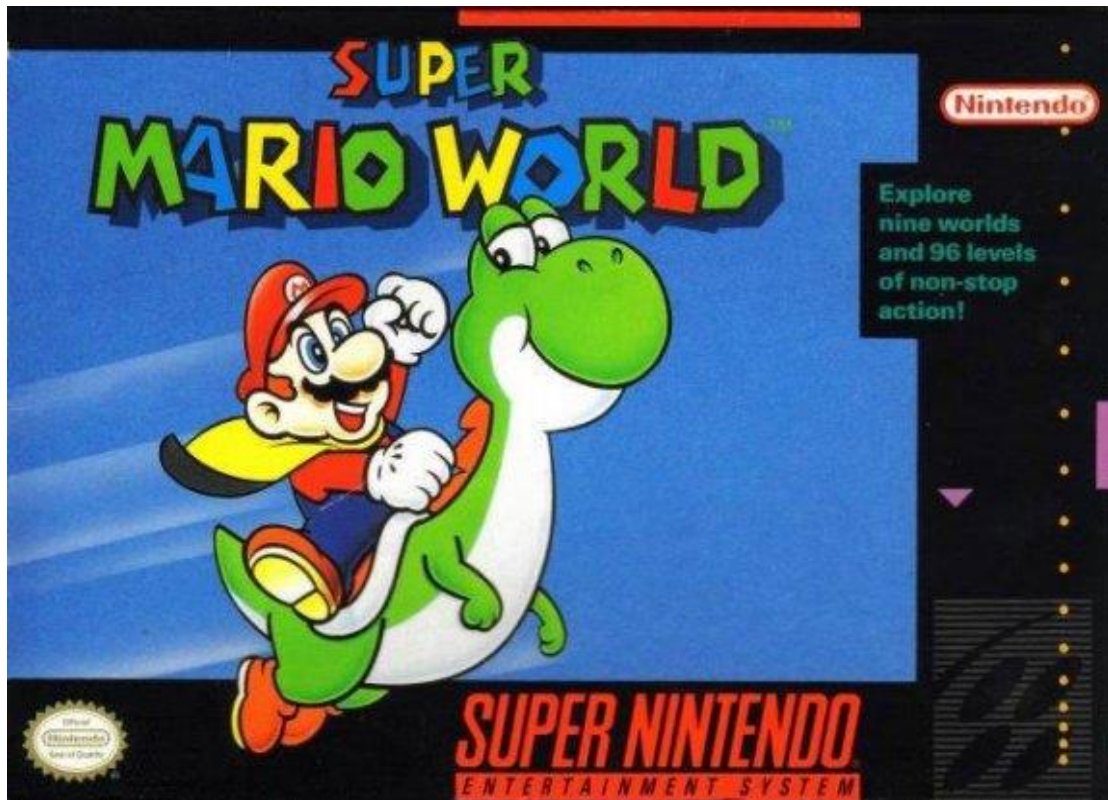


Running Head: OLD SCHOOL GAMES



Old School Games
Zailyn Tamayo
National University

Table of Contents

Executive Summary.....	3
Introduction	4
Project Description.....	4
Technology Overview	4
VirtualBox.....	4
Ruby on Rails	4
PostgreSQL	5
Heroku.....	5
Implementation.....	6
Deploying the Application	6
Database.....	7
Demonstration	8
Home Page.....	8
Products	8
Place Order	8
Challenges and Workarounds	9
Conclusion	10
References.....	11
Appendix I: Functional Specifications.....	12
Appendix II: Screenshots	13

Executive Summary

The idea for this application was inspired by an example application in a book called Pragmatic Agile Web Development with Rails. The example was a depot application that sold textbooks. Initially, we were going to make an application that would sell used baby toys, but we were not too interested in that idea. I am an avid gamer, and I had the idea to make a site that would sell used video games. Video games are very expensive and I think there is a large market for used games; used games are more affordable, and oftentimes they are in almost new condition.

Used games have to be functioning in order to be sold, meaning that the games still run/work fine. A lot of times, I see used games that are in good condition and still in their cases. Sometimes, games will have damaged cases or no cases at all. This could deter people from buying used games. However, a person could buy empty games cases and either make their own artwork for the game or find artwork online. This could be fun, creative and a way to personalize your games.

With our Old School Games site, a person can find used games and purchase them quickly and easily without having to hunt for them at various stores. In the future, I would like to either build off of this site, or create a new web application where I could sell games or textbooks that I no longer want or need. It would be interesting to explore how to get my site onto Amazon's list of third-party sellers, or how to link the site to Ebay, if that can be done.

For our web application, we chose to use Ruby on Rails as our programming language. No one on our team had ever used Rails, but we were willing to experiment with something new. I had previous experience with Java, C++ and Python, as well as others, but I had only heard about Ruby on Rails from various sources. My role in the project was to deploy the app into the cloud, but I also had to dig into the code in order to have a full understanding of the project. For our cloud service provider, we chose to launch the application using Heroku; it was not our first choice, but there was enough information on the Internet that allowed us to learn how to launch the application.

There were two major challenges in this project, which I personally experienced. One was trying to change the database, and the other was figuring out how to deploy the application on the cloud. This paper will document my role in the Old School Games project, mainly focusing on deployment of the application and challenges and workarounds.

Introduction

In the following two sections, I will provide a description of our project and also give a brief overview of the technology used for the project.

Project Description

The goal of our project was to create a web application and launch it on the cloud. The name of the project is Old School Games; it is a site where people can purchase used video games. The Home page of the site has a listing of the games for sale with a description and the ability to add the game to cart. Once a game is added to cart, you can view what games have been added on the left hand side by refreshing the page. From this section, you are able to checkout. Clicking the checkout button will send you to a new page where you can input your information and confirm purchase of the order. It is a simple web application due to our lack of experience with Rails and limited time to create the application.

Technology Overview

Appendix I provides a detailed and complete list of the functional specifications for this project. I will provide a brief overview of the software used to create our web application.

VirtualBox

All work for this project was done in a virtual environment; the virtualization software used for this was Oracle's VirtualBox, running a virtual machine instance of Ubuntu OS. VirtualBox is a cross-platform virtualization application, meaning that it gives you the ability to run multiple operating systems simultaneously. Virtualization is useful because you can experiment on the virtual machine without damaging anything on your native operating system. Also, you can create snapshots of your virtual machine that you can revert back to if you make a mistake.

Ruby on Rails

The application was written using Ruby on Rails. Ruby on Rails is an open source web application framework written in Ruby; it is often referred to as just Rails. Rails uses the principles of

“Convention over Configuration” and “Don't Repeat Yourself”. “Convention over Configuration” means a developer only needs to specify unconventional aspects of the application and “Don't Repeat Yourself” means that information is located in a single, unambiguous place (Ruby on Rails). Rails uses the model-view-controller (MVC) pattern, which we will discuss later in the Implementation section. Rails includes WEBrick, which is a Ruby web server and RubyGems, a package manager. Applications built with Rails can use MySQL, PostgreSQL and SQLite3 for their database.

PostgreSQL

We chose PostgreSQL for our project; initially we wrote the application with SQLite3 because it was the default when creating a new Rails app, but we ran into problems when it came to deployment. PostgreSQL is an open source database management system that runs on all major operating systems. I was glad that we ended up with PostgreSQL after reading that is a top choice for cloud implementations. “Another important point to consider for any cloud implementation of a database tier is the security of the database. Accordingly, PostgreSQL is considered by many to be the most secure out-of-the-box configuration available for a database” (Rittinghouse & Ransome, 2013, p. 83).

Heroku

Heroku was our choice for Cloud Service Provider, entirely because of its ease of use compared to other CSPs we were researching. Heroku is a cloud platform-as-a-service (PaaS); it allows developers to focus on building applications without having to worry about the infrastructure. Another great feature about Heroku is that it has PostgreSQL built-in. According to their website, “Heroku Postgres provides a SQL database-as-a-service that lets you focus on building your application instead of messing around with database management” (Heroku).

Implementation

For the implementation documentation, I can only write about my personal experience and part in the project. My role was to deploy the application on to the cloud. I received the application already completed and will talk about what I saw in the application and how I managed to deploy it on to Heroku. A big portion of my work was done in the “Challenges and Workarounds” section of this paper. This section will cover the point of the project where I received the application and then the initial deployment on to Heroku. The first deployment on to Heroku was done before the presentation, and it had formatting issues. We needed to get the application deployed and that is what I did. After the presentation, a teammate managed to deploy the application on Heroku with the correct formatting. I will also try to deploy on Heroku again and maintain formatting.

The first thing I did when I received the file containing the application was to look through the folders. I looked at the javascript file as well as looked for the model-view-controller folders. Figures 1 through 5 show the contents of the application folder. The next thing I did was to open up my Terminal and attempt to launch the application locally. My first command was **sudo apt-get update** to make sure that Ubuntu was up to date. After this I used the **cd** command to change to my application directory. Ubuntu already contains Ruby on Rails, but I needed to install Bundler; Bundler is a program that lets you work on projects by tracking and installing the exact gems and versions that are needed for Ruby. Figure 6 shows the Bundler install.

Deploying the Application

After the bundle command, I used the command **rails s (or “server”)** to launch WEBrick, the Ruby server. Figures 7 and 8 show the server running. I opened my Firefox browser and went to “localhost:3000” to view our application. Now that I knew I could launch locally, the next step was to create a Heroku account and deploy via git. The steps for deployment are as follows:

- **sudo gem install heroku** – Install heroku.
- **git init** – Creates a git repository out of the project. Must be done in the project folder.
- **heroku create** – Creates a Heroku app. You will get a long app name.
- **heroku apps:rename (place new name here)** – Change the name of your project. You can log in to Heroku and view your app now. Figure 9 shows the application on the Heroku site.
- **git add -A** – Add all files in the project to a commit.

- **git commit -m 'Initial commit'** – Commit the files.
- **git push heroku master** – Push the code to Heroku.
- **Heroku rake db:migrate** – Migrate the database on Heroku.
- **heroku rake db:seed** – Seed the database on Heroku.
- **heroku restart** – Restarts the Heroku server.
- **heroku open** – Opens the running application. Figure 10 shows the first deployment of our application.

These are the steps I used to launch our initial deployment of our application; this application had formatting issues. Since then, a teammate was able to launch the website with formatting.

Database

Figure 11 shows the database model. Since I did not create the application, I am not familiar with the database. However, I know that Heroku has PostgreSQL built-in and I know that I created and migrated the database when I deployed the application. I was able to view the database schema from the Terminal by issuing the following commands:

- **cd (project directory)** – Change the directory to the project.
- **heroku pg:psql** – Open the PostgreSQL console.
- **\d** – View all tables.
 - **\d (table name)** – View specific table.

Figure 12 shows this process and the database schema for the Heroku application.

Demonstration

In this section, I will be describing each page in our application. The web application has three pages that I will be talking about.

Home Page

The home page lists the games available for sale with a picture and description of the games. Below the description is the price of the game along with a button that lets you add the product to cart. Figure 13 shows the home page of the site when you first open the page. Once you have added products to your cart, if you refresh the page, you will see on the left hand side a list of the products you have added to your cart. Underneath the list of items you will see two buttons: one to proceed to checkout and one to clear your cart and start over. Figure 14 shows this.

Products

There is a Products page, but I think it is only for the developers to use. On this page, there is a list of the products offered for sale; each product has three options next to them. “Show” will take you to a different page with the product description on it. From the Show page, you can edit the product or go back to the products page. “Edit” will take you to a page where you can edit the product; the fields you can edit are title, description, image url and price. “Destroy” I assume will delete the product from the site; I did not try for fear of messing up the web page. Under the products, there is a button titled “New product”, which takes you to a page where you can create a new product. Figures 15 and 16 show these features. The fact that you can create, edit and delete products from this page leads me to believe that it is a page for developers only.

Place Order

When you click “Checkout” from the home page, you are taken to a page where the buyer can input their information. The fields to input are name, address, email and pay type. At the end of the fields there is the button to “Place Order”. You cannot click the place order button without filling out all the fields; an error message will occur if you do not fill out all the boxes. Figures 17 and 18 show the error messages. Once you are done, you are taken back to the home page but there is a message that

says, “Thank you for your order”. Figure 19 shows this message.

Challenges and Workarounds

There were several issues and workarounds that came up with the development of this project. The first and most important issue was that no one on the team had any experience with Rails nor with deploying an application to the cloud. This made things more difficult and slow moving for us. The other challenges, I will present in bullet form. This is a list of only my personal challenges and workarounds.

- **Challenge:** Once the application was written and created, we had problems figuring out how to deploy on to Amazon Web Services (EC2), which was our first choice of Cloud Service Provider. **Workaround:** We researched on how to deploy applications to the cloud. Due to the extensive amount of documentation, we chose to deploy the application using Heroku.
- **Challenge:** Our application was written using Sqlite3 because it was the default for Rails. However, Heroku does not allow Sqlite3 and suggested converting to PostgreSQL. **Workaround:** I attempted to convert the database, but did not succeed. In the end, I believe a teammate managed to convert the database or the application was re-written in PostgreSQL.
- **Challenge:** When I tried to create the Heroku application using the “heroku create” command, I would get an error saying, “ExecJS::ProgramError: Unexpected token: operator (=)”. Figure 20 shows the error message. **Workaround:** I assumed this meant there was an error in the code but I did not know exactly where to look. I found the error in the application.js file. It was a syntax error in which the programmer forgot to place the comment lines (“//”).
- **Challenge:** Another error I would get when trying to deploy was, “Sprockets::FileNotFound: couldn't find file jquery.ui.effect-blind”. Figure 21 shows this error message. **Workaround:** In the application.js file, I could see that jquery.ui.effect-blind was present, as seen in Figure 22, so I did not know what was wrong. After doing research, I discovered that jquery.ui.effect-blind was replaced by jquery-ui/effect-blind. I changed the application.js file, as seen in Figure 23 and ran the “bundle” command to update.
- **Challenge:** When attempting to migrate the database using the “heroku run rake db:migrate” command, I would get an error stating that the “role does not exist” or that the “role does not have access” because there was no password authentication. **Workaround:** The solution was to

go into PostgreSQL from the Terminal and create roles and alter the password. Figure 24 shows the roles that I created using the “CREATE ROLE” command. The database.yml file in the config folder of the application also needed to be updated to mirror these changes.

Conclusion

This project was very challenging for me and for my teammates. However, at least speaking for myself, I learned a lot from this project. Before starting this project, I did not know about virtualization, Rails or Cloud Service Providers. Now that this class is complete, I feel confident about these subjects.

I think that our web application has potential; it is a good idea to have a site that will sell used games. There is a market for used games and it might be nice to go to a site dedicated to it. Though I doubt neither of us will maintain this site, I am glad that I now know how to create and deploy a web application.

References

- Hawkins, A. (2010, October 8). Zero-to-Sixty: Creating and Deploying a Rails App in Under an Hour. Retrieved August 22, 2014, from <http://code.tutsplus.com/tutorials/zero-to-sixty-creating-and-deploying-a-rails-app-in-under-an-hour--net-8252>.
- Hawthorne, A. (2012, September 27). Deploying a Rails Application. Retrieved August 22, 2014, from <http://www.sitepoint.com/deploying-a-rails-application>.
- Heroku. (n.d.). Retrieved August 30, 2014, from <https://www.heroku.com/>.
- Heroku Wikipedia. (n.d.). Retrieved August 30, 2014, from <http://en.wikipedia.org/wiki/Heroku>.
- Ruby on Rails Wikipedia. (n.d.). Retrieved August 31, 2014, from http://en.wikipedia.org/wiki/Ruby_on_Rails.
- Rittinghouse, J.W., & Ransome, J.F. (2010). *Cloud Computing: Implementation, Management, and Security*. Boca Raton: Taylor and Francis Group, LLC.
- Ruby, S., Thomas, D., & Heinemeier Hansson, D. (2013). *Agile Web Development with Rails 4*. Dallas: Pragmatic Programmers, LLC.

Appendix I: Functional Specifications

Host OS Specifications:

- Toshiba Satellite laptop
- 3GB RAM
- 250 GB hard drive
- Ubuntu 14.04 (Trusty Tahr)
- Mozilla Firefox version 31.0

Guest OS Specifications (Virtual Machine):

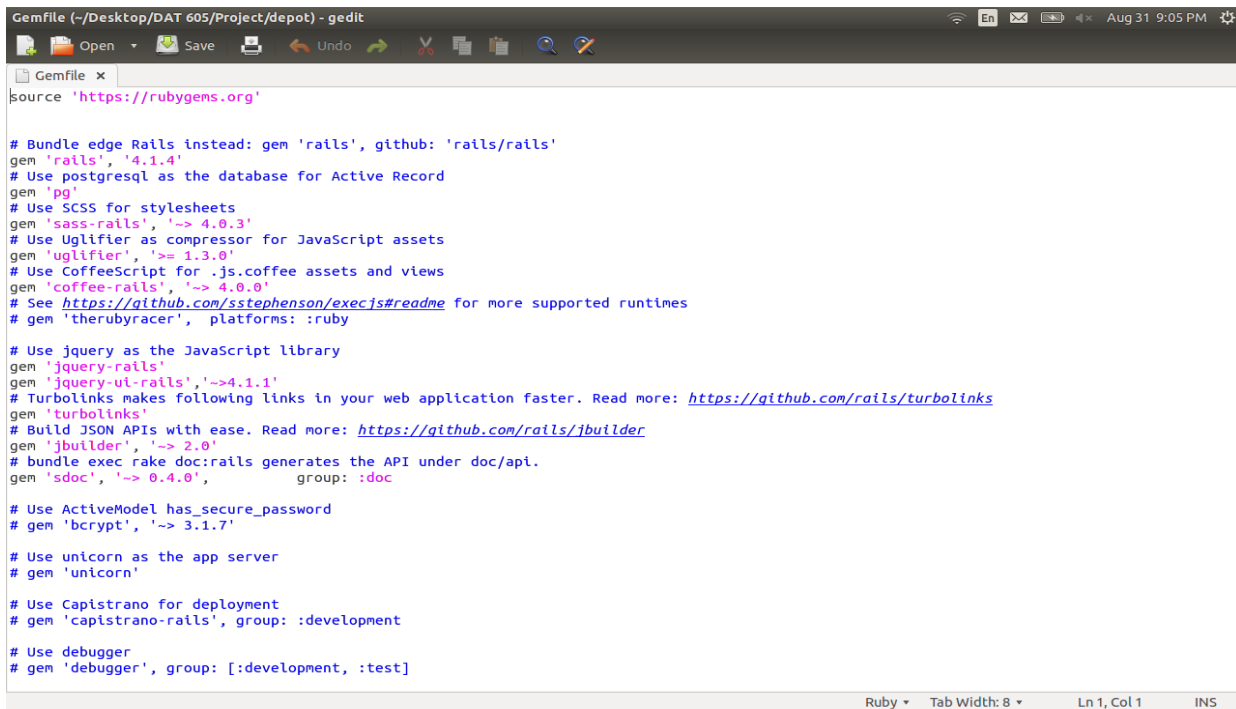
- VirtualBox version 4.3.12
- 1 GB RAM
- 10 GB hard drive
- Ubuntu 12.04 (Precise Pangolin)
- Mozilla Firefox version 31.0

Web Application Specifications:

- Ruby version 2.0.0
- Rails version 4.1.4
- PostgreSQL

Appendix II: Screenshots

Figure 1: Application.js file



The screenshot shows a Gedit editor window titled "Gemfile (~/.Desktop/DAT 605/Project/depot) - gedit". The window displays the content of a Gemfile, which is a Ruby file used to manage the dependencies of a Rails application. The file includes comments and gem specifications for various libraries and frameworks. The status bar at the bottom indicates the file is a Ruby script, with a tab width of 8, and the cursor is at line 1, column 1.

```
source 'https://rubygems.org'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '4.1.4'
# Use postgresql as the database for Active Record
gem 'pg'
# Use SCSS for stylesheets
gem 'sass-rails', '~> 4.0.3'
# Use Uglifier as compressor for JavaScript assets
gem 'uglifier', '>= 1.3.0'
# Use CoffeeScript for .js.coffee assets and views
gem 'coffee-rails', '~> 4.0.0'
# See https://github.com/sstephenson/execjs#readme for more supported runtimes
# gem 'therubyracer', platforms: :ruby

# Use jquery as the JavaScript library
gem 'jquery-rails'
gem 'jquery-ui-rails', '~>4.1.1'
# Turbolinks makes following links in your web application faster. Read more: https://github.com/rails/turbolinks
gem 'turbolinks'
# Build JSON APIs with ease. Read more: https://github.com/rails/jbuilder
gem 'jbuilder', '~> 2.0'
# bundle exec rake doc:rails generates the API under doc/api.
gem 'sdoc', '~> 0.4.0', group: :doc

# Use ActiveModel has_secure_password
# gem 'bcrypt', '~> 3.1.7'

# Use unicorn as the app server
# gem 'unicorn'

# Use Capistrano for deployment
# gem 'capistrano-rails', group: :development

# Use debugger
# gem 'debugger', group: [:development, :test]
```

Figure 2: App folder containing model-view-controller folders

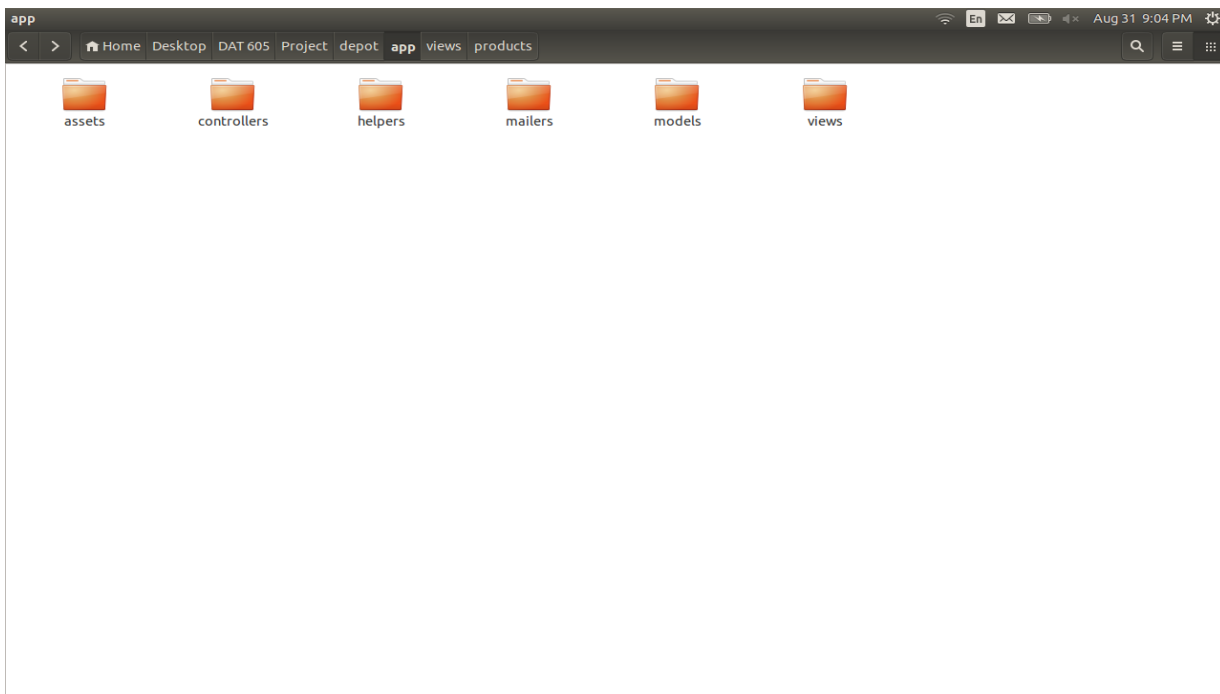


Figure 3: Models folder

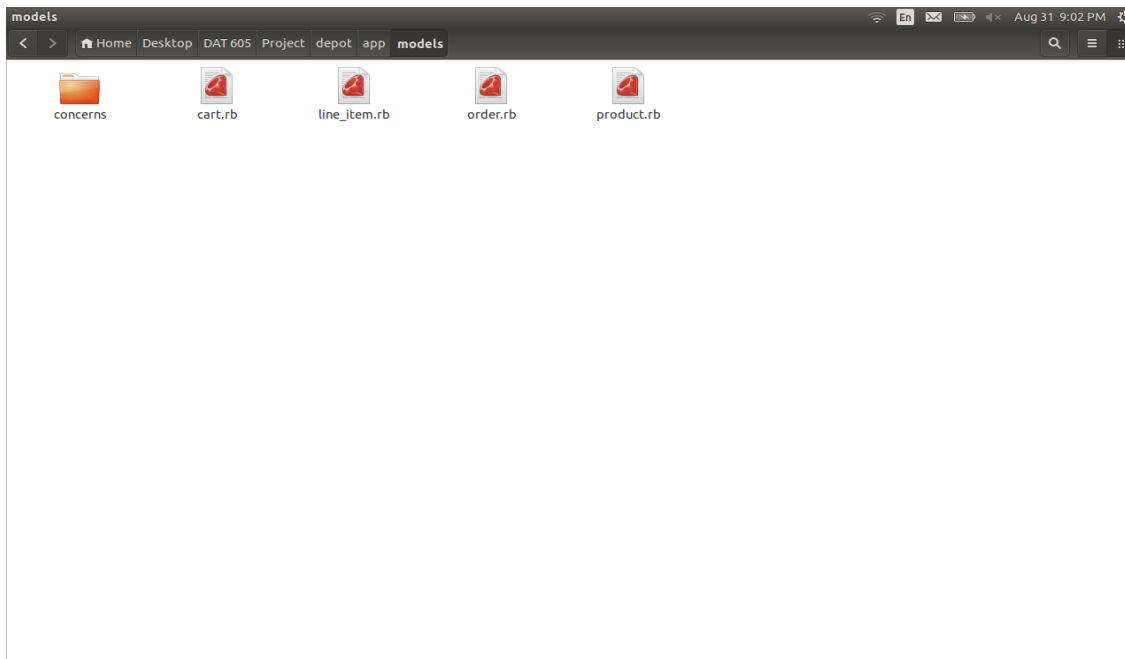


Figure 4: Controllers folder

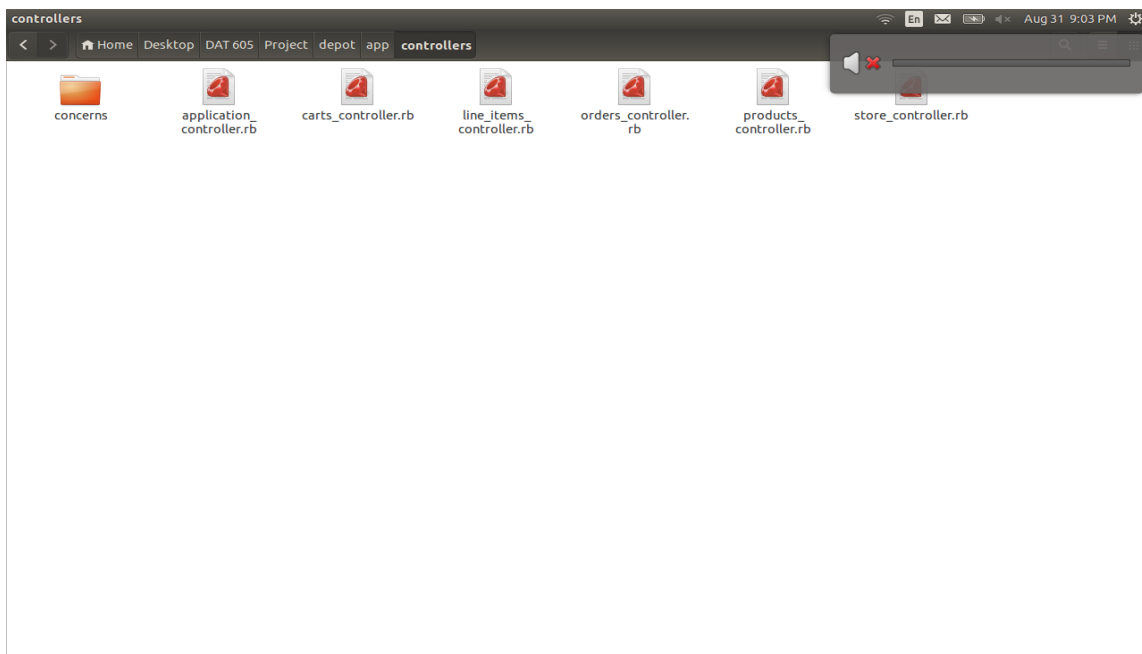


Figure 5: Views folder

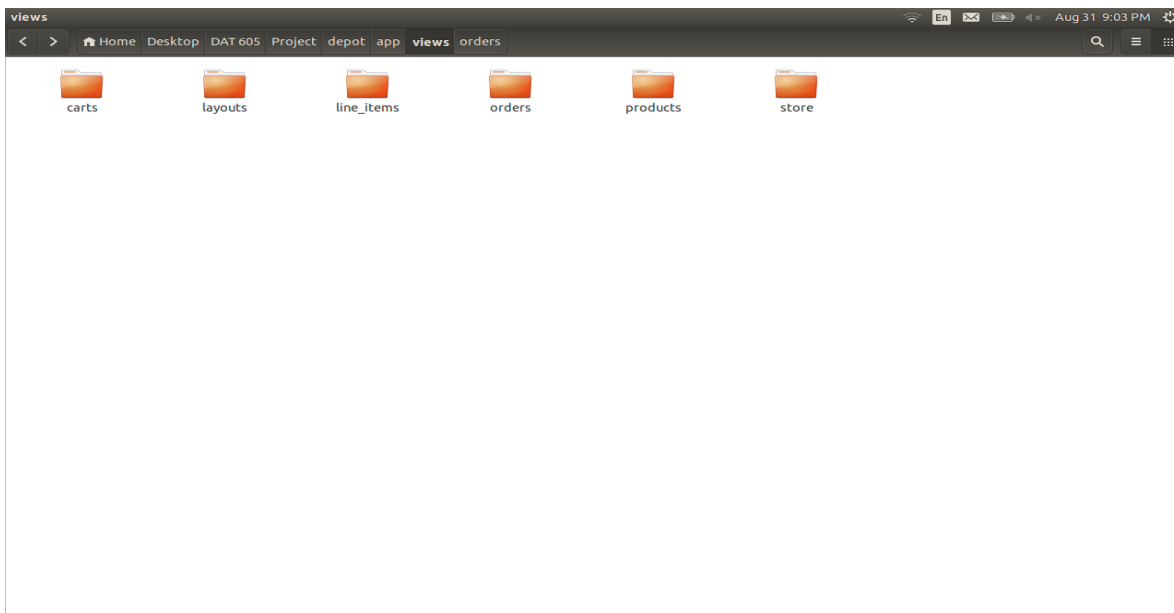


Figure 6: Bundler install

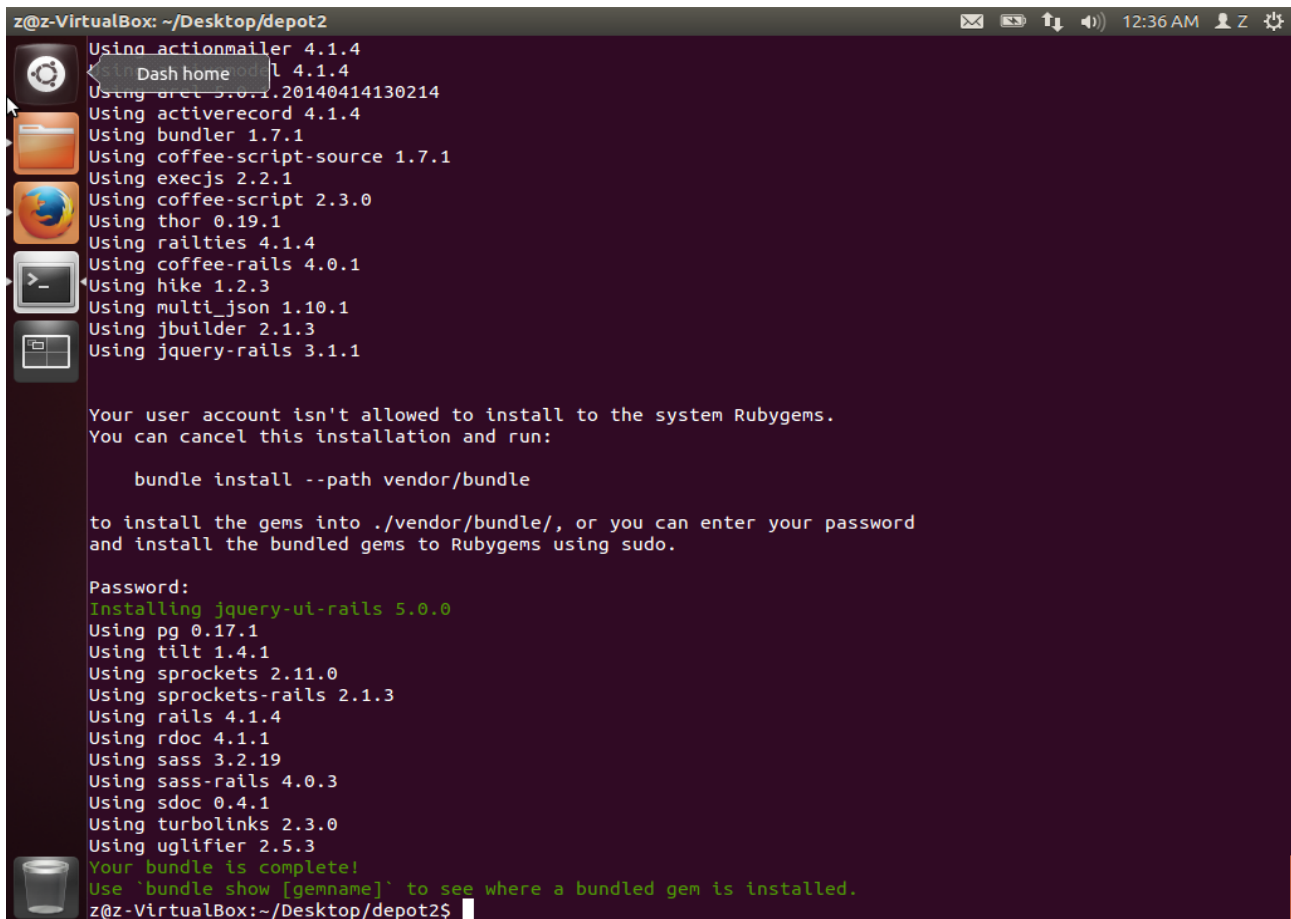
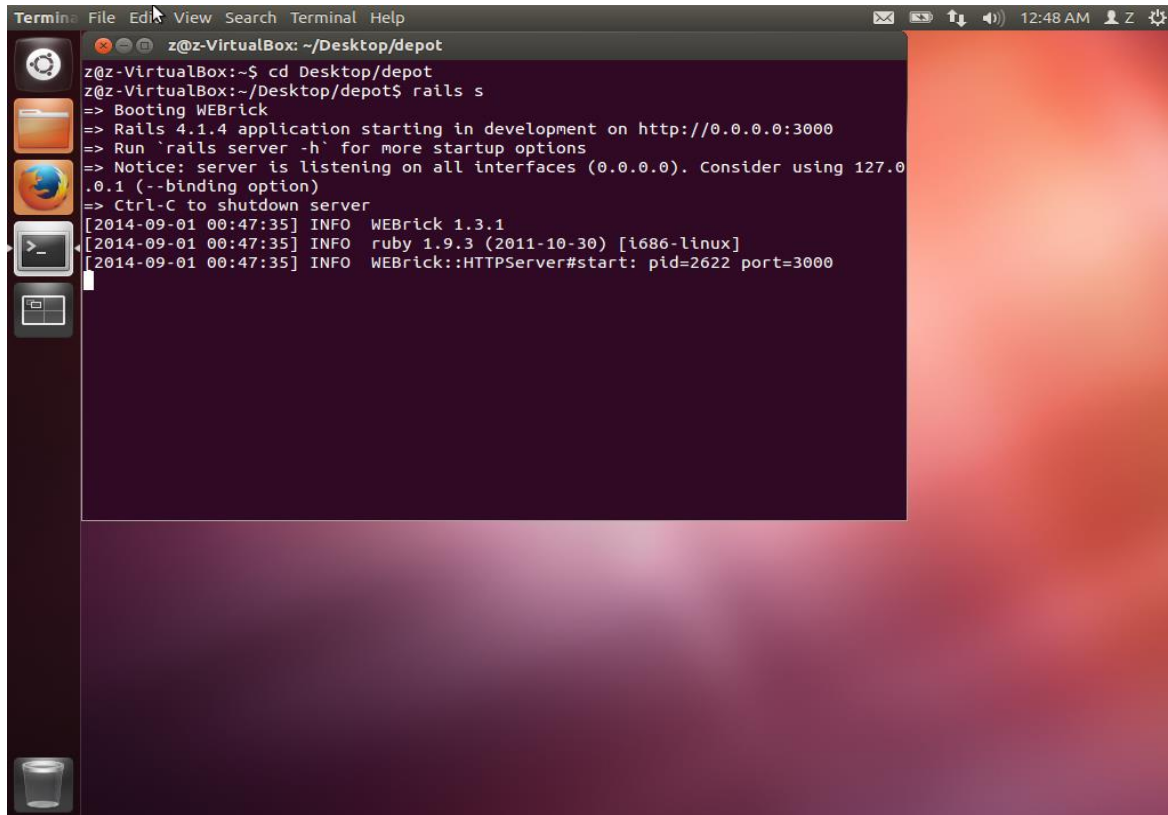


Figure 7: Rails server



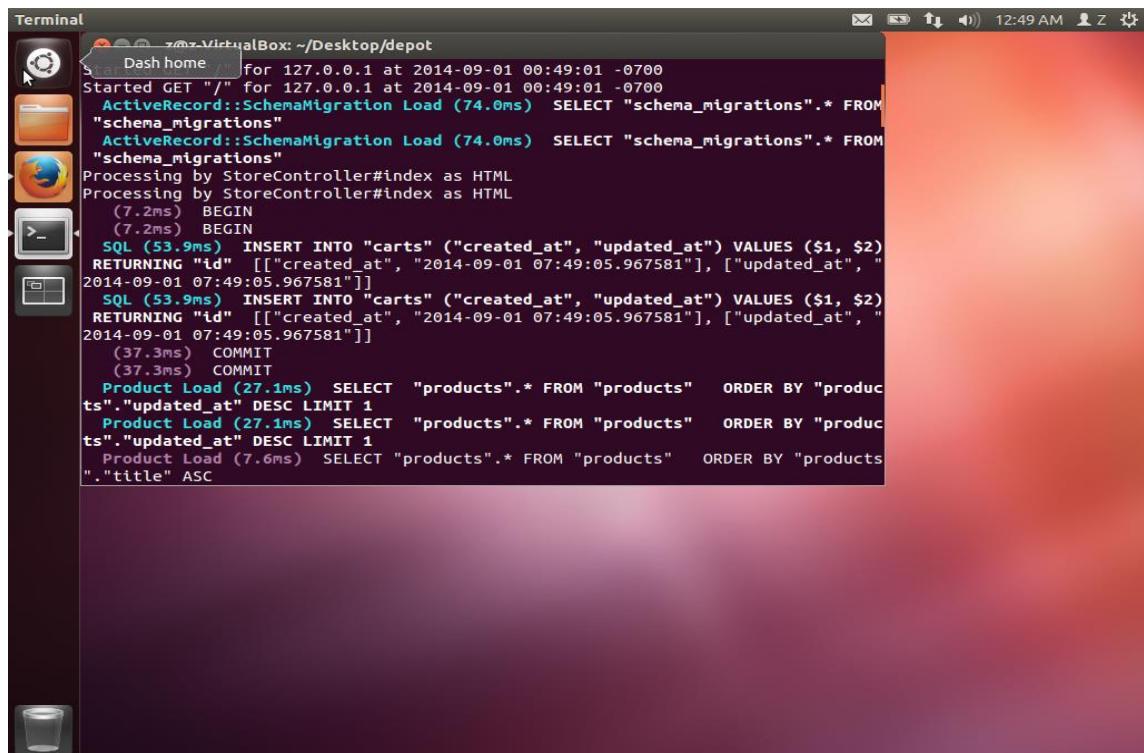
A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (12:48 AM, user Z). The prompt is "z@z-VirtualBox: ~/Desktop/depot". The user enters "cd Desktop/depot" and "rails s". The output shows the Rails 4.1.4 application starting on http://0.0.0.0:3000. It lists startup options and shows log messages: "[2014-09-01 00:47:35] INFO WEBRick 1.3.1", "[2014-09-01 00:47:35] INFO ruby 1.9.3 (2011-10-30) [i686-linux]", and "[2014-09-01 00:47:35] INFO WEBRick::HTTPServer#start: pid=2622 port=3000".

```

z@z-VirtualBox: ~/Desktop/depot
z@z-VirtualBox:~$ cd Desktop/depot
z@z-VirtualBox:~/Desktop/depot$ rails s
=> Booting WEBRick
=> Rails 4.1.4 application starting in development on http://0.0.0.0:3000
=> Run 'rails server -h' for more startup options
=> Notice: server is listening on all interfaces (0.0.0.0). Consider using 127.0.0.1 (--binding option)
=> Ctrl-C to shutdown server
[2014-09-01 00:47:35] INFO  WEBRick 1.3.1
[2014-09-01 00:47:35] INFO  ruby 1.9.3 (2011-10-30) [i686-linux]
[2014-09-01 00:47:35] INFO  WEBRick::HTTPServer#start: pid=2622 port=3000

```

Figure 8: Rails server after launch



A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (12:49 AM, user Z). The prompt is "z@z-VirtualBox: ~/Desktop/depot". The user enters "rails s". The output shows the Rails 4.1.4 application starting on http://0.0.0.0:3000. It lists startup options and shows log messages: "[2014-09-01 00:49:01 -0700] Started GET '/' for 127.0.0.1 at 2014-09-01 00:49:01 -0700", "[2014-09-01 00:49:01 -0700] ActiveRecord::SchemaMigration Load (74.0ms) SELECT 'schema_migrations'.* FROM 'schema_migrations'", "[2014-09-01 00:49:01 -0700] ActiveRecord::SchemaMigration Load (74.0ms) SELECT 'schema_migrations'.* FROM 'schema_migrations'", "[2014-09-01 00:49:01 -0700] Processing by StoreController#index as HTML", "[2014-09-01 00:49:01 -0700] Processing by StoreController#index as HTML", "[2014-09-01 00:49:01 -0700] (7.2ms) BEGIN", "[2014-09-01 00:49:01 -0700] (7.2ms) BEGIN", "[2014-09-01 00:49:01 -0700] SQL (53.9ms) INSERT INTO 'carts' ('created_at', 'updated_at') VALUES (\$1, \$2) RETURNING 'id' [2014-09-01 00:49:01 -0700] [2014-09-01 00:49:01 -0700] SQL (53.9ms) INSERT INTO 'carts' ('created_at', 'updated_at') VALUES (\$1, \$2) RETURNING 'id' [2014-09-01 00:49:01 -0700] [2014-09-01 00:49:01 -0700] (37.3ms) COMMIT", "[2014-09-01 00:49:01 -0700] (37.3ms) COMMIT", "[2014-09-01 00:49:01 -0700] Product Load (27.1ms) SELECT 'products'.* FROM 'products' ORDER BY 'products'. 'updated_at' DESC LIMIT 1", "[2014-09-01 00:49:01 -0700] Product Load (27.1ms) SELECT 'products'.* FROM 'products' ORDER BY 'products'. 'updated_at' DESC LIMIT 1", "[2014-09-01 00:49:01 -0700] Product Load (7.6ms) SELECT 'products'.* FROM 'products' ORDER BY 'products'. 'title' ASC".

```

z@z-VirtualBox: ~/Desktop/depot
z@z-VirtualBox:~$ rails s
=> Booting WEBRick
=> Rails 4.1.4 application starting in development on http://0.0.0.0:3000
=> Run 'rails server -h' for more startup options
=> Notice: server is listening on all interfaces (0.0.0.0). Consider using 127.0.0.1 (--binding option)
=> Ctrl-C to shutdown server
[2014-09-01 00:49:01 -0700] Started GET '/' for 127.0.0.1 at 2014-09-01 00:49:01 -0700
[2014-09-01 00:49:01 -0700] ActiveRecord::SchemaMigration Load (74.0ms) SELECT 'schema_migrations'.* FROM 'schema_migrations'
[2014-09-01 00:49:01 -0700] ActiveRecord::SchemaMigration Load (74.0ms) SELECT 'schema_migrations'.* FROM 'schema_migrations'
[2014-09-01 00:49:01 -0700] Processing by StoreController#index as HTML
[2014-09-01 00:49:01 -0700] Processing by StoreController#index as HTML
[2014-09-01 00:49:01 -0700] (7.2ms) BEGIN
[2014-09-01 00:49:01 -0700] (7.2ms) BEGIN
[2014-09-01 00:49:01 -0700] SQL (53.9ms) INSERT INTO 'carts' ('created_at', 'updated_at') VALUES ($1, $2) RETURNING 'id' [2014-09-01 00:49:01 -0700] [2014-09-01 00:49:01 -0700] SQL (53.9ms) INSERT INTO 'carts' ('created_at', 'updated_at') VALUES ($1, $2) RETURNING 'id' [2014-09-01 00:49:01 -0700] [2014-09-01 00:49:01 -0700] (37.3ms) COMMIT
[2014-09-01 00:49:01 -0700] (37.3ms) COMMIT
[2014-09-01 00:49:01 -0700] Product Load (27.1ms) SELECT 'products'.* FROM 'products' ORDER BY 'products'. 'updated_at' DESC LIMIT 1
[2014-09-01 00:49:01 -0700] Product Load (27.1ms) SELECT 'products'.* FROM 'products' ORDER BY 'products'. 'updated_at' DESC LIMIT 1
[2014-09-01 00:49:01 -0700] Product Load (7.6ms) SELECT 'products'.* FROM 'products' ORDER BY 'products'. 'title' ASC

```


Figure 9: Heroku.com application page

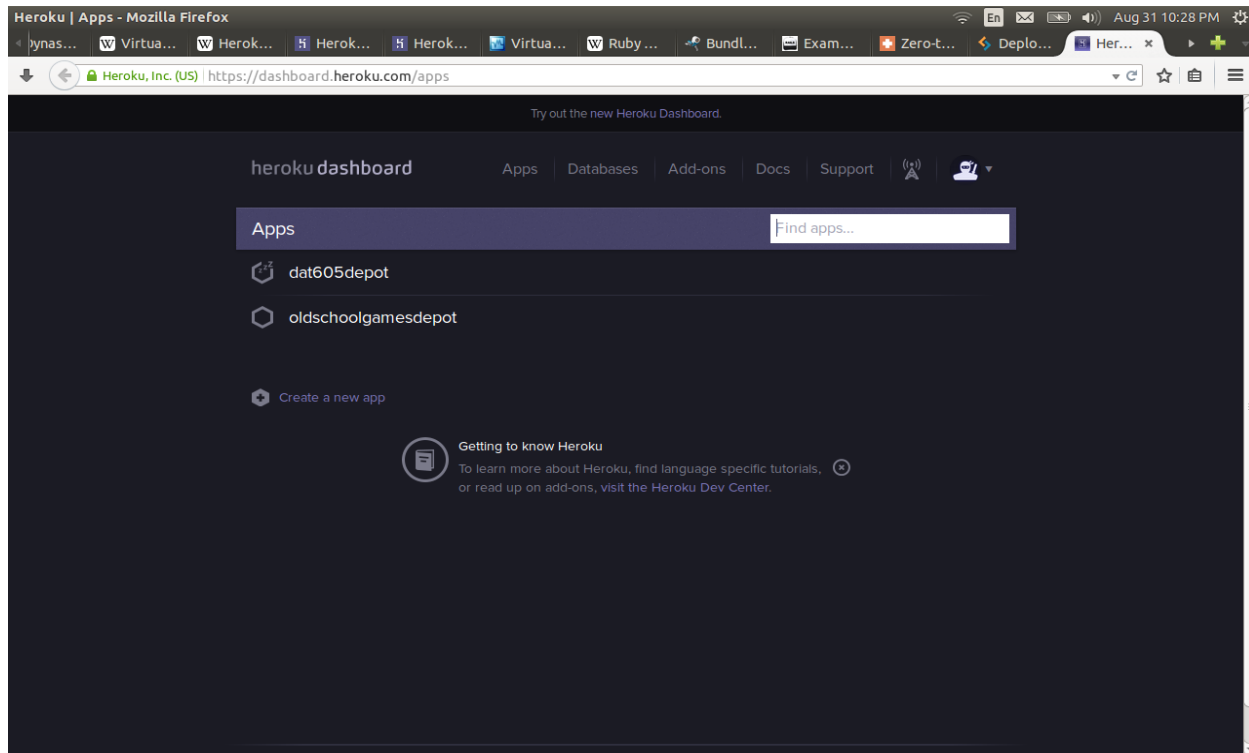


Figure 10: First deployment of application

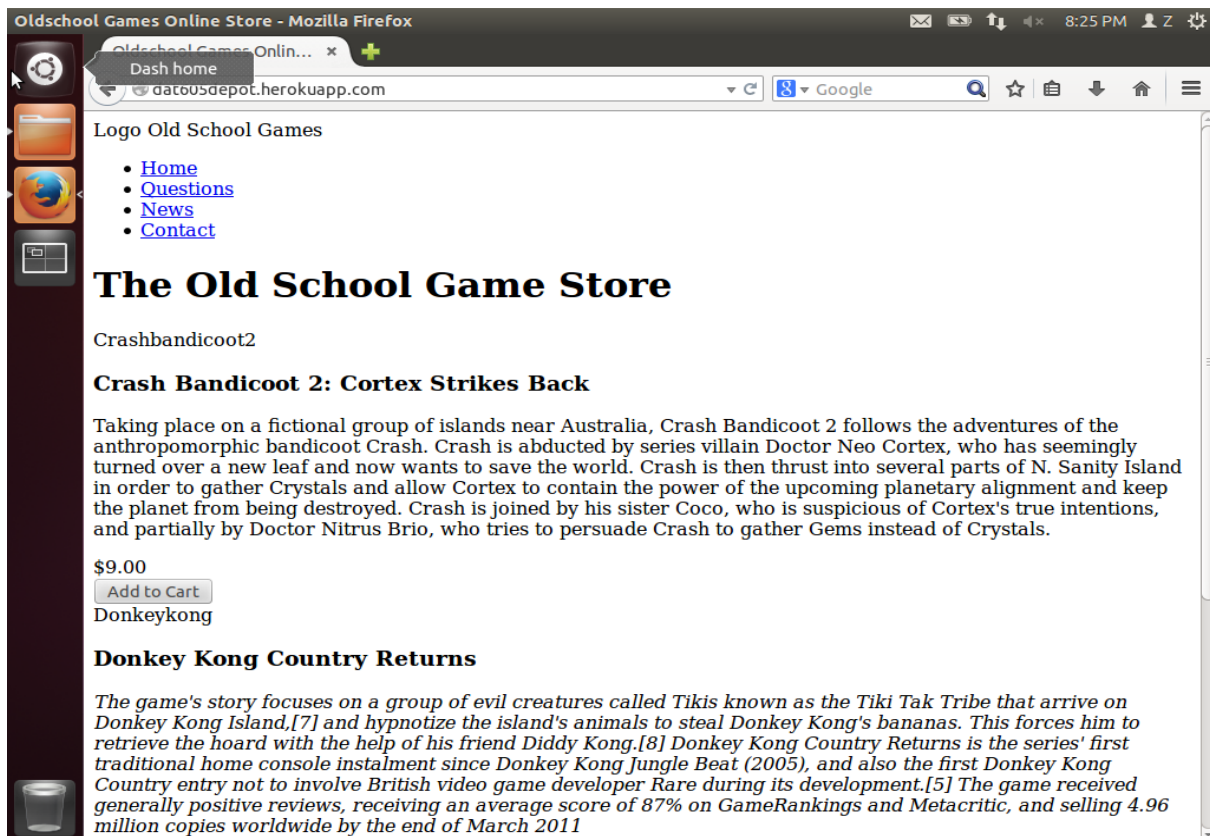


Figure 11: Database model in 3NF

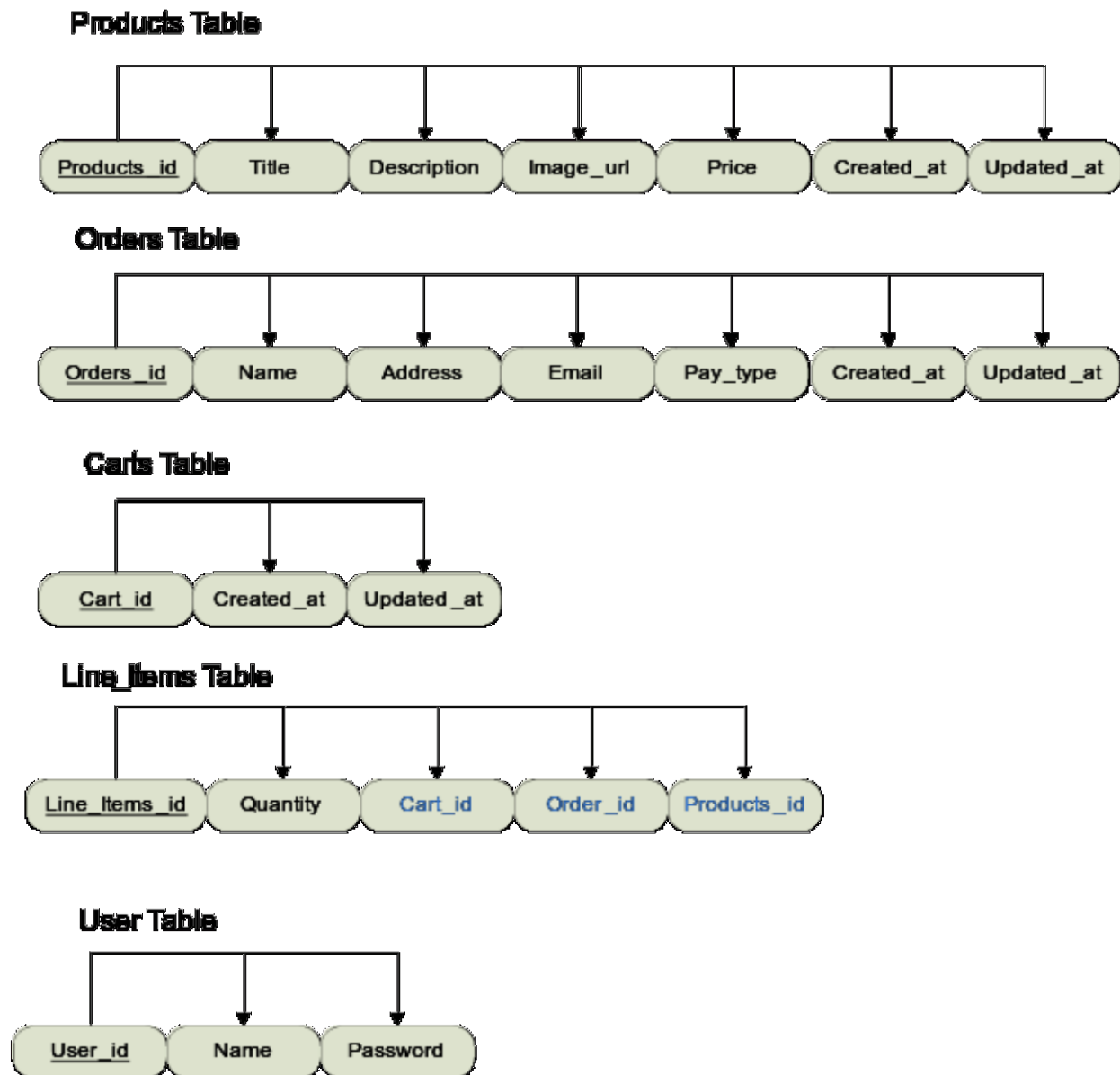
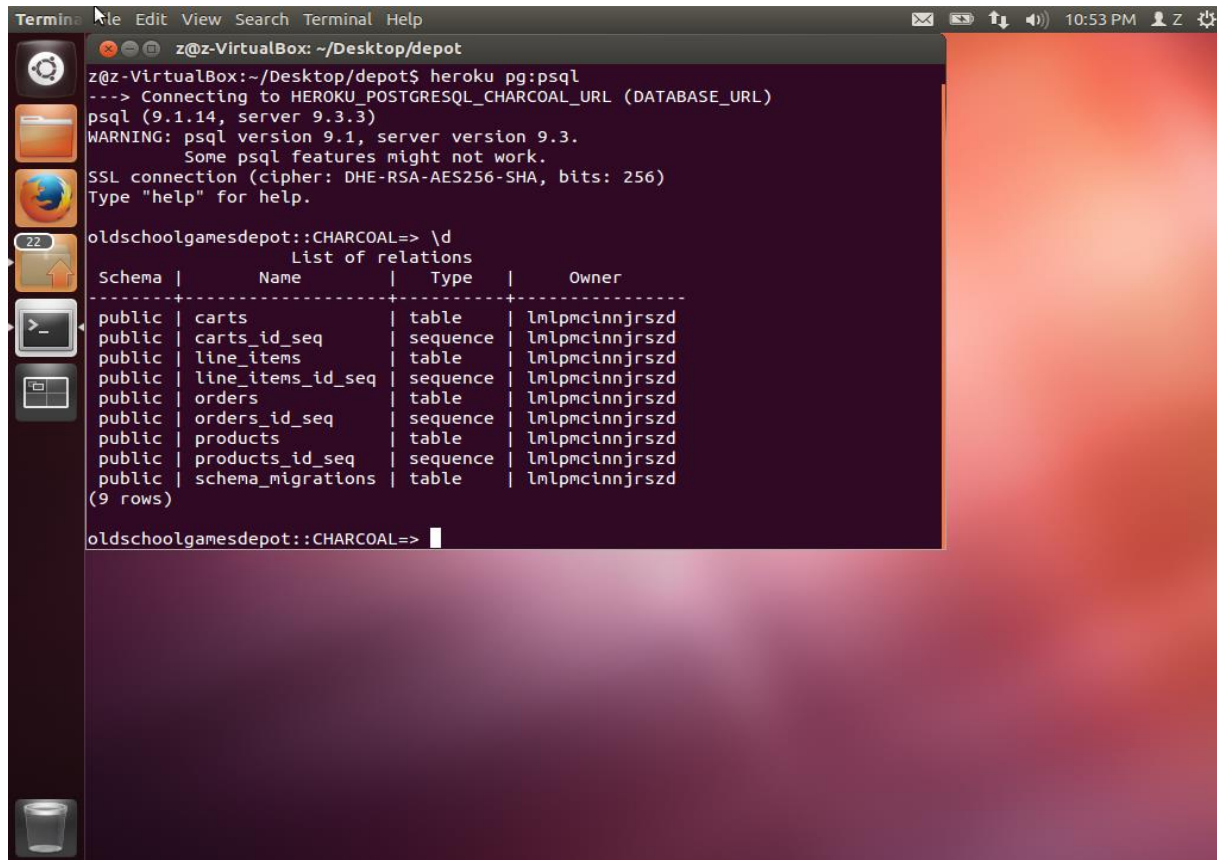


Figure 12: Database schema from Terminal using PostgreSQL console



```

Terminal z@z-VirtualBox: ~/Desktop/depot
z@z-VirtualBox:~/Desktop/depot$ heroku pg:psql
---> Connecting to HEROKU_POSTGRESURL_CHARCOAL_URL (DATABASE_URL)
psql (9.1.14, server 9.3.3)
WARNING: psql version 9.1, server version 9.3.
        Some psql features might not work.
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.

oldschoolgamesdepot::CHARCOAL=> \d
List of relations
Schema |      Name      | Type  | Owner
-----+-----+-----+-----
public | carts           | table | lmlpmcinnjrszd
public | carts_id_seq    | seq   | lmlpmcinnjrszd
public | line_items      | table | lmlpmcinnjrszd
public | line_items_id_seq | seq   | lmlpmcinnjrszd
public | orders          | table | lmlpmcinnjrszd
public | orders_id_seq   | seq   | lmlpmcinnjrszd
public | products        | table | lmlpmcinnjrszd
public | products_id_seq | seq   | lmlpmcinnjrszd
public | schema_migrations | table | lmlpmcinnjrszd
(9 rows)

oldschoolgamesdepot::CHARCOAL=>

```

Figure 13: Home Page

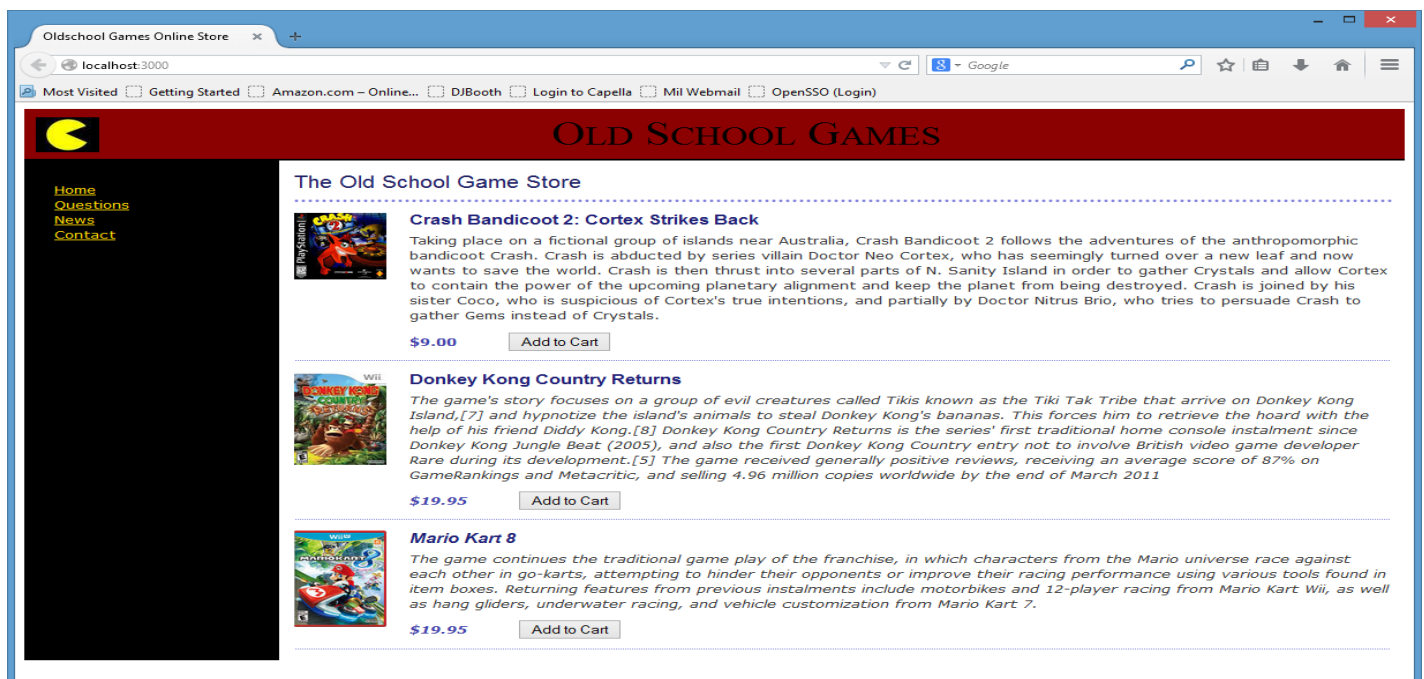


Figure 14: Home page with Cart

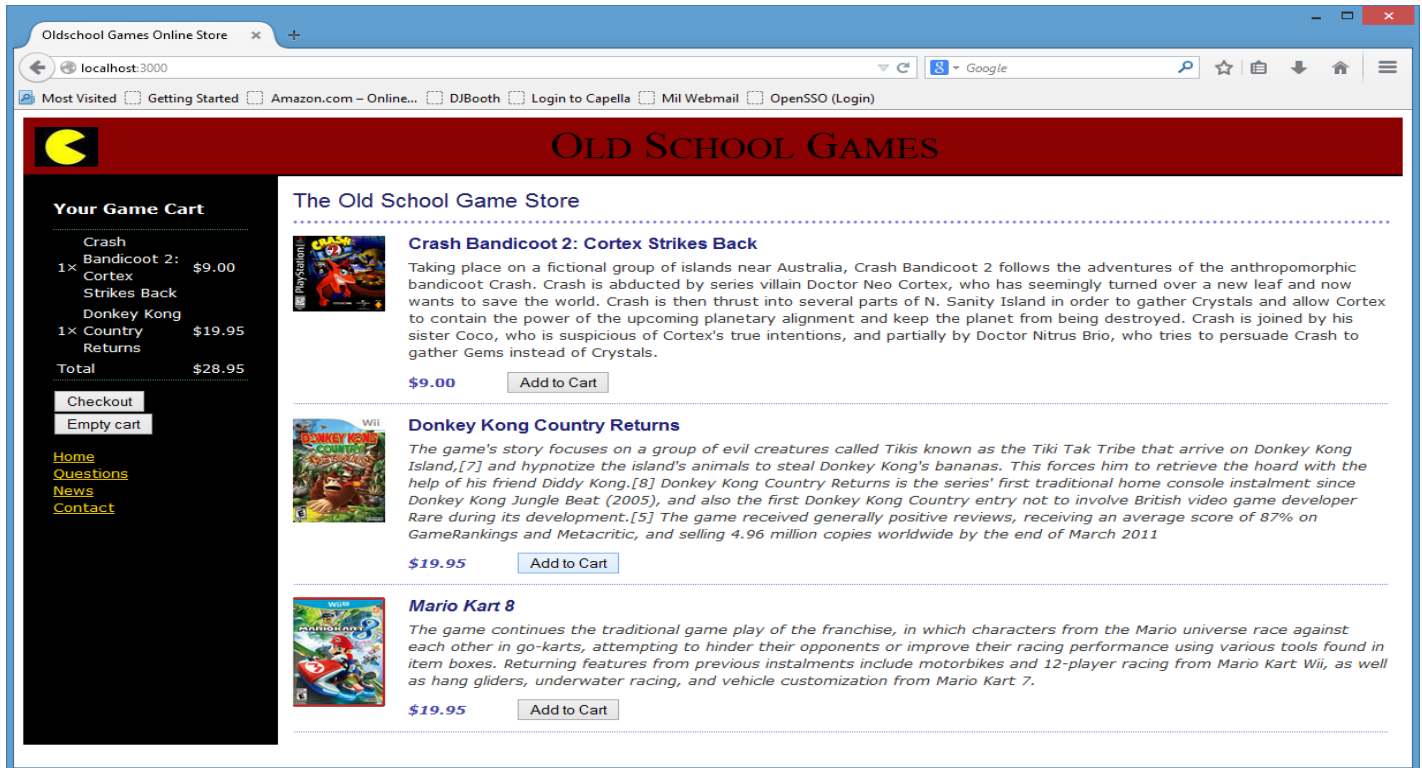


Figure 15: Edit product page

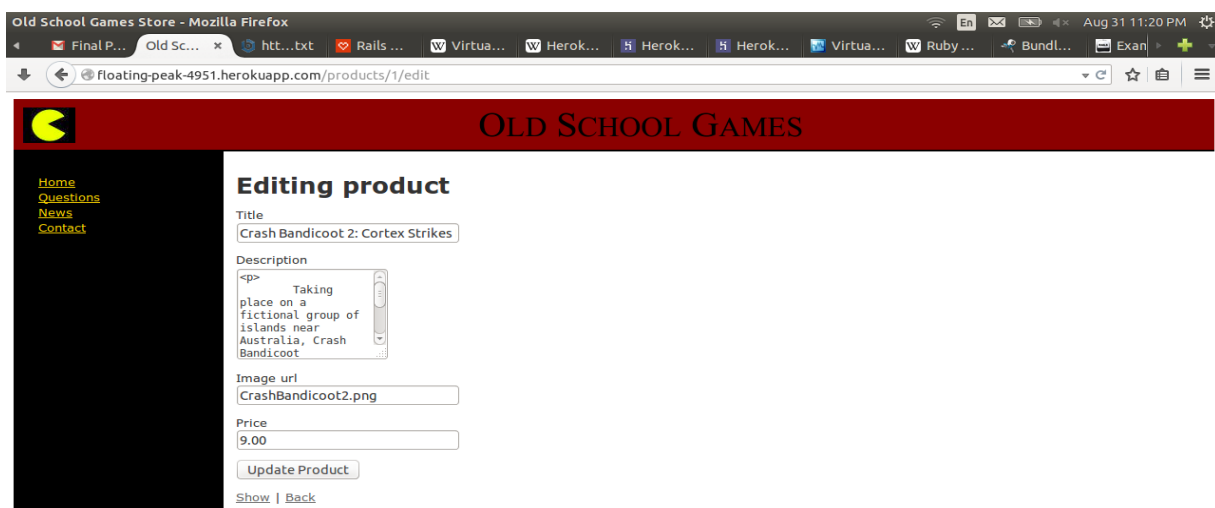


Figure 16: New product page

Old School Games Store - Mozilla Firefox

floating-peak-4951.herokuapp.com/products/new

OLD SCHOOL GAMES

[Home](#)
[Questions](#)
[News](#)
[Contact](#)

New product

Title

Description

Image url

Price

[Back](#)

Figure 17: Error message 1

Old School Games Store - Mozilla Firefox

floating-peak-4951.herokuapp.com/orders

OLD SCHOOL GAMES

Your Cart

Crash
1x Bandicoot 2: Cortex Strikes Back \$9.00
Total \$9.00

[Home](#)
[Questions](#)
[News](#)
[Contact](#)

New order

4 errors prohibited this order from being saved:

- Name can't be blank
- Address can't be blank
- Email can't be blank
- Pay type is not included in the list

Name

Address

Email

Pay type

Select a payment method

[Back](#)

Figure 18: Error message 2

Old School Games Store - Mozilla Firefox

floating-peak-4951.herokuapp.com/orders

OLD SCHOOL GAMES

Your Cart

Crash
1x Bandicoot 2: Cortex Strikes Back
Back

Total \$9.00

Checkout
Empty cart

[Home](#)
[Questions](#)
[News](#)
[Contact](#)

New order

1 error prohibited this order from being saved:

- Address can't be blank

Name
Zailyn Tamayo

Address

Email
zailyn@gmail.com

Pay type
Credit card

Place Order

[Back](#)

Figure 19: Order complete home page

Old School Games Store - Mozilla Firefox

floating-peak-4951.herokuapp.com

OLD SCHOOL GAMES

[Home](#)
[Questions](#)
[News](#)
[Contact](#)

Thank you for your order.

Your Old School Game Catalog

Crash Bandicoot 2: Cortex Strikes Back

Taking place on a fictional group of islands near Australia, Crash Bandicoot 2 follows the adventures of the anthropomorphic bandicoot Crash. Crash is abducted by series villain Doctor Neo Cortex, who has seemingly turned over a new leaf and now wants to save the world. Crash is then thrust into several parts of N. Sanity Island in order to gather Crystals and allow Cortex to contain the power of the upcoming planetary alignment and keep the planet from being destroyed. Crash is joined by his sister Coco, who is suspicious of Cortex's true intentions, and partially by Doctor Nitrus Brio, who tries to persuade Crash to gather Gems instead of Crystals.

\$9.00 [Add to Cart](#)

Donkey Kong Country Returns

The game's story focuses on a group of evil creatures called Tikis known as the Tiki Tak Tribe that arrive on Donkey Kong Island,[7] and hypnotize the island's animals to steal Donkey Kong's bananas. This forces him to retrieve the hoard with the help of his friend Diddy Kong.[8] Donkey Kong Country Returns is the series' first traditional home console instalment since Donkey Kong Jungle Beat (2005), and also the first Donkey Kong Country entry not to involve British video game developer Rare during its development.[5] The game received generally positive reviews, receiving an average score of 87% on GameRankings and Metacritic, and selling 4.96 million copies worldwide by the end of March 2011

\$19.95 [Add to Cart](#)

Mario Kart 8

continues the traditional game play of the franchise, in which characters from the Mario universe race against each other in go-karts, to hinder their opponents or improve their racing performance using various tools found in item boxes. Returning features from previous include motorbikes and 12-player racing from Mario Kart Wii, as well as hang gliders, underwater racing, and vehicle customization Kart 7.

\$19.95 [Add to Cart](#)

Figure 20: Unexpected token error

```

z@z-VirtualBox: ~/Desktop/depot2
0bd-9838-4971-973c-fc09c0d9f418/public/assets/logo-e5cfefe0bcf226125721da60749c3
Dash home
rake aborted!
ExecJS::ProgramError: Unexpected token: operator (=) (line: 10323, col: 0
, pos: 283348)
Error
  at new JS_Parse_Error (/tmp/execjs20140825-1071-xwrhnhjs:2357:10623)
  at js_error (/tmp/execjs20140825-1071-xwrhnhjs:2357:10842)
  at croak (/tmp/execjs20140825-1071-xwrhnhjs:2357:19067)
  at token_error (/tmp/execjs20140825-1071-xwrhnhjs:2357:19204)
  at unexpected (/tmp/execjs20140825-1071-xwrhnhjs:2357:19292)
  at /tmp/execjs20140825-1071-xwrhnhjs:2357:27484
  at /tmp/execjs20140825-1071-xwrhnhjs:2357:29977
  at expr_ops (/tmp/execjs20140825-1071-xwrhnhjs:2357:30735)
  at /tmp/execjs20140825-1071-xwrhnhjs:2357:30827
  at /tmp/execjs20140825-1071-xwrhnhjs:2357:31270
  (in /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/app/assets/javascript
s/application.js)/tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/r
uby/2.0.0/gems/execjs-2.2.1/lib/execjs/external_runtime.rb:64:in `extract_result'

    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/execjs-2.2.1/lib/execjs/external_runtime.rb:29:in `exec'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/uglifier-2.5.3/lib/uglifier.rb:187:in `run uglifyjs'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/uglifier-2.5.3/lib/uglifier.rb:170:in `compile'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/sprockets-2.11.0/lib/sprockets/uglifier_compressor.rb:25:in `evaluate'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/tilt-1.4.1/lib/tilt/template.rb:103:in `render'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/sprockets-2.11.0/lib/sprockets/context.rb:197:in `block in evaluate'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/sprockets-2.11.0/lib/sprockets/context.rb:194:in `each'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/sprockets-2.11.0/lib/sprockets/context.rb:194:in `evaluate'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/sprockets-2.11.0/lib/sprockets/bundled_asset.rb:25:in `initialize'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/
gems/sprockets-2.11.0/lib/sprockets/base.rb:377:in `new'
    /tmp/build_1ee640bd-9838-4971-973c-fc09c0d9f418/vendor/bundle/ruby/2.0.0/

```

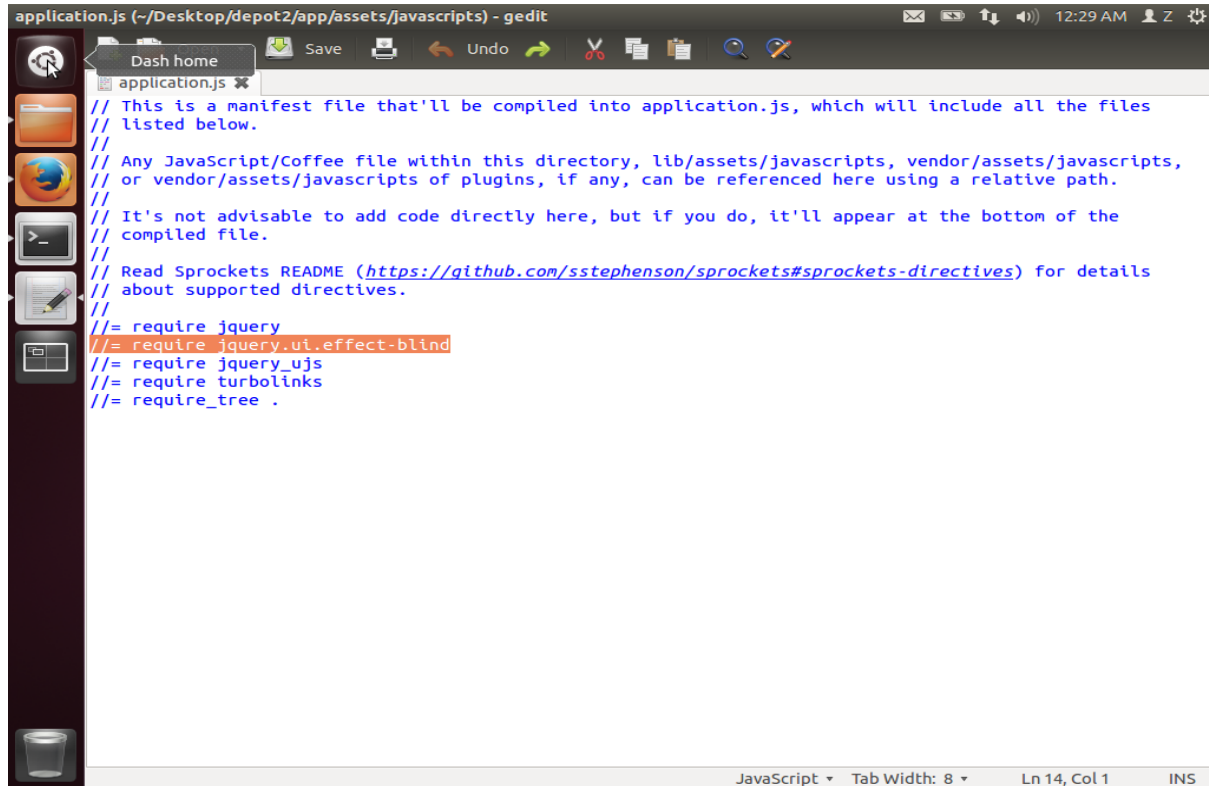
Figure 21: JQuery error

```

Terminal
z@z-VirtualBox: ~/Desktop/depot2
I, [2014-08-25T07:26:51.790193 #975] INFO -- : Writing /tmp/build_a1f9ec
3c83/public/assets/logo-e5cfefe0bcf226125721da60749c3b95.png
rake aborted!
Sprockets::FileNotFound: couldn't find file 'jquery.ui.effect-blind'
(in /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/app/assets/javascript
/tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/context.rb:106:in `resolve'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/context.rb:146:in `require_asset'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/directive_processor.rb:217:in `process_require_directive'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/directive_processor.rb:167:in `block in process_directives'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/directive_processor.rb:165:in `each'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/directive_processor.rb:165:in `process_directives'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/directive_processor.rb:99:in `evaluate'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
emplate.rb:103:in `render'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/context.rb:197:in `block in evaluate'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/context.rb:194:in `each'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/context.rb:194:in `evaluate'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/processed_asset.rb:12:in `initialize'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/base.rb:374:in `new'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/base.rb:374:in `block in build_asset'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/base.rb:395:in `circular_call_protection'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/base.rb:373:in `build_asset'
    /tmp/build_a1f9ec6f-4129-4b2e-ae8a-312ec7e3c83/vendor/bundle/ruby/2.0.0/
sprockets/index.rb:94:in `block in build_asset'

```

Figure 22: Application.js file jquery 1



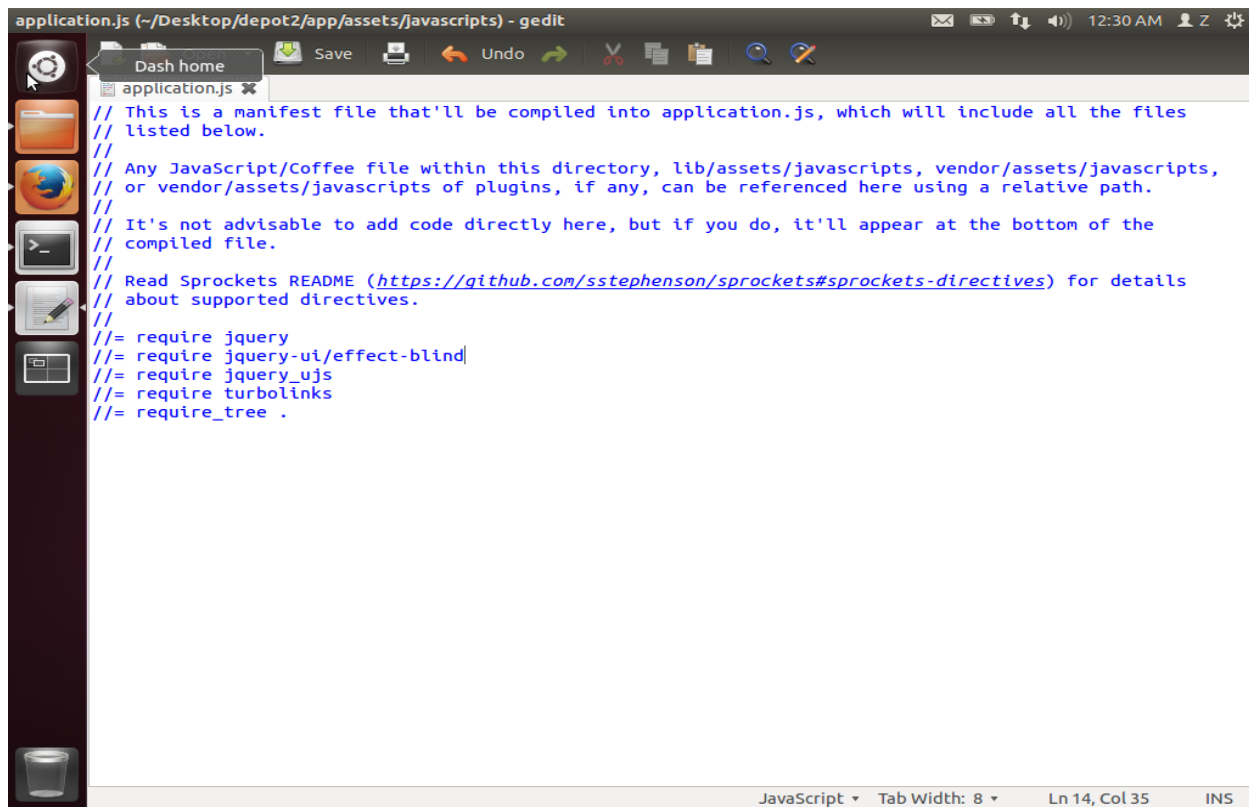
```

application.js (~/.Desktop/depot2/app/assets/javascripts) - gedit
application.js x
// This is a manifest file that'll be compiled into application.js, which will include all the files
// listed below.
//
// Any JavaScript/Coffee file within this directory, lib/assets/javascripts, vendor/assets/javascripts,
// or vendor/assets/javascripts of plugins, if any, can be referenced here using a relative path.
//
// It's not advisable to add code directly here, but if you do, it'll appear at the bottom of the
// compiled file.
//
// Read Sprockets README (https://github.com/sstephenson/sprockets#sprockets-directives) for details
// about supported directives.
//
//= require jquery
//= require jquery.ui.effect-blind
//= require jquery_ujs
//= require turbolinks
//= require_tree .

```

JavaScript Tab Width: 8 Ln 14, Col 1 INS

Figure 23: Application.js file jquery 2



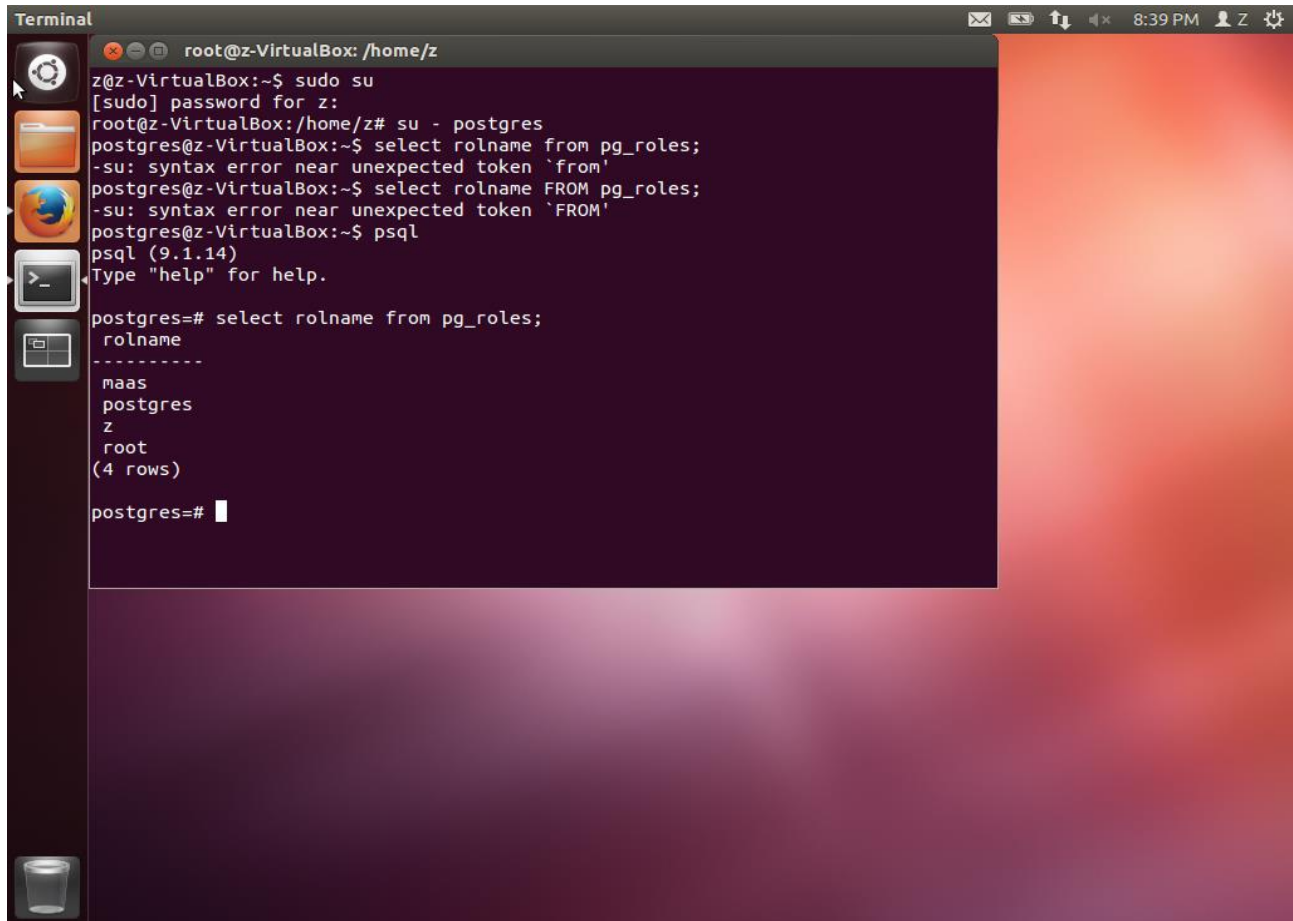
```

application.js (~/.Desktop/depot2/app/assets/javascripts) - gedit
application.js x
// This is a manifest file that'll be compiled into application.js, which will include all the files
// listed below.
//
// Any JavaScript/Coffee file within this directory, lib/assets/javascripts, vendor/assets/javascripts,
// or vendor/assets/javascripts of plugins, if any, can be referenced here using a relative path.
//
// It's not advisable to add code directly here, but if you do, it'll appear at the bottom of the
// compiled file.
//
// Read Sprockets README (https://github.com/sstephenson/sprockets#sprockets-directives) for details
// about supported directives.
//
//= require jquery
//= require jquery-ui/effect-blind
//= require jquery_ujs
//= require turbolinks
//= require_tree .

```

JavaScript Tab Width: 8 Ln 14, Col 35 INS

Figure 24: PostgreSQL roles



The image shows a terminal window titled "Terminal" with a dark background. The window displays a series of commands and their outputs related to PostgreSQL roles. The user starts as 'root' and switches to 'postgres' using 'sudo su'. They attempt to run 'select rolname from pg_roles;' but receive a syntax error. After correcting the command to 'select rolname FROM pg_roles;', they run 'psql' and then 'select rolname from pg_roles;'. The output shows a list of roles: 'maas', 'postgres', 'z', and 'root'.

```
Terminal
root@z-VirtualBox: /home/z
z@z-VirtualBox:~$ sudo su
[sudo] password for z:
root@z-VirtualBox:/home/z# su - postgres
postgres@z-VirtualBox:~$ select rolname from pg_roles;
-su: syntax error near unexpected token `from'
postgres@z-VirtualBox:~$ select rolname FROM pg_roles;
-su: syntax error near unexpected token `FROM'
postgres@z-VirtualBox:~$ psql
psql (9.1.14)
Type "help" for help.

postgres=# select rolname from pg_roles;
 rolname 
-----
 maas
 postgres
 z
 root
(4 rows)

postgres=#
```