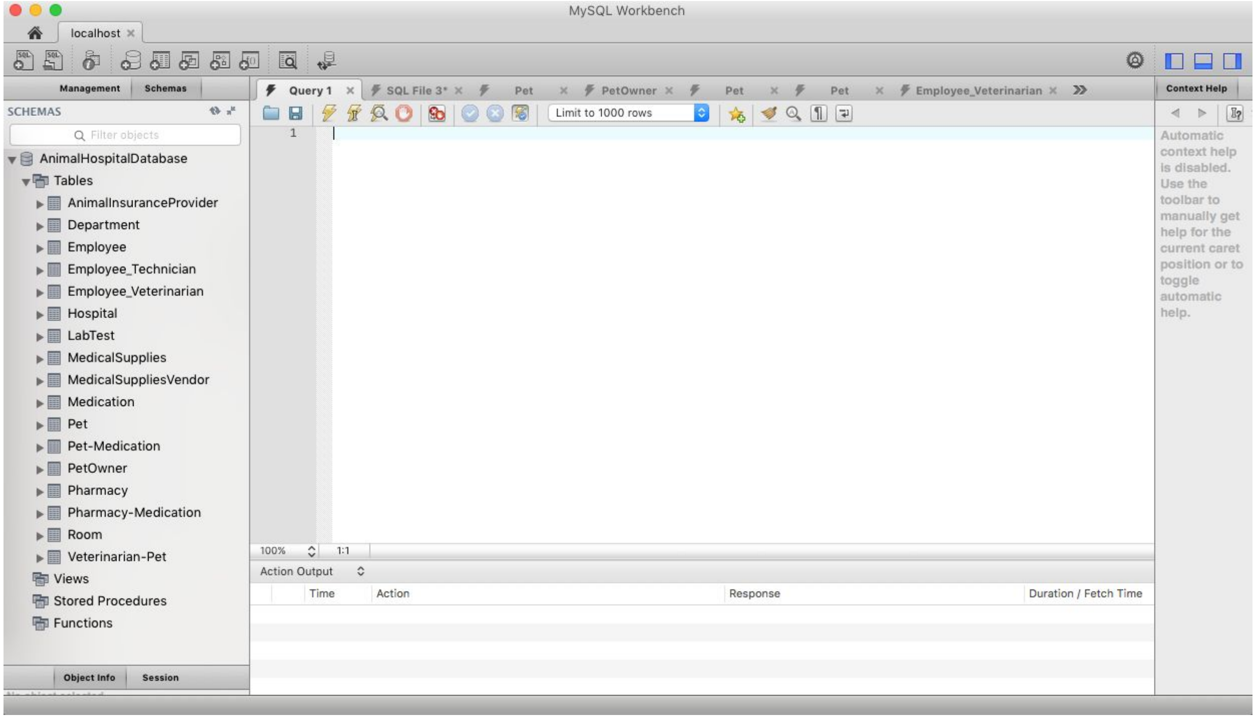
Stop MySQL: launchctl unload -F /Library/LaunchDaemons/com.oracle.oss.mysql.mysqld.plist

☑ Elevate privileges to execute start/stop commands and write configuration data

When sudo is used in Linux like systems, a certain set of parameters are passed to it. In certain environments, it may be necessary to override these parameters. Look at the Workbench log file to determine the parameters being currently used. Leave it blank if unsure.

Override sudo command line:

MySQL Workbench: All Tables



MySQL Workbench

localhost

Query 1 | SQL File 3* | Pet | PetOwner | Pet | Pet | Employee_Veterinarian

Limit to 1000 rows

SCHEMAS

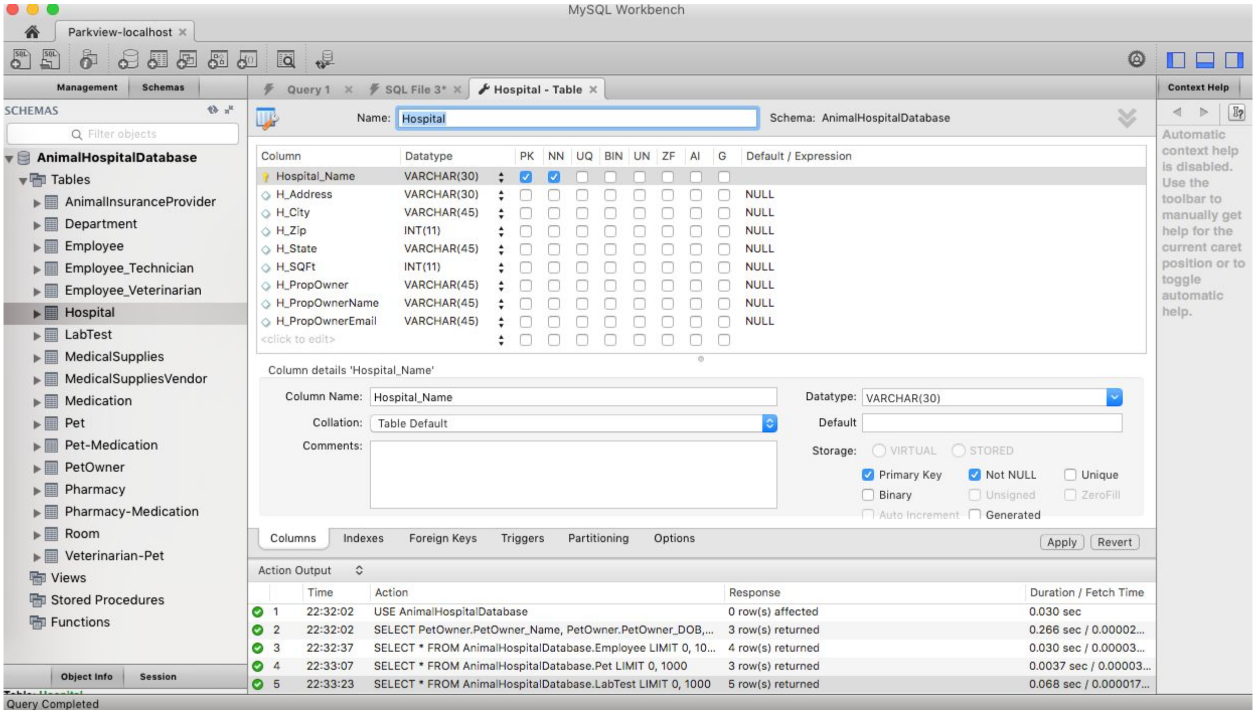
- AnimalHospitalDatabase
  - Tables
    - AnimalInsuranceProvider
    - Department
    - Employee
    - Employee_Technician
    - Employee_Veterinarian
    - Hospital
    - LabTest
    - MedicalSupplies
    - MedicalSuppliesVendor
    - Medication
    - Pet
    - Pet-Medication
    - PetOwner
    - Pharmacy
    - Pharmacy-Medication
    - Room
    - Veterinarian-Pet
  - Views
  - Stored Procedures
  - Functions

Context Help: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Action Output

	Time	Action	Response	Duration / Fetch Time

MySQL Workbench: Configuration of Hospital Table



MySQL Workbench

Parkview-localhost

Query 1 | SQL File 3* | Hospital - Table

SCHEMAS

- AnimalHospitalDatabase
  - Tables
    - AnimalInsuranceProvider
    - Department
    - Employee
    - Employee_Technician
    - Employee_Veterinarian
    - Hospital
    - LabTest
    - MedicalSupplies
    - MedicalSuppliesVendor
    - Medication
    - Pet
    - Pet-Medication
    - PetOwner
    - Pharmacy
    - Pharmacy-Medication
    - Room
    - Veterinarian-Pet
  - Views
  - Stored Procedures
  - Functions

Name: Hospital   Schema: AnimalHospitalDatabase

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
Hospital_Name	VARCHAR(30)	☑	☑	□	□	□	□	□	□	
H_Address	VARCHAR(30)	□	□	□	□	□	□	□	□	NULL
H_City	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
H_Zip	INT(11)	□	□	□	□	□	□	□	□	NULL
H_State	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
H_SQFt	INT(11)	□	□	□	□	□	□	□	□	NULL
H_PropOwner	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
H_PropOwnerName	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
H_PropOwnerEmail	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
<click to edit>		□	□	□	□	□	□	□	□	

Column details 'Hospital_Name'

Column Name: Hospital_Name

Collation: Table Default

Comments:

Datatype: VARCHAR(30)

Default:

Storage: ○ VIRTUAL ○ STORED

☑ Primary Key ☑ Not NULL □ Unique

□ Binary □ Unsigned □ ZeroFill

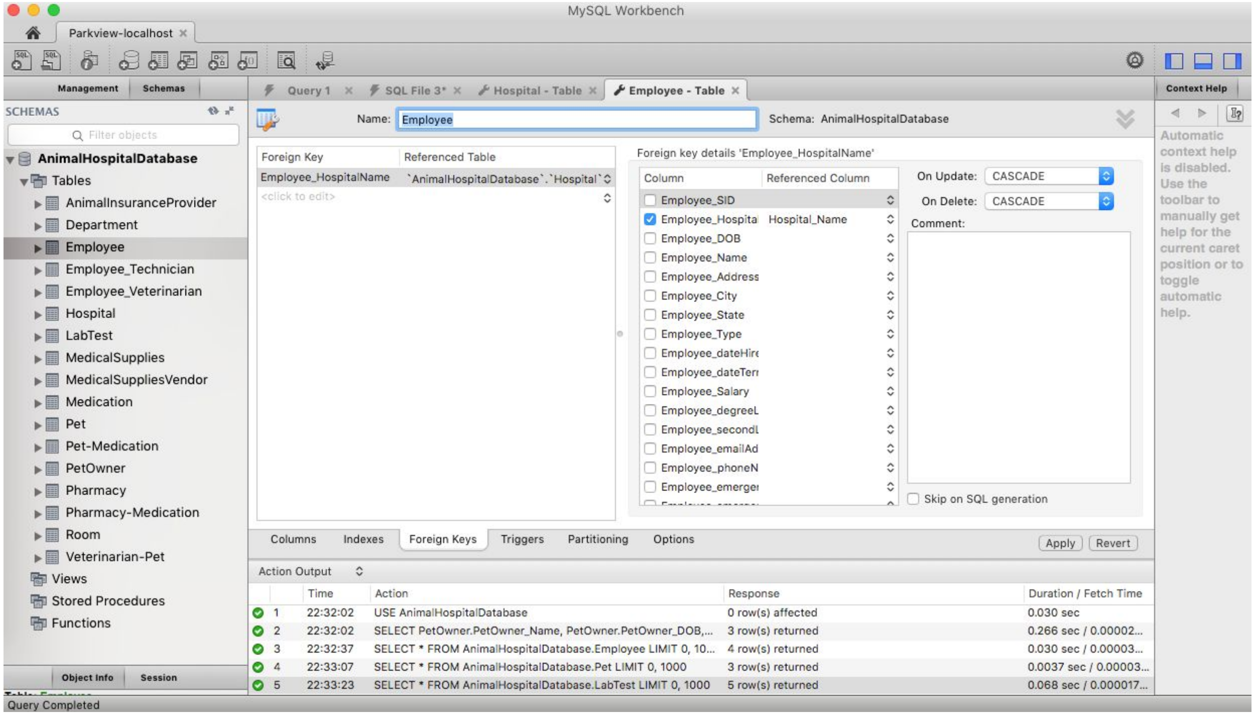
□ Auto Increment □ Generated

Action Output

	Time	Action	Response	Duration / Fetch Time
1	22:32:02	USE AnimalHospitalDatabase	0 row(s) affected	0.030 sec
2	22:32:02	SELECT PetOwner.PetOwner_Name, PetOwner.PetOwner_DOB,...	3 row(s) returned	0.266 sec / 0.00002...
3	22:32:37	SELECT * FROM AnimalHospitalDatabase.Employee LIMIT 0, 10...	4 row(s) returned	0.030 sec / 0.00003...
4	22:33:07	SELECT * FROM AnimalHospitalDatabase.Pet LIMIT 0, 1000	3 row(s) returned	0.0037 sec / 0.00003...
5	22:33:23	SELECT * FROM AnimalHospitalDatabase.LabTest LIMIT 0, 1000	5 row(s) returned	0.068 sec / 0.000017...

Query Completed

MySQL Workbench: Configuration of Employee Foreign Keys



SCHEMAS

AnimalHospitalDatabase
- Tables
  - AnimalInsuranceProvider
  - Department
  - Employee
  - Employee_Technician
  - Employee_Veterinarian
  - Hospital
  - LabTest
  - MedicalSupplies
  - MedicalSuppliesVendor
  - Medication
  - Pet
  - Pet-Medication
  - PetOwner
  - Pharmacy
  - Pharmacy-Medication
  - Room
  - Veterinarian-Pet
- Views
- Stored Procedures
- Functions

Name: Employee   Schema: AnimalHospitalDatabase

Foreign Key	Referenced Table
Employee_HospitalName	\`AnimalHospitalDatabase\`.\`Hospital\`
<click to edit>	

Foreign key details 'Employee_HospitalName'

Column	Referenced Column
□ Employee_SID	
☑ Employee_Hospita	Hospital_Name
□ Employee_DOB	
□ Employee_Name	
□ Employee_Address	
□ Employee_City	
□ Employee_State	
□ Employee_Type	
□ Employee_dateHire	
□ Employee_dateTerr	
□ Employee_Salary	
□ Employee_degreeL	
□ Employee_secondL	
□ Employee_emailAd	
□ Employee_phoneN	
□ Employee_emerger	

On Update: CASCADE

On Delete: CASCADE

Comment:

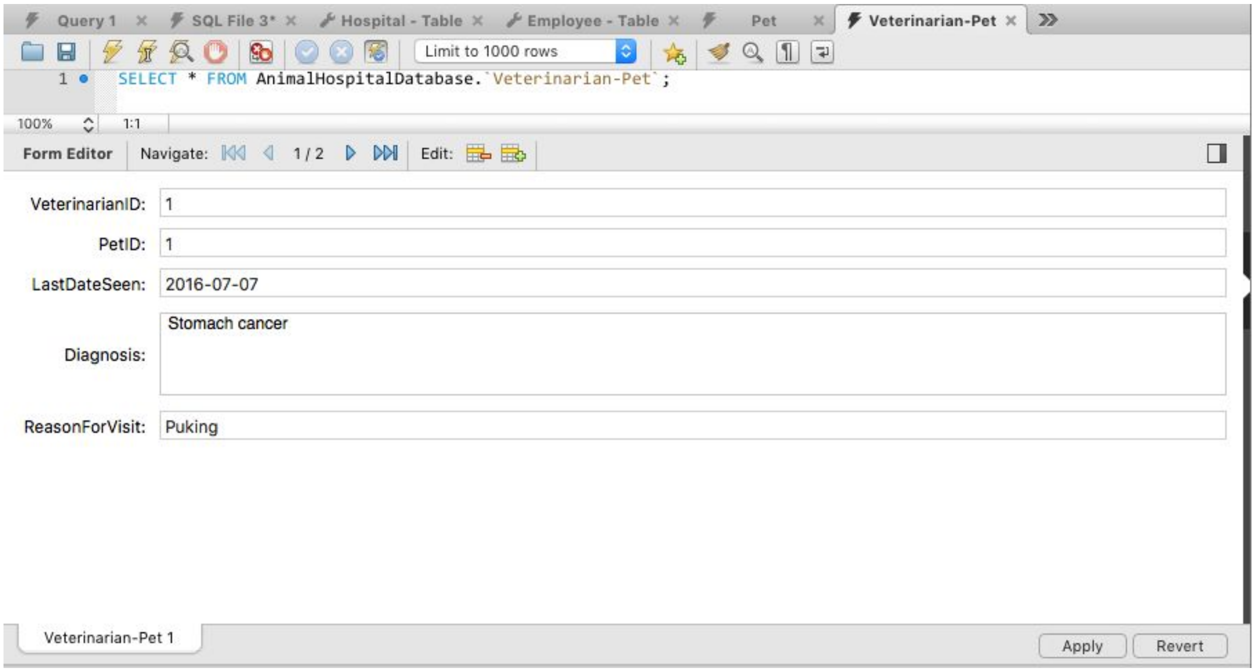
□ Skip on SQL generation

Action Output

		Time	Action	Response	Duration / Fetch Time
✓	1	22:32:02	USE AnimalHospitalDatabase	0 row(s) affected	0.030 sec
✓	2	22:32:02	SELECT PetOwner.PetOwner_Name, PetOwner.PetOwner_DOB,...	3 row(s) returned	0.266 sec / 0.00002...
✓	3	22:32:37	SELECT * FROM AnimalHospitalDatabase.Employee LIMIT 0, 10...	4 row(s) returned	0.030 sec / 0.00003...
✓	4	22:33:07	SELECT * FROM AnimalHospitalDatabase.Pet LIMIT 0, 1000	3 row(s) returned	0.0037 sec / 0.00003...
✓	5	22:33:23	SELECT * FROM AnimalHospitalDatabase.LabTest LIMIT 0, 1000	5 row(s) returned	0.068 sec / 0.000017...

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

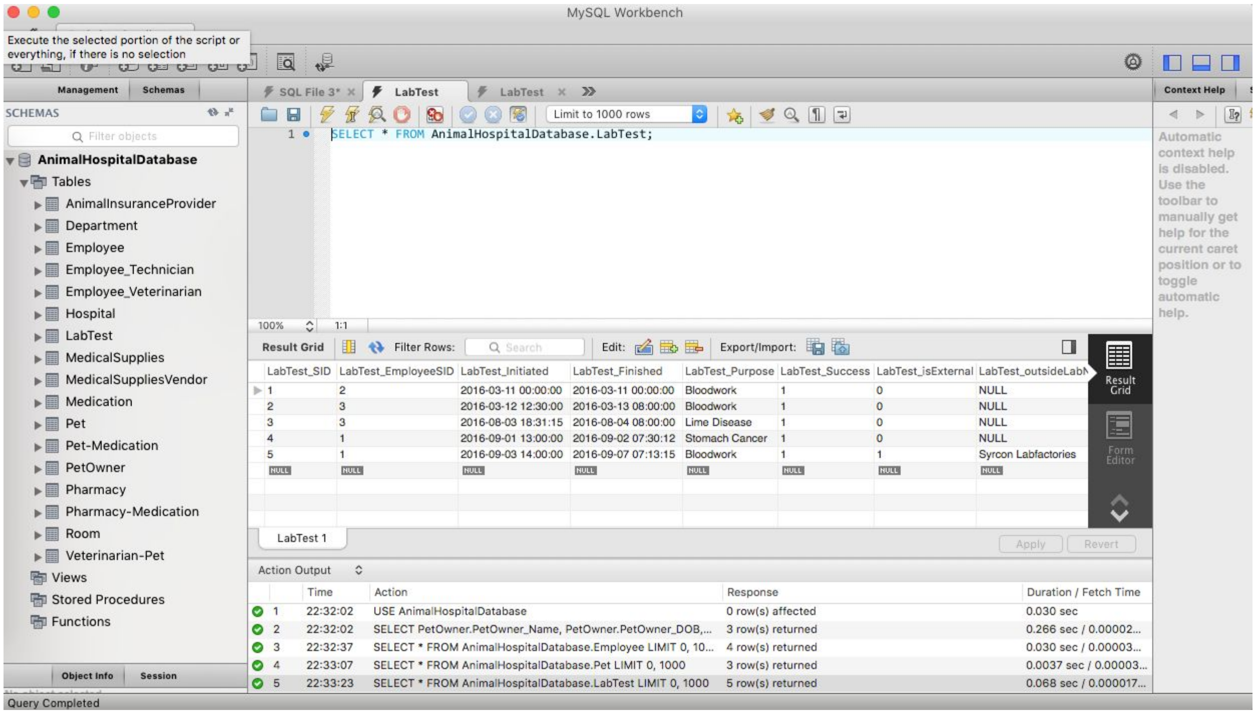
MySQL Workbench: Editing Records in the Veterinarian-Pet Table



```
SELECT * FROM AnimalHospitalDatabase.`Veterinarian-Pet`;
```

VeterinarianID:	1
PetID:	1
LastDateSeen:	2016-07-07
Diagnosis:	Stomach cancer
ReasonForVisit:	Puking

MySQL Workbench: Basic query of LabTest table:

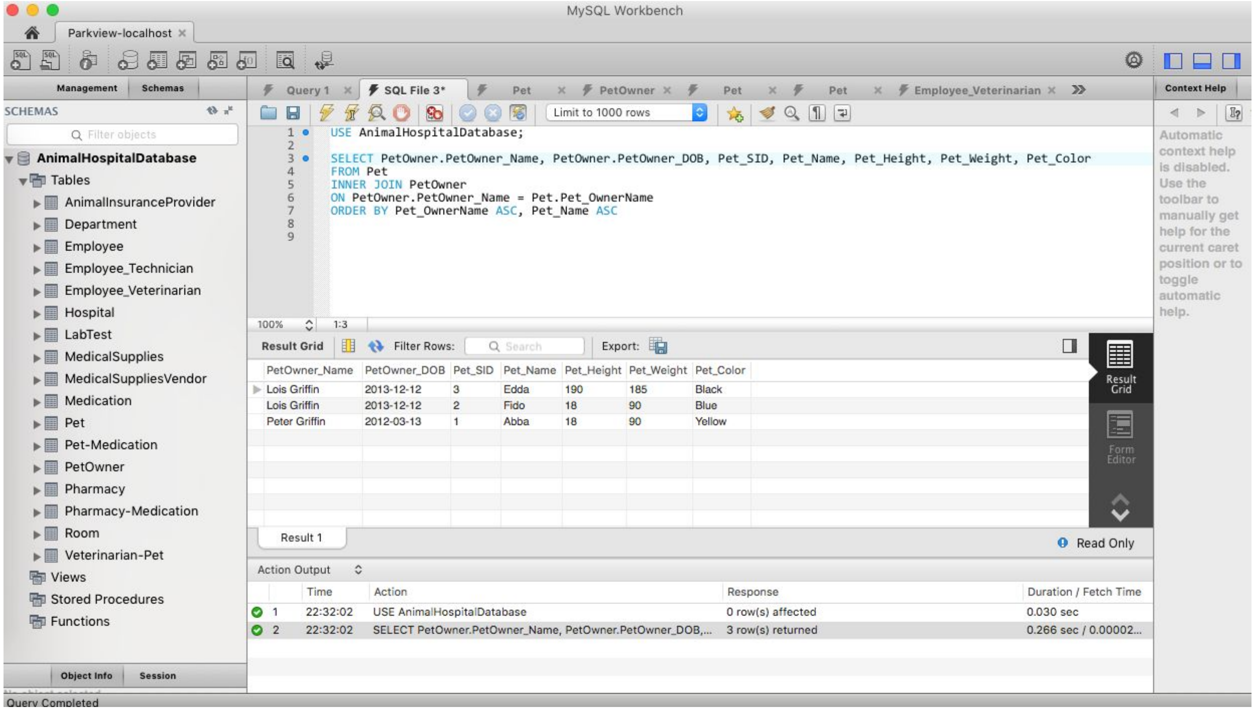


```
SELECT * FROM AnimalHospitalDatabase.LabTest;
```

LabTest_SID	LabTest_EmployeeSID	LabTest_Initiated	LabTest_Finished	LabTest_Purpose	LabTest_Success	LabTest_isExternal	LabTest_outsideLabN
1	2	2016-03-11 00:00:00	2016-03-11 00:00:00	Bloodwork	1	0	NULL
2	3	2016-03-12 12:30:00	2016-03-13 08:00:00	Bloodwork	1	0	NULL
3	3	2016-08-03 18:31:15	2016-08-04 08:00:00	Lime Disease	1	0	NULL
4	1	2016-09-01 13:00:00	2016-09-02 07:30:12	Stomach Cancer	1	0	NULL
5	1	2016-09-03 14:00:00	2016-09-07 07:13:15	Bloodwork	1	1	Syrcon Labfactories
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
   Time      Action                                                   Response             Duration / Fetch Time
1  22:32:02  USE AnimalHospitalDatabase                               0 row(s) affected    0.030 sec
2  22:32:02  SELECT PetOwner.PetOwner_Name, PetOwner.PetOwner_DOB,... 3 row(s) returned    0.266 sec / 0.00002...
3  22:32:37  SELECT * FROM AnimalHospitalDatabase.Employee LIMIT 0, 10... 4 row(s) returned 0.030 sec / 0.00003...
4  22:33:07  SELECT * FROM AnimalHospitalDatabase.Pet LIMIT 0, 1000   3 row(s) returned    0.0037 sec / 0.00003...
5  22:33:23  SELECT * FROM AnimalHospitalDatabase.LabTest LIMIT 0, 1000 5 row(s) returned  0.068 sec / 0.000017...
```

MySQL Workbench: Advanced query of PetOwner and Pet tables:

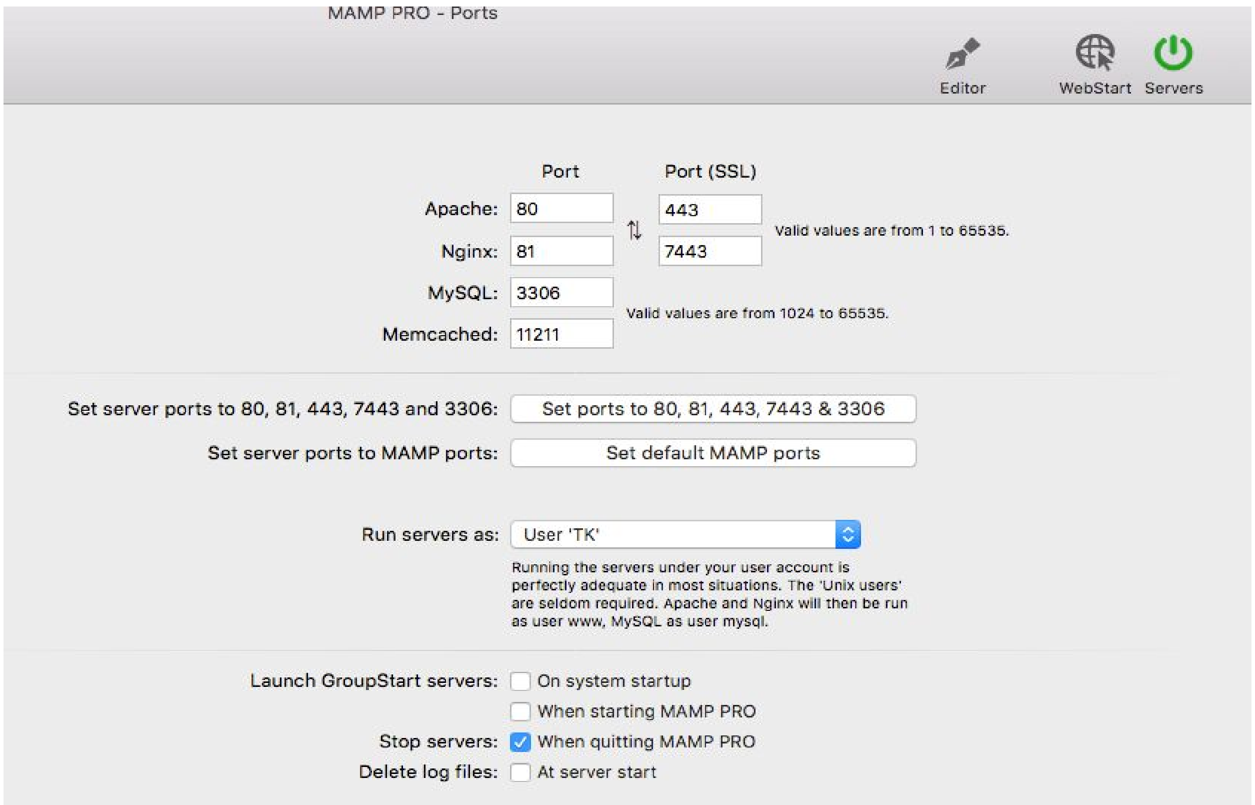


```
USE AnimalHospitalDatabase;

SELECT PetOwner.PetOwner_Name, PetOwner.PetOwner_DOB, Pet_SID, Pet_Name, Pet_Height, Pet_Weight, Pet_Color
FROM Pet
INNER JOIN PetOwner
ON PetOwner.PetOwner_Name = Pet.Pet_OwnerName
ORDER BY Pet_OwnerName ASC, Pet_Name ASC
```

PetOwner_Name	PetOwner_DOB	Pet_SID	Pet_Name	Pet_Height	Pet_Weight	Pet_Color
Lois Griffin	2013-12-12	3	Edda	190	185	Black
Lois Griffin	2013-12-12	2	Fido	18	90	Blue
Peter Griffin	2012-03-13	1	Abba	18	90	Yellow

```
   Time      Action                                                   Response             Duration / Fetch Time
1  22:32:02  USE AnimalHospitalDatabase                               0 row(s) affected    0.030 sec
2  22:32:02  SELECT PetOwner.PetOwner_Name, PetOwner.PetOwner_DOB,... 3 row(s) returned    0.266 sec / 0.00002...
```

MAMP Pro: Port Configurations



	Port		Port (SSL)
Apache:	80	⇅	443
Nginx:	81		7443
MySQL:	3306		
Memcached:	11211		

Valid values are from 1 to 65535.

Valid values are from 1024 to 65535.

Set server ports to 80, 81, 443, 7443 and 3306: Set ports to 80, 81, 443, 7443 & 3306

Set server ports to MAMP ports: Set default MAMP ports

Run servers as: User 'TK'

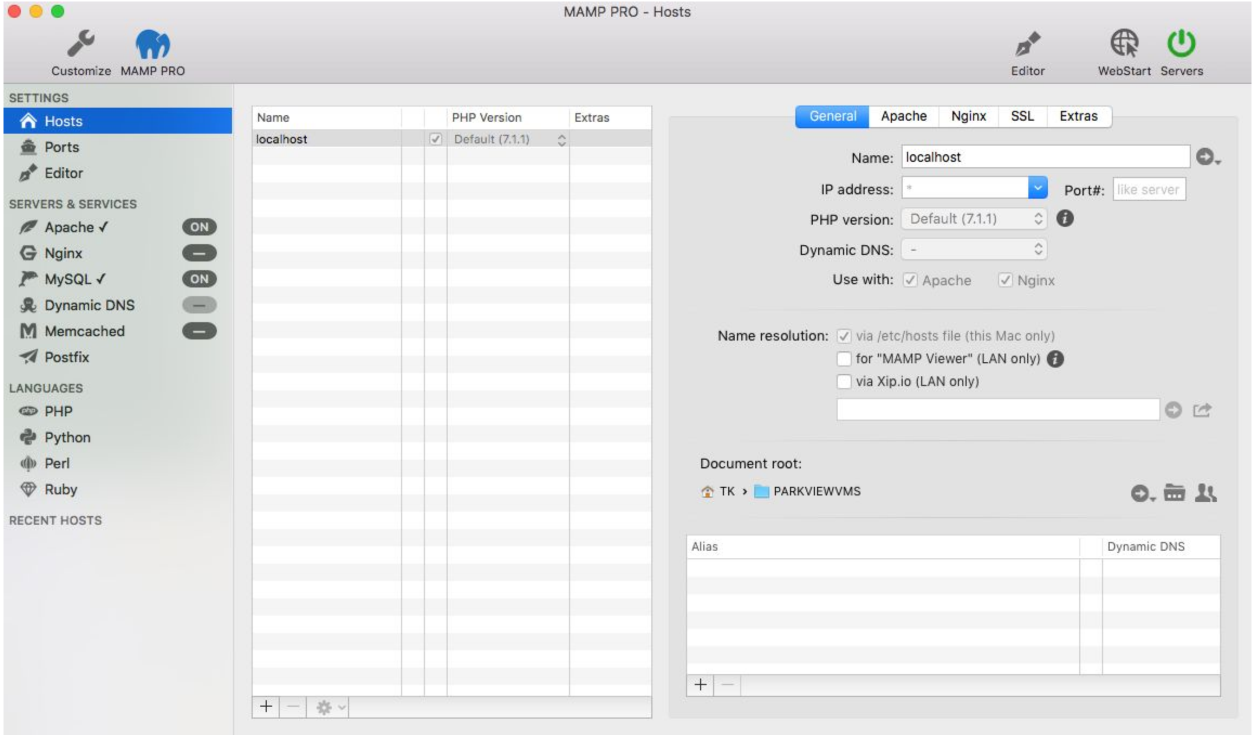
Running the servers under your user account is perfectly adequate in most situations. The 'Unix users' are seldom required. Apache and Nginx will then be run as user www, MySQL as user mysql.

Launch GroupStart servers: □ On system startup
□ When starting MAMP PRO

Stop servers: ☑ When quitting MAMP PRO

Delete log files: □ At server start

MAMP Pro: General Settings



Name		PHP Version	Extras
localhost	☑	Default (7.1.1)	

Name: localhost

IP address: *

Port#: like server

PHP version: Default (7.1.1)

Dynamic DNS: -

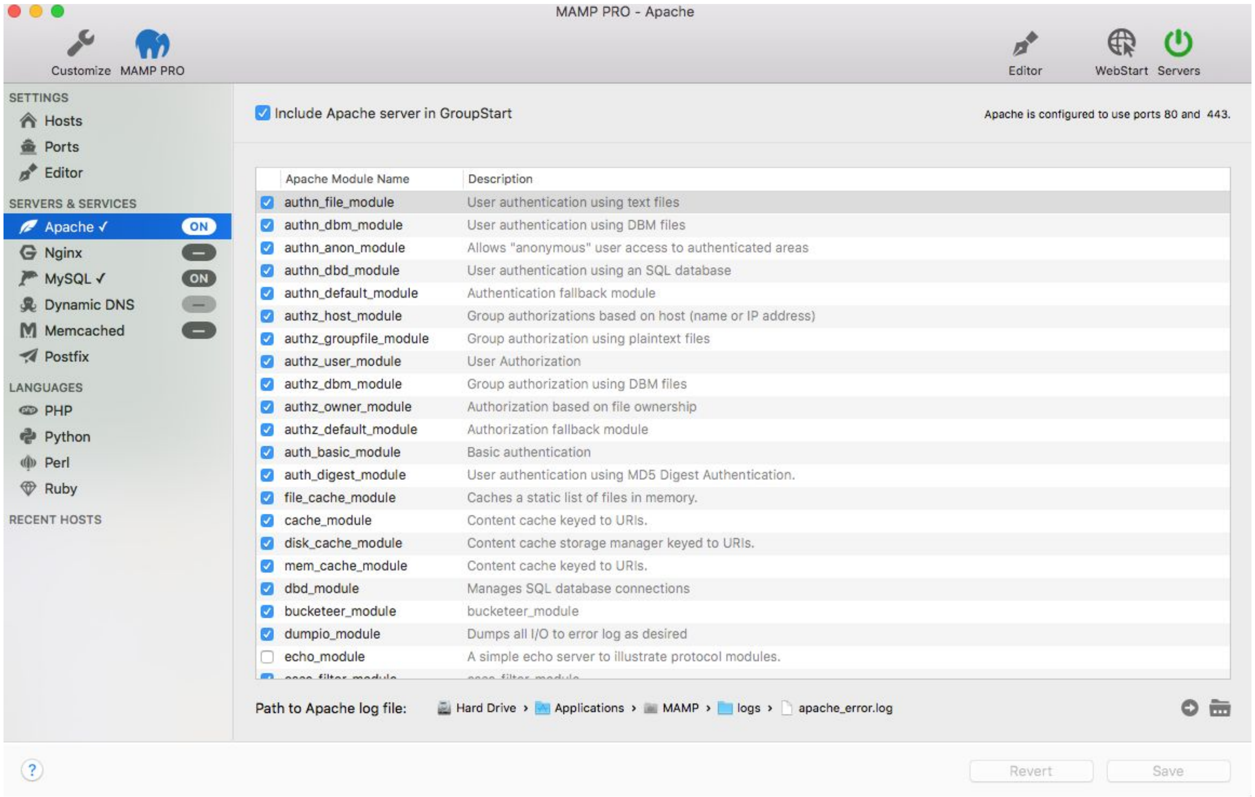
Use with: ☑ Apache ☑ Nginx

Name resolution: ☑ via /etc/hosts file (this Mac only)
□ for "MAMP Viewer" (LAN only)
□ via Xip.io (LAN only)

Document root:
TK › PARKVIEWVMS

Alias	Dynamic DNS

MAMP Pro: Apache configurations



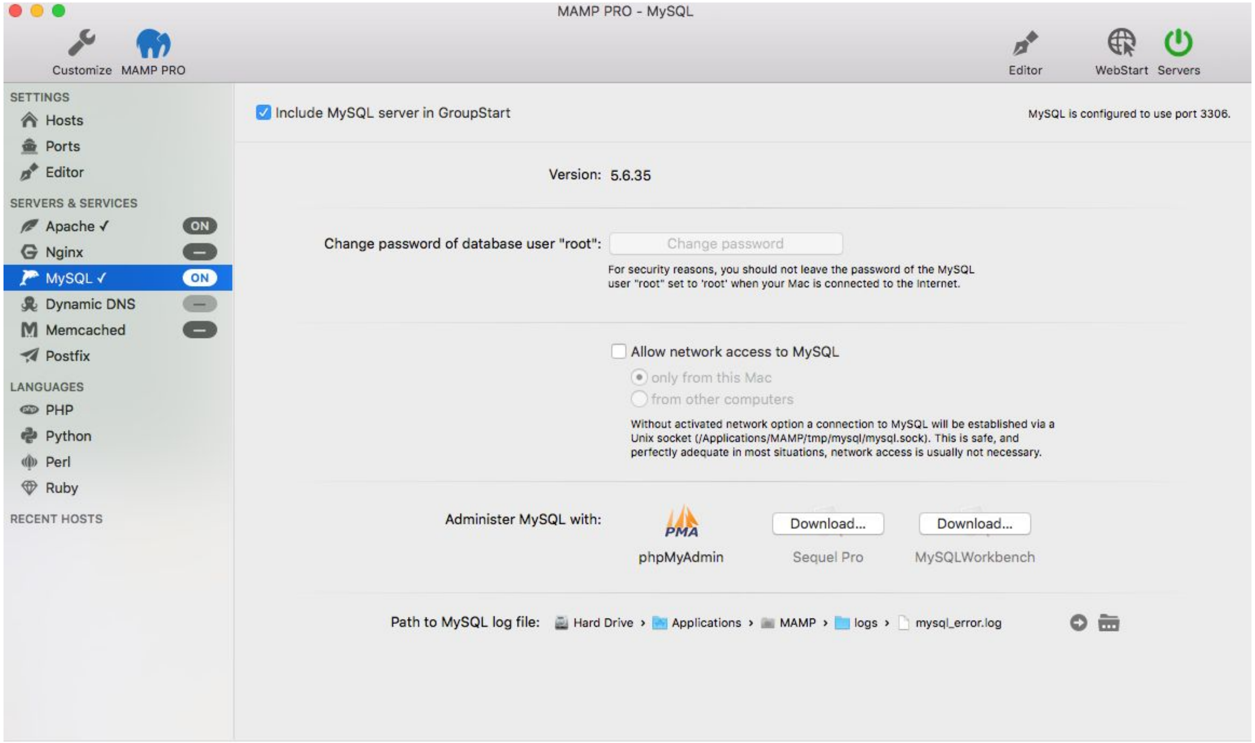
☑ Include Apache server in GroupStart

Apache is configured to use ports 80 and 443.

	Apache Module Name	Description
☑	authn_file_module	User authentication using text files
☑	authn_dbm_module	User authentication using DBM files
☑	authn_anon_module	Allows "anonymous" user access to authenticated areas
☑	authn_dbd_module	User authentication using an SQL database
☑	authn_default_module	Authentication fallback module
☑	authz_host_module	Group authorizations based on host (name or IP address)
☑	authz_groupfile_module	Group authorization using plaintext files
☑	authz_user_module	User Authorization
☑	authz_dbm_module	Group authorization using DBM files
☑	authz_owner_module	Authorization based on file ownership
☑	authz_default_module	Authorization fallback module
☑	auth_basic_module	Basic authentication
☑	auth_digest_module	User authentication using MD5 Digest Authentication.
☑	file_cache_module	Caches a static list of files in memory.
☑	cache_module	Content cache keyed to URIs.
☑	disk_cache_module	Content cache storage manager keyed to URIs.
☑	mem_cache_module	Content cache keyed to URIs.
☑	dbd_module	Manages SQL database connections
☑	bucketeer_module	bucketeer_module
☑	dumpio_module	Dumps all I/O to error log as desired
□	echo_module	A simple echo server to illustrate protocol modules.

Path to Apache log file: Hard Drive › Applications › MAMP › logs › apache_error.log

MAMP Pro: MySQL Server configurations



☑ Include MySQL server in GroupStart

MySQL is configured to use port 3306.

Version: 5.6.35

Change password of database user "root":

For security reasons, you should not leave the password of the MySQL user "root" set to 'root' when your Mac is connected to the Internet.

□ Allow network access to MySQL

● only from this Mac

○ from other computers

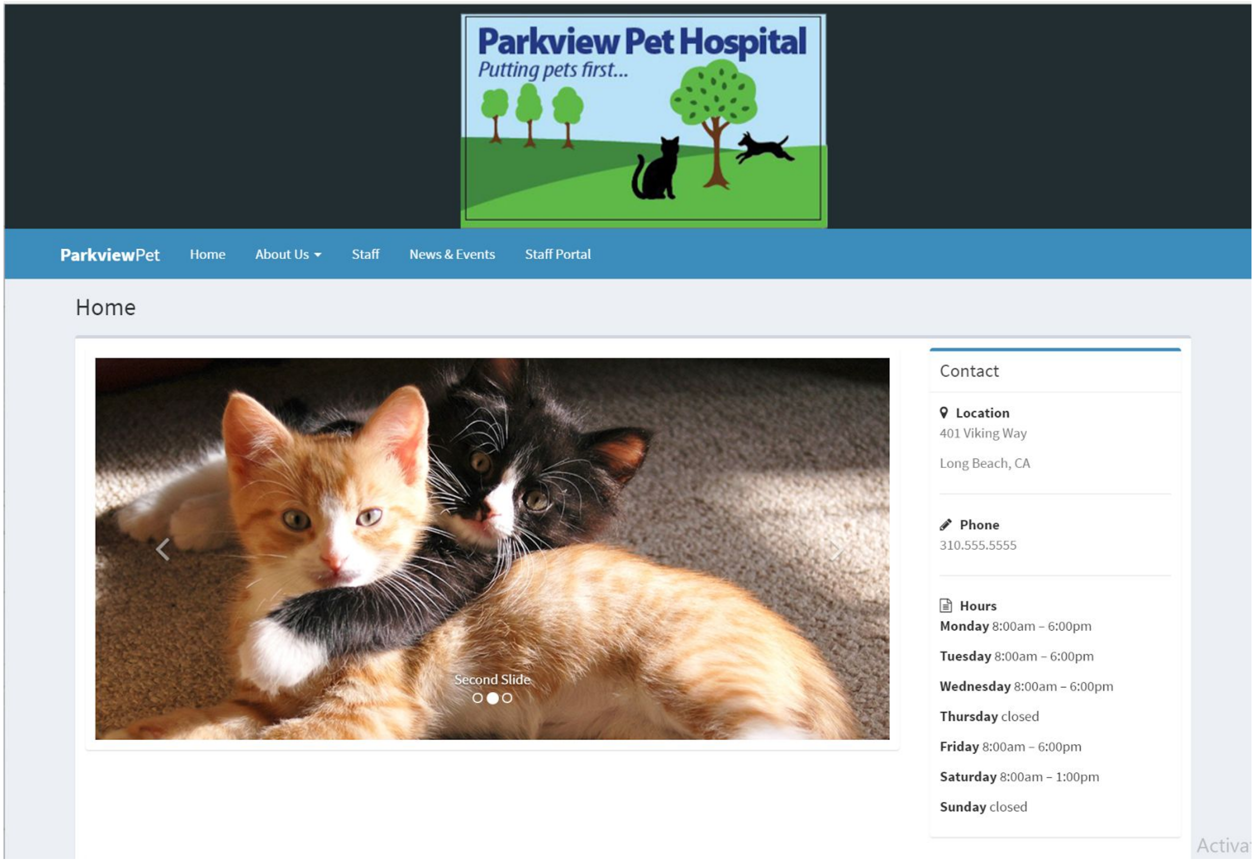
Without activated network option a connection to MySQL will be established via a Unix socket (/Applications/MAMP/tmp/mysql/mysql.sock). This is safe, and perfectly adequate in most situations, network access is usually not necessary.

Administer MySQL with: phpMyAdmin, Sequel Pro, MySQLWorkbench

Path to MySQL log file: Hard Drive › Applications › MAMP › logs › mysql_error.log

Full Stack Demonstration

A regular user would use the application primarily as a source of information. These users would be clients of Parkview Pet Hospital. After navigating to the website, public users have access to 6 pages: Home, Overview & Services, Mission, Gallery, Staff, as well as News & Events. Below are examples of some of these pages. Home, for example, displays the most basic information about the clinic. There are a number of images provided in a carousel, and the hospital contact information is listed to the right. Below these items is a quick description of Parkview Pet Hospital.



Home

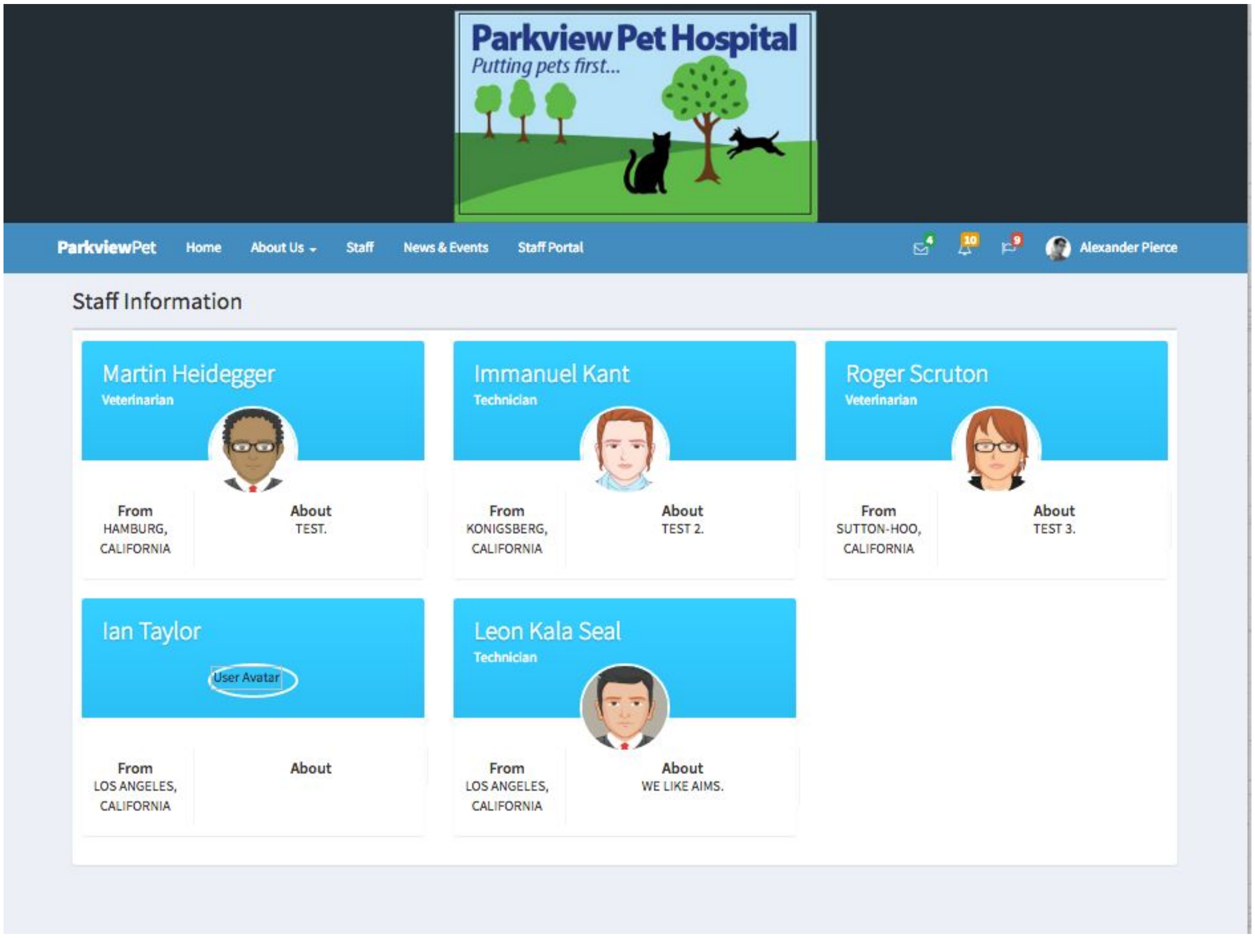
Contact

Location
401 Viking Way
Long Beach, CA

Phone
310.555.5555

Hours
Monday 8:00am – 6:00pm
Tuesday 8:00am – 6:00pm
Wednesday 8:00am – 6:00pm
Thursday closed
Friday 8:00am – 6:00pm
Saturday 8:00am – 1:00pm
Sunday closed

The tabs titled Overview, Mission, and Gallery are very similar in that they are purely static information. The Staff tab gives information about the staff members at Parkview to the clients, but this tab pulls directly from the database. Shown below is a screenshot of this page.



Staff Information

Martin Heidegger
Veterinarian
From
HAMBURG, CALIFORNIA
About
TEST.

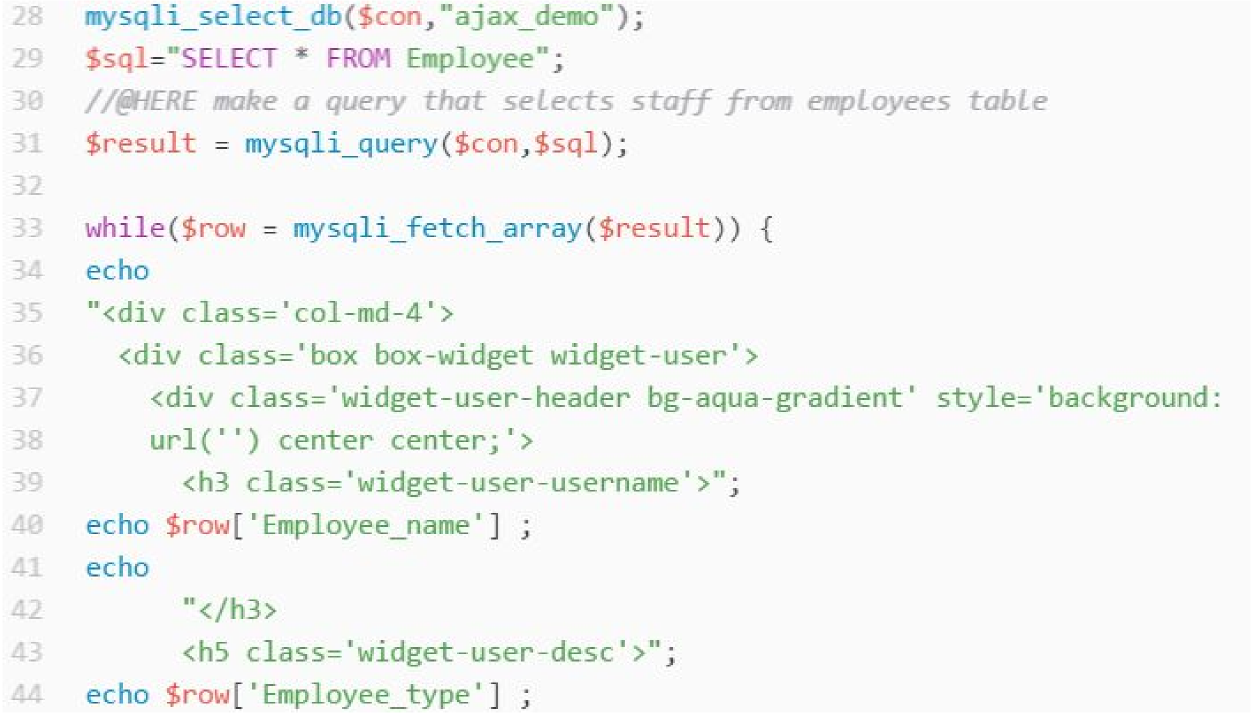
Immanuel Kant
Technician
From
KONIGSBERG, CALIFORNIA
About
TEST 2.

Roger Scruton
Veterinarian
From
SUTTON-HOO, CALIFORNIA
About
TEST 3.

Ian Taylor
From
LOS ANGELES, CALIFORNIA
About

Leon Kala Seal
Technician
From
LOS ANGELES, CALIFORNIA
About
WE LIKE AIMS.

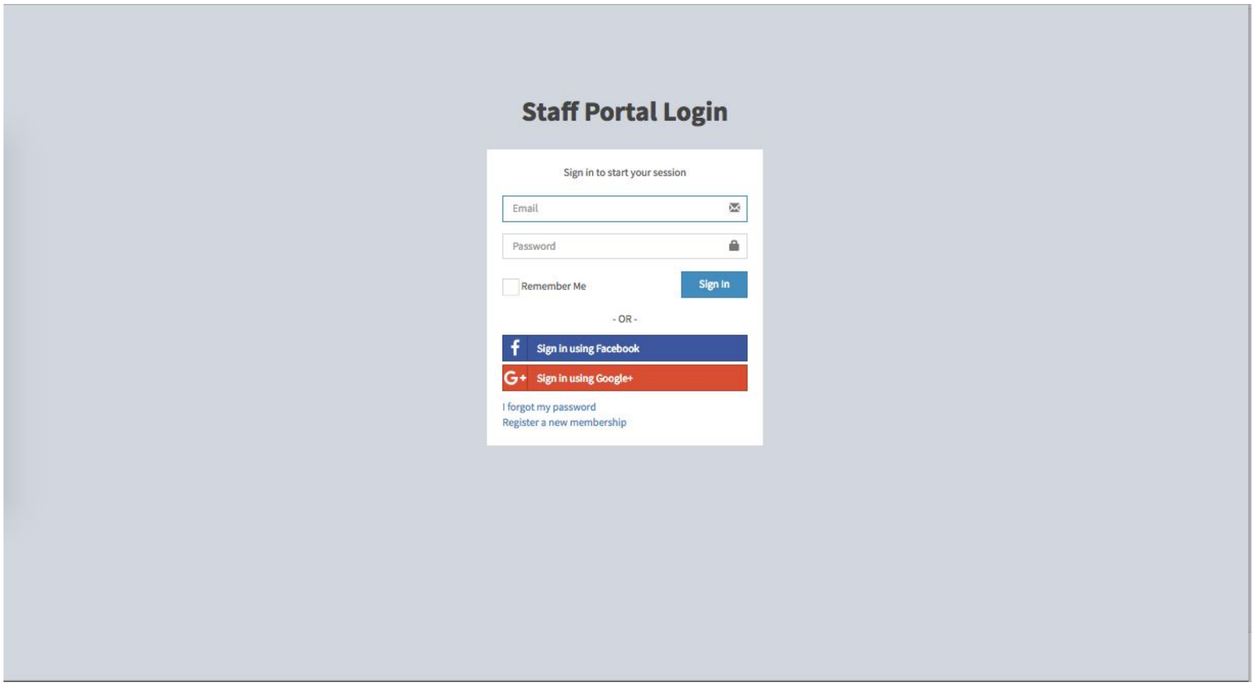
This is accomplished by running a query that selects all records from the Employee table and then loops through each record to echo each relevant field into HTML bootstrap cards. The image below shows the query being called as well as the echo events that display the field data as well as the surrounding HTML to make the employee information fit nicely into a card.



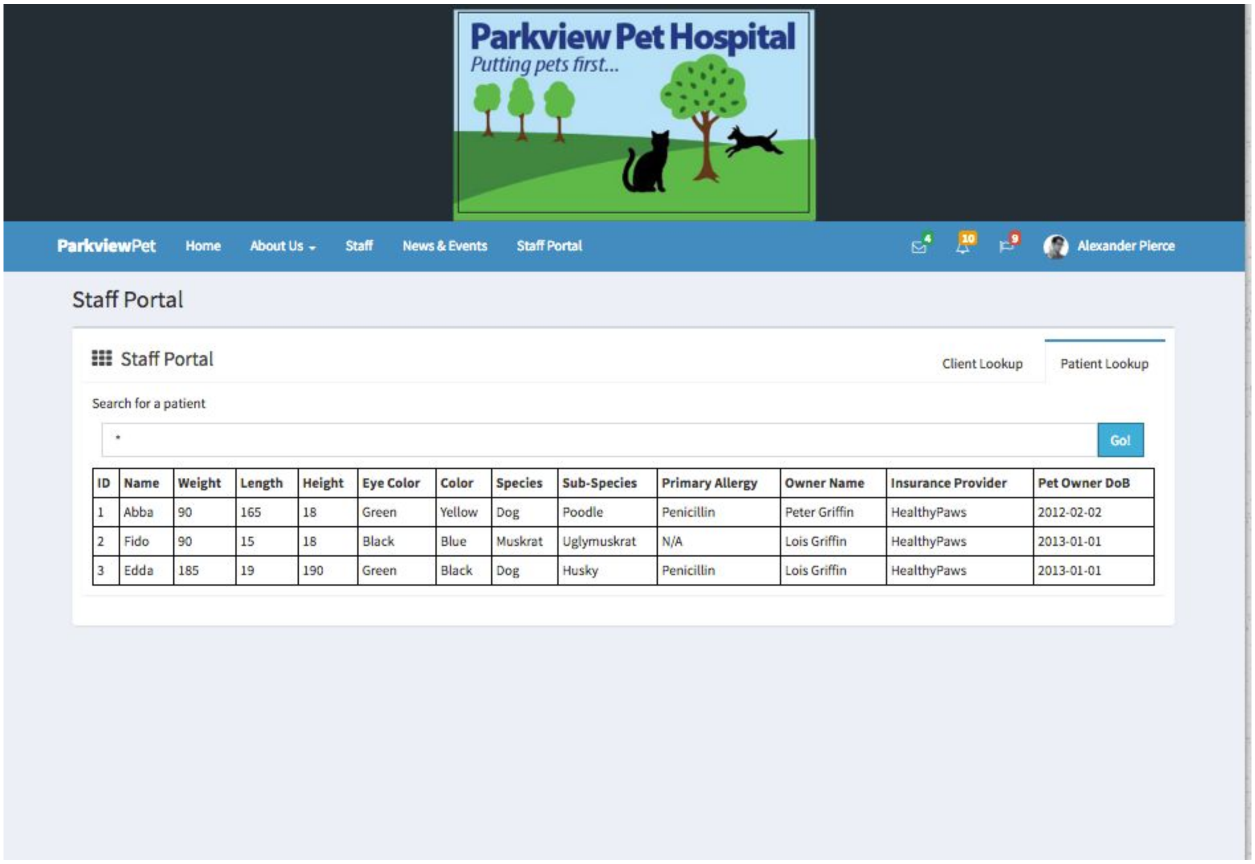
```
mysqli_select_db($con,"ajax_demo");
$sql="SELECT * FROM Employee";
//@HERE make a query that selects staff from employees table
$result = mysqli_query($con,$sql);

while($row = mysqli_fetch_array($result)) {
echo
"<div class='col-md-4'>
  <div class='box box-widget widget-user'>
    <div class='widget-user-header bg-aqua-gradient' style='background:
    url('') center center;'>
      <h3 class='widget-user-username'>";
echo $row['Employee_name'] ;
echo
      "</h3>
      <h5 class='widget-user-desc'>";
echo $row['Employee_type'] ;
```

A staff member would use the application primarily to interact with and update information about clients. Staff members can visit all of the public pages as a client would, but they can access an additional page titled “Staff Portal” after logging in. Before accessing this page, staff members must first log in. Below is the process for reaching this page.

First, a staff member clicks on the tab titled “Staff Portal”. They are brought to a confirmation screen where they are required to sign in. This page is listed below.

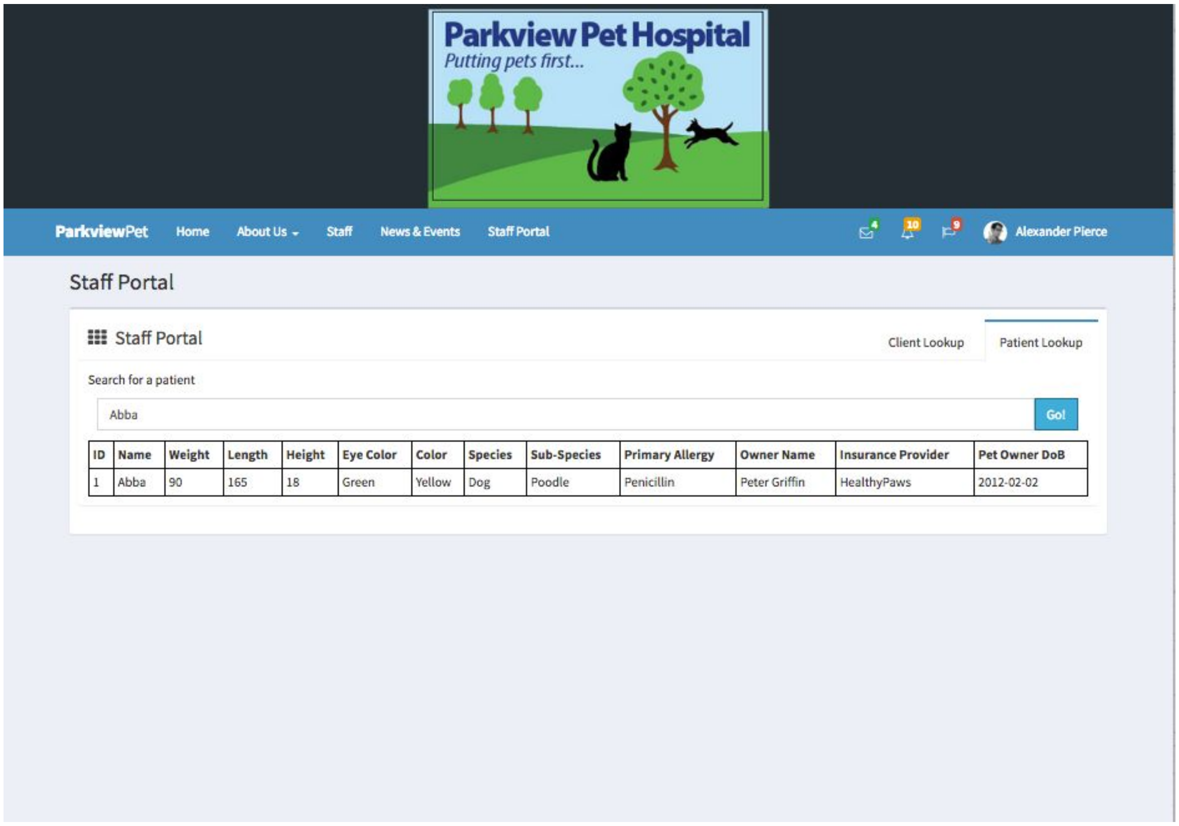


After a successful sign-in, staff members are brought to the staff portal. The prototype of this application provides two functions: patient lookup and client lookup. Clicking patient lookup brings users to an interface where they can search the database using a form to look-up patients. This page gives employees access to relevant patient information. Entering a ‘*’ into the search field and pressing “Go!” will pull all records from the database. This is shown below:



ID	Name	Weight	Length	Height	Eye Color	Color	Species	Sub-Species	Primary Allergy	Owner Name	Insurance Provider	Pet Owner DoB
1	Abba	90	165	18	Green	Yellow	Dog	Poodle	Penicillin	Peter Griffin	HealthyPaws	2012-02-02
2	Fido	90	15	18	Black	Blue	Muskrat	Uglymuskrat	N/A	Lois Griffin	HealthyPaws	2013-01-01
3	Edda	185	19	190	Green	Black	Dog	Husky	Penicillin	Lois Griffin	HealthyPaws	2013-01-01

Entering a specific name of a patient will yield all records matching the query. For example, entering “Abba” and pressing “Go!” will yield one record of a patient with that name. This is shown below:

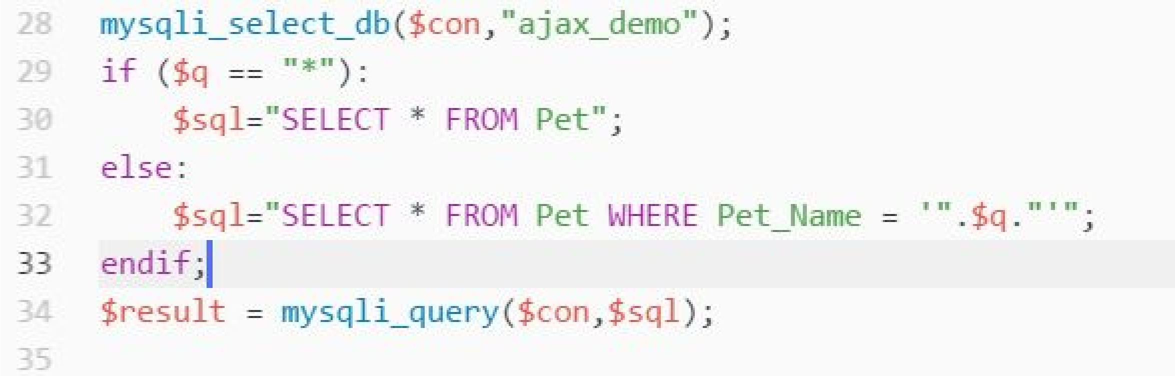


Search for a patient

Abba

ID	Name	Weight	Length	Height	Eye Color	Color	Species	Sub-Species	Primary Allergy	Owner Name	Insurance Provider	Pet Owner DoB
1	Abba	90	165	18	Green	Yellow	Dog	Poodle	Penicillin	Peter Griffin	HealthyPaws	2012-02-02

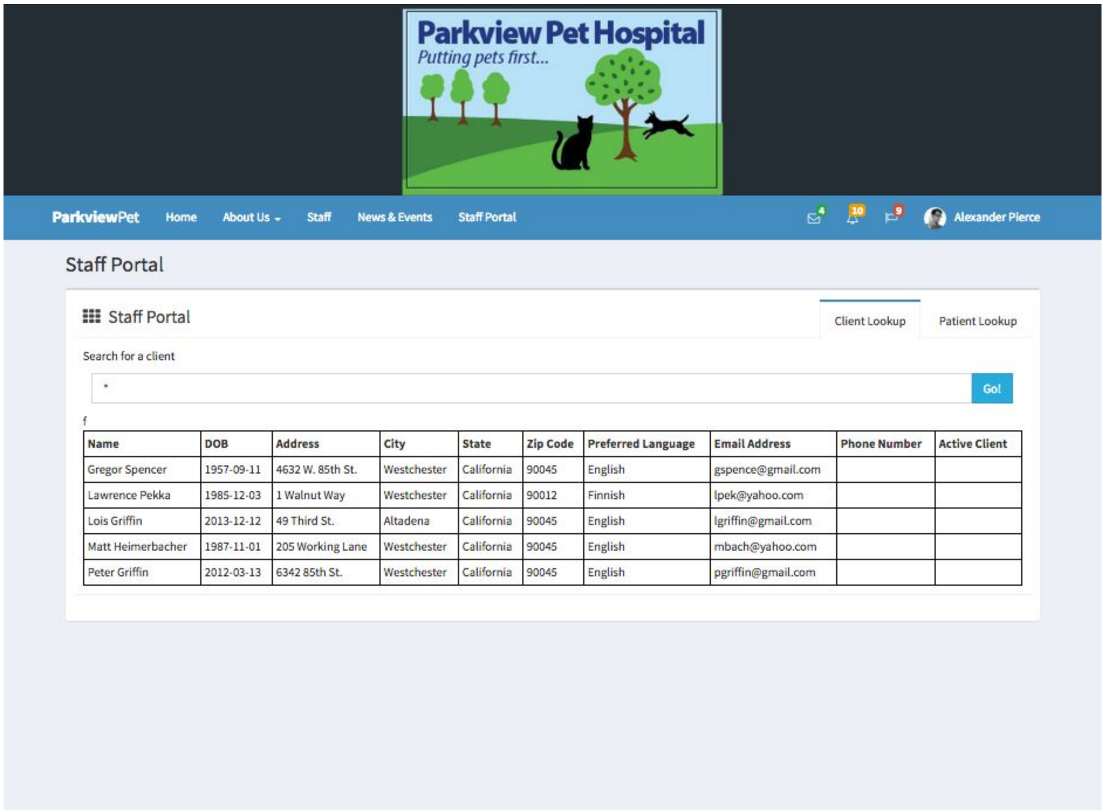
To accomplish this, we wrote an if statement that determines whether to run the query that selects all fields or selects only the fields where the Pet_Name is equal to $q, which is set to the value of the text box near the Go button.



```
mysqli_select_db($con,"ajax_demo");
if ($q == "*"):
    $sql="SELECT * FROM Pet";
else:
    $sql="SELECT * FROM Pet WHERE Pet_Name = '".$q."'";
endif;
$result = mysqli_query($con,$sql);

```

The results of the query are echoed to a table that is displayed as HTML.

By navigating to the second tab titled “Client Lookup”, staff members will be able to perform the same search functions on clients as opposed to patients. Entering a ‘*’ and pressing “Go!” will yield all records in the client table as shown below:



Staff Portal

Staff Portal

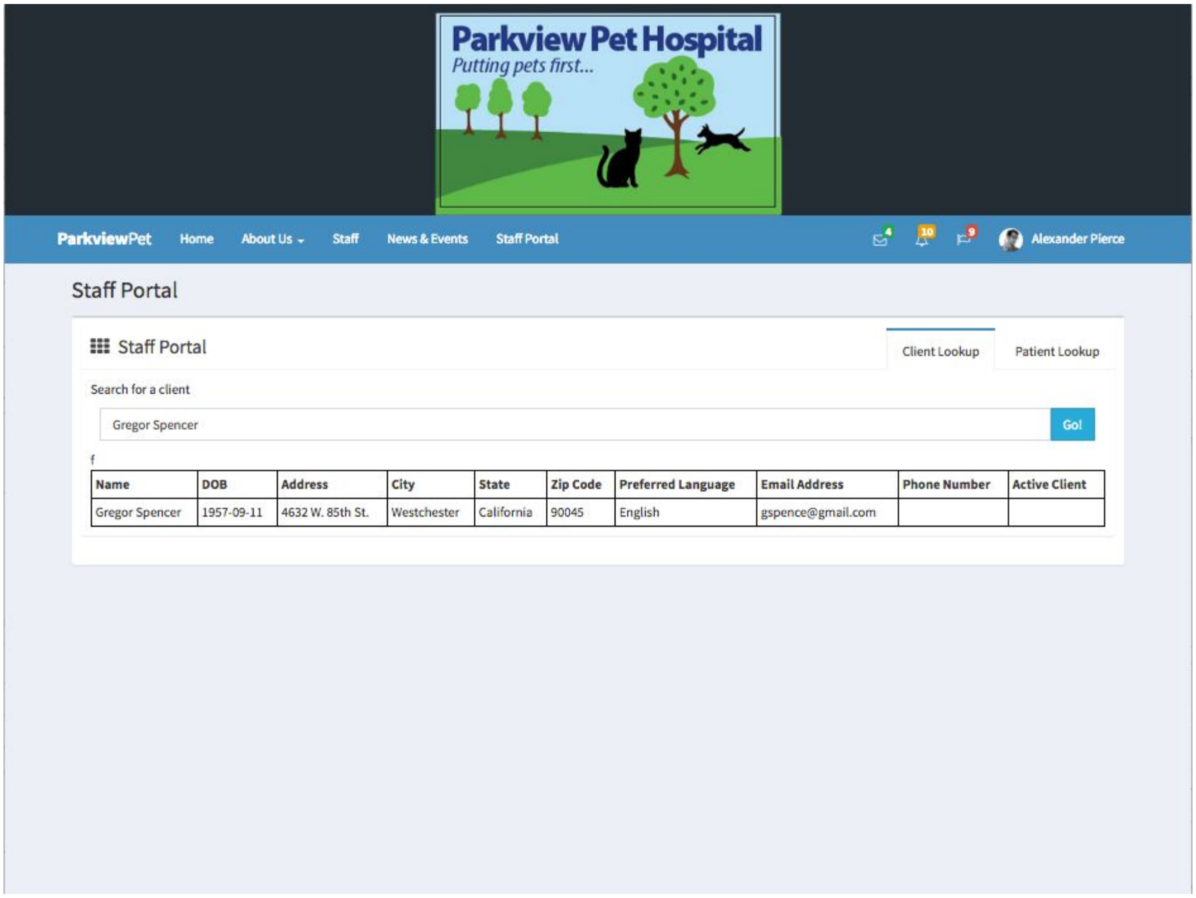
Search for a client

*

f

Name	DOB	Address	City	State	Zip Code	Preferred Language	Email Address	Phone Number	Active Client
Gregor Spencer	1957-09-11	4632 W. 85th St.	Westchester	California	90045	English	gspence@gmail.com		
Lawrence Pekka	1985-12-03	1 Walnut Way	Westchester	California	90012	Finnish	lpek@yahoo.com		
Lois Griffin	2013-12-12	49 Third St.	Altadena	California	90045	English	lgriffin@gmail.com		
Matt Heimerbacher	1987-11-01	205 Working Lane	Westchester	California	90045	English	mbach@yahoo.com		
Peter Griffin	2012-03-13	6342 85th St.	Westchester	California	90045	English	pgriffin@gmail.com		

Entering a specific name such as “Gregor Spencer” will yield one record as shown below.



Staff Portal

Staff Portal

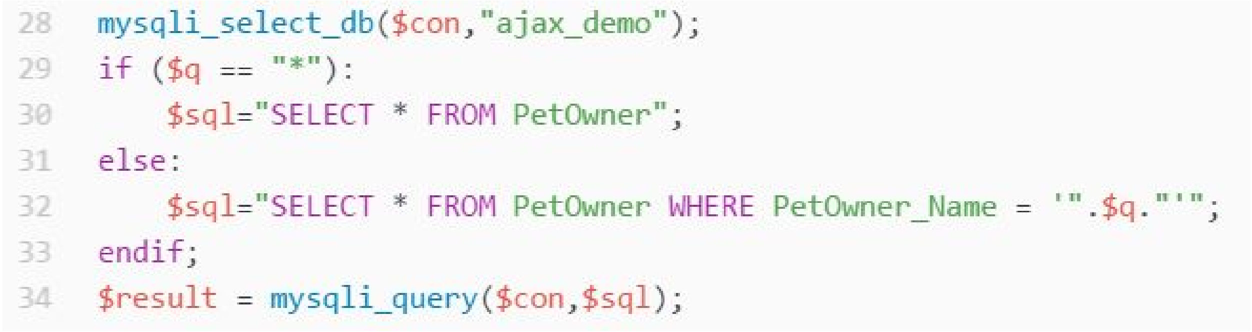
Search for a client

Gregor Spencer

f

Name	DOB	Address	City	State	Zip Code	Preferred Language	Email Address	Phone Number	Active Client
Gregor Spencer	1957-09-11	4632 W. 85th St.	Westchester	California	90045	English	gspence@gmail.com		

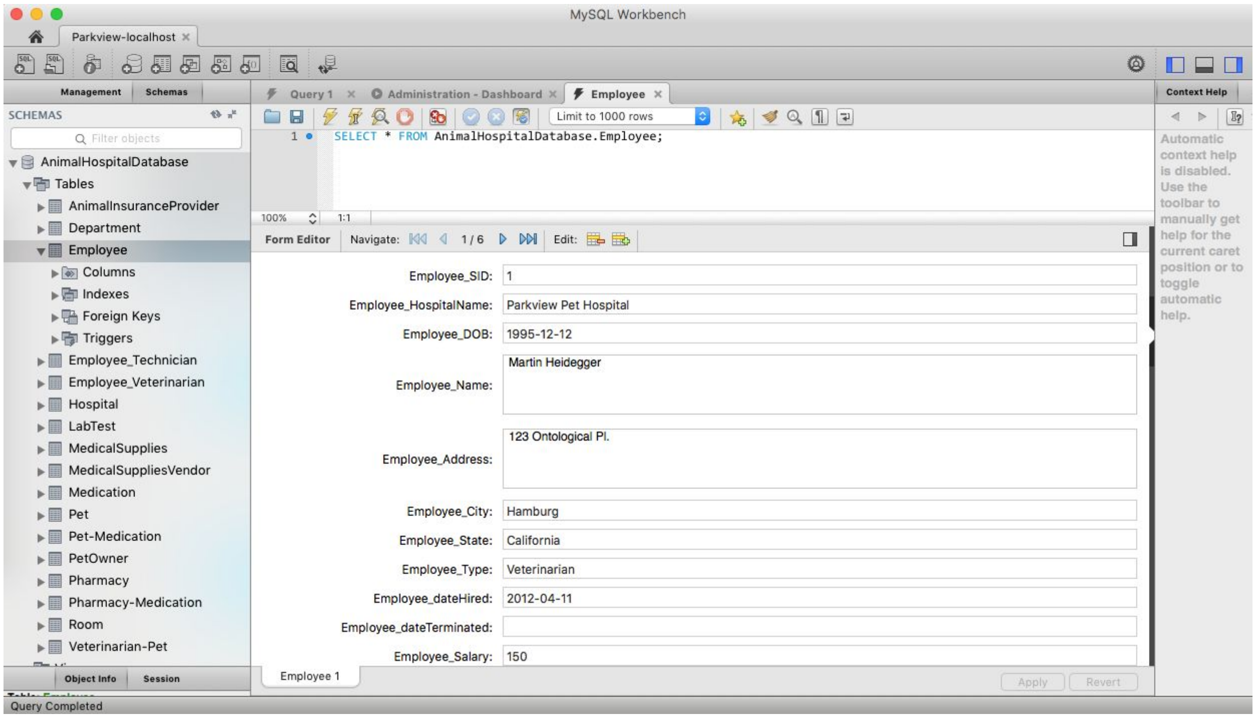
To accomplish this, we wrote an if statement that determines whether to run the query that selects all fields or selects only the fields where the Pet_Name is equal to $q, which is set to the value of the text box near the Go button.



```
mysqli_select_db($con,"ajax_demo");
if ($q == "*"):
    $sql="SELECT * FROM PetOwner";
else:
    $sql="SELECT * FROM PetOwner WHERE PetOwner_Name = '".$q."'";
endif;
$result = mysqli_query($con,$sql);
```

The results of the query are echoed to a table.

Finally, staff members will have the ability to use the built in form function in MySQL Workbench to enter new information into the database and update existing information. For instance, a staff member can add a new employee to the database, as shown below. Filling out the form and clicking submit will create a new record in the Employee table.



```
SELECT * FROM AnimalHospitalDatabase.Employee;
```

Employee_SID: 1

Employee_HospitalName: Parkview Pet Hospital

Employee_DOB: 1995-12-12

Employee_Name: Martin Heidegger

Employee_Address: 123 Ontological Pl.

Employee_City: Hamburg

Employee_State: California

Employee_Type: Veterinarian

Employee_dateHired: 2012-04-11

Employee_dateTerminated:

Employee_Salary: 150

Conclusion

What we accomplished

As a team, we successfully developed a functioning instance of the MySQL Server with a working database, using the InnoDB engine, along with several foreign keys, indexes, and queries utilized to show the interaction of data within the system. This database fulfilled many of the initial requests made by the organizational administration, as well as general characteristics that would allow it to be used generally for businesses that are not necessarily Parkview.

Next, we successfully designed and implemented a local deployment of our website, in a proof-of-concept form, with the Admin LTE platform. We also successfully secured the staff portal login that allows for direct interaction with enterprise data.

After developing both platforms independent of one another, we managed to install and run an instance of the MAMP Pro software, regenerating the previous MySQL Server instance into the integrated MAMP stack version, and integrating our HTML/CSS webpages with the database using the built application sockets and self-written PHP code to facilitate data transfer.

Finally, we successfully integrated out MAMP Pro with our organizational GitHub repository, allowing for remote changes and management of pages, and SQL/PHP scripts without needing direct access to the localhost machine.

Issues Faced

We both suffered from a lack of prior, deep experience with the general (X)AMP-stack architectures for web-hosting. Configuration of our MAMP (Mac OSX, Apache Server, MySQL, PHP) stack was incredibly complex; it presented us with a lot of trial-and-error and database table corruption. In this same vein, we both had low knowledge of programming PHP scripts. MySQL, specifically the InnoDB engine, is error-prone and small mistakes in the creation of constraints, indexes, etc., can require a complete re-deployment of the database. DML is also less easy to set up than Microsoft products (Access, SQL Server). There were limitations of the local machine (Taylor’s 2011 MacBook Pro), and before integration of programming efforts through Github Desktop, difficulty in version control and dynamically testing deployments of new code and configurations.

Further Work To Be Done

Firstly, a transition of (X)AMP-stack architecture to a preferably locally-based server, with dedicated software and hardware. We both have vague experience with this, although we simply ran out of time to implement and test this system. Next, development of multiple databases for varying purposes. An operational database should remain independent of a database utilized for much of the daily functioning of the website, such as logins, passwords, web-data storage, etc. After that, integration of Wordpress Engine for better site/page administration. We experienced some strange issues in PHP deployment of this software, and had to move on to more critical requirements. Finally, Analytics/BI services would be ideal, once there was an adequate amount of data stored in the system to warrant it. Ideally we could deploy this on a dedicated ‘application server’, preferably with Tableau, PyCharm/Tensorflow, Excel (with integrated Solver software) and other analytics software.