Ruby on Rails

*Start: 3/6/2016 at 12:45 pm*

**Section 1: Intro**

Introduction

Section 1: Setup and Preview Web Apps we will build  
 Section 2: How to get help! Discussions or <https://discordapp.com/channels>  
 Learn about Ruby, language behind Rails  
 Section 3: Introduction to Rails, framework  
 Section 4 and on: Start building some Web Apps

Course Structure

Discussion Area will become your best friend, use it regularly   
 After each video is a text with directions to reference   
 Rails Version 4.2.3

Development Environment  
 nitrous vs cloud9

Development Environment Update  
 Read Article

Lecture 5 and Lecture 6 Skipped

Get Familiar with IDE (Lecture 7)  
 Get Familiar with Nitrous (course taught in nitrous)

Alternate IDE Cloud9

rvm allows us to switch ruby versions  
 heroku installing process will be same as nitrous

Lecture 10 - SaaS App -   
 This is the preview video of the Software as a Service app built as the final bonus project of this course, this will be built in section 10 of this course. Please note: This is a preview video for viewing purposes only, the url for the web app built in section 10 is different so don't try to enter in the app url and follow along as it will not work.

Lecture 11 - Alpha-blog -   
 This is the preview video for the Alpha-blog built as the first complete web app in this course, building this app will cover sections 4 through 7 of this course. Building of this app will feature all the basics of Rails and no shortcuts and generators (except for migrations) will be used, all code will be done from scratch

Lecture 12 - Stock-tracker -   
 This is the preview video for the Stock tracker social media app built in section 8 of this course. This app introduces rapid development techniques and use of generators to speed up the prototyping process.

Lecture 13 - Photo-app -   
 This is the preview video for the Photo-app which will be covered in section 9 of this course. Email authentication, Image uploads (production and development) and Payment processing is introduced in building this web app.

**Section 2: The Ruby Programming Language**

How to Get Free Live Help!

Discussions Board  
 www.mashrurhossain.com

Introduction to Ruby

Object Oriented Programming Language   
 puts put string run in terminal by ruby /filename  
 def define < -- method

Advantage of using Method: we can pass in arguments and keep calling

Working with Strings

Group of characters in a sequence, written in ‘ ’ or “ ”   
 Ruby irb is a playground to try out code ctrl l will clear the screen exit leaves irb

*Finish: 3/6/2016 at 3:00 pm (2 hours 15 minutes) 6% complete*

*Start: 3/7/2016 at 9:50 pm*

Working with Numbers

puts 20/3.to\_f (to float will give decimal) puts 20/3.0 will do same   
 x = 3 y=20 y = y.to\_f puts y/x (will give decimal)

23.odd? true 22.odd? even instead of having to use %

rand random # rand(10) random number between 0 and 9

Methods and Branching

# will comment in ruby

*Finish: 3/7/2016 at 11:50 pm (2 hours) 9% complete*

*Start: 3/9/2016 at 12:30 pm*

Arrays and Iterators

! bang will actually update the array  
 << shovel adds to the end  
 .push( ) adds to the end  
 .unshift( ) adds to the beginning  
 .pop( ) remove last item from array  
 .uniq gives unique entries (removes repeats)

Iterators loop through array and perform a certain function on the array

.each { do something }

Hashes

{‘key’ => ‘value’} myHash = {a: 1, b: 2… }

myhash.delete(:b)

*Finish: 3/7/2016 at 2:00 pm (1 hour 30 min) 10% complete*

*Start: 3/10/2016 at 1:45 pm*

Ruby Style Guide

<https://github.com/bbatsov/ruby-style-guide>

Practice what you learn

[www.tryruby.org](http://www.tryruby.org)

*Finish: 3/10/2016 at 3:45 pm (2 hours) 12% complete*

*Start: 3/11/2016 at 1:45 pm*

Introduction to Object Oriented Programming

Everything in ruby is an object

Inheritance and Modules

Modules – similar to class but is collection of behaviors other classes can use via mix ins

You can’t have an object of a module, a module has to be mixed into a class and then can use methods of a modules

Module variable names end with –able if you wanted talk the variable would say module talkable

*Finish: 3/11/2016 at 3:15 pm (1 hour 30 min) 13% complete*

*Start: 3/12/2016 at 9:15 am*

Final Ruby Project – working with JSON

*Finish: 3/12/2016 at 10:15 am (1 hour) 14% complete*

*Start: 3/12/2016 at 12:00 pm*

**Section 3: Introduction to Ruby on Rails**

Introduction to Section 3 (Read Ruby on Rails Guides… listed below)

Model, View, Controller, and Rails App Structure

What is Rails? What is rubyonrails.org? <http://guides.rubyonrails.org/>

What is rubygems.org? <https://rubygems.org/>

Who is DHH? <http://david.heinemeierhansson.com/>

What are Ruby Meetups? <http://www.meetup.com/topics/ruby/>

What are the components of a Rails Application? Model, View, Control (MVC)

$ rails new /project\_name will create a new rails folder, bundle install etc

*Finish: 3/12/2016 at 4 pm (4 hours) 17% complete*

*Start: 3/13/2016 at 10:30 pm*

Root Route, Git, and Version Control

git init git add –A git commit –m “type message here”

in settings in the cloud you can go to settings – show hidden files and enter files you don’t want git to track

git checkout –f (rejects the latest change)

Setup GitHub Repository

cat ~/.ssh/id\_rsa.pub displays my public ssh key can add that shh key to your github account 🡪 settings – SSH key, add keys, (give it a title like cloud9), key is what you paste from the term.

Create New Repository in GitHub

Change to ssh (from http) since we already did that above

git remote add origin https://github.com/code/cuddy/FILE\_NAME

git remote –v will check that it was added correctly

git push –u origin master

Production Deploy

You can update the toolbelt by typing the following command in the Terminal:

wget -O- https://toolbelt.heroku.com/install-ubuntu.sh | sh

heroku login [charliecuddy4@gmail.com](mailto:charliecuddy4@gmail.com) Worms123

heroku keys:add now don’t have to enter info each time (add shh from above)

heroku create  
 git push heroku master  
 heroku rename typenewnamehere

[*https://alpha-blog-cuddy.herokuapp.com/*](https://alpha-blog-cuddy.herokuapp.com/)

*Finish: 3/14/2016 at 12:30 am (2 hours) 20% complete*

*Start: 3/14/2016 at 9:15 pm*

CRUD, Scaffold and Wrap Section

Create, Read, Update, Delete the point of the data base is CRUD

Rails generate scaffold Article title:string description:text (capital A signify it’s a model)

*Finish: 3/14/2016 at 10:45 pm (1 hour 30 min) 21% complete*

*Start: 3/15/2016 at 10:45 pm*

**Section 4: CRUD Operations in Ruby on Rails**

Introduction to Section 4: Models, Migrations and Rails Console

model name: Article (singular, first letter uppercase)  
 table name: articles (plural, lower case of model name)  
 model name filename: all lowercase but singular, article.rb  
 controller name: plural of model (snake case) so articles\_controller.rb

ex: model name: User  
 table name: users  
 model name filename: user.rb  
 controller name: users\_controller.rb

To generate a migration to create a table (in this example articles):

rails generate migration create\_articles

To add attributes for the table in the migration file, add the following inside create\_table block:

t.string :title

t.text :description

t.timestamps

To run the migration file and create the articles table:

rake db:migrate (I like this one better)

OR

bundle exec rake db:migrate

To rollback a migration (undo the last migration):

rake db:rollback

To add a column (example: created\_at column) to the articles table:

rails generate migration add\_created\_at\_to\_articles

Then within the def change method in the migration file:

add\_column :articles, :created\_at, :datetime

To add a different column (example: name) to a users table:

rails generate migration add\_name\_to\_users

Then within the def change method in the migration file:

add\_column :users, :name, :string

In the above two adding column methods, the first argument is the name of the table, second is the attribute name and third is the type

To create a model file for Article:

- In the app/models folder create a file called article.rb

- Fill it in with the following ->

class Article < ActiveRecord::Base

end

To start the rails console:

rails console

To test connection to the articles table:

Article.all # classname.all will list all the articles in the articles table

Then simply type in Article (classname) to view the attributes

To create a new article with attributes title and description:

article = Article.new(title: "This is a test title", description: "This is a test description")

article.save

OR

article = Article.new

article.title = "This is a test title"

article.description = "This is a test description"

article.save

Another method to do the same:

article = Article.create(title: "This is a test title", description: "This is a test description") # This will hit the table right away without needing the article.save line

*Finish: 3/15/2016 at 11:15 pm (30 min) 22% complete*

*Start: 3/16/2016 at 9:45 pm*

Edit, Delete, and Validations

To find an article with id 2 and edit it's title:

article = Article.find(2) # Here assumption is article with id of 2 was being looked for

article.title = "This is an edited title"

article.save

To delete an article, example with id 5:

article = Article.find(5)

article.destroy

To add validations presence and length validations to article model for title and description:

class Article < ActiveRecord::Base

validates :title, presence: true, length: {minimum: 3, maximum: 50}

validates :description, presence: true, length: {minimum: 10, maximum: 300}

end

To find errors in article object while saving (if it's rolled back):

article.errors.any?

article.errors.full\_messages

Create New Articles from UI

In the config/routes.rb file add the following line to add all the routes for articles:

resources :articles

This will add the following routes:

routes path HTTP verb link controller#action

articles index articles GET /articles articles#index

new article new\_article GET /articles/new articles#new

create article POST /articles articles#create

edit article edit\_article GET /articles/:id articles#edit

update article PATCH /articles/:id articles#update

show article article GET /articles/:id articles#show

delete article DELETE /articles/:id articles#destroy

*Finish: 3/15/2016 at 11:45 pm (2 hours) 23% complete*

*Start: 3/17/2016 at 10:30 pm*

Complete New and Show

See Lecture 65 Complete New and Show Actions for Code used

Edit Articles

See Lecture 67 Complete New and Show Actions for Code used

List Articles and Layout Links

See Lecture 69 Complete New and Show Actions for Code used

*Finish: 3/17/2016 at 11:45 pm (1 hour and 15 min) 25% complete*

*Start: 3/19/2016 at 12:15 am*

Destroy and Partials

Partials help get rid of redundant code

See Lecture 71 Destroy and Partials for Code used

*Finish: 3/19/2016 at 12:45 pm (30 min) 26% complete*

*Start: 3/20/2016 at 11:30 pm*

Deploy and Wrap Section

Use Partial for @article = Article.find(params[:id])

Private  
 def set\_article  
 @article = Article.find(params[:id])  
 end

Then at the top we’ll call it for the variables that it was a part of

before\_action :set\_article, only: [:edit, :update, :show, :delete]

*Finish: 3/21/2016 at 12:15 am (45 min) 28% complete*

*Start: 3/21/2016 at 9:30 pm*

**Section 5: Styling for your Rails Application**

Introduction

Cacoo wireframing [charliecuddy4@gmail.com](mailto:charliecuddy4@gmail.com) Password5

Install Bootstrap

<https://github.com/twbs/bootstrap-sass>

Bootstrap sass gem page:

<https://github.com/twbs/bootstrap-sass>

Add the following gem to your gemfile above the gem 'sass-rails':

gem 'bootstrap-sass', '~> 3.3.5'

To install the gem to your app run:

bundle install --without production

Create a file called custom.css.scss under app/assets/stylesheets folder

Add the following lines to the file:

@import "bootstrap-sprockets";

@import "bootstrap";

Add the following line to your application.js file in the app/assets/javascripts folder under the line that says //= require jquery\_ujs:

//= require bootstrap-sprockets

Make a commit to your git repo

Build Homepage

Deployed to heroku <https://alpha-blog-cuddy.herokuapp.com>

*Finish: 3/21/2016 at 11:15 pm (1 hour 45 min) 30% complete*

*Start: 3/22/2016 at 10:30 pm*

Style Form Template

See Lecture 81 Style Form Template for code used

*Finish: 3/22/2016 at 11:15 pm (45 min) 30% complete*

*Start: 3/23/2016 at 9:30 pm*

Style Messages Template

Style Show View

Style List View

Deploy to Production

*Finish: 3/23/2016 at 11:30 pm (2 hours) 33% complete*

*3/25/2016 (30 min)*

added button

*Start: 3/26/2016 at 8:45 pm*

**Section 6: Associations and Authentication Systems**

Introduction to Section 6 and Associations

Create Users  
 See Lecture 92 Notes

Add User Validation

One to Many Association

*Finish: 3/27/2016 at 12:45 pm (2 hours… getting ready for easter) 35% complete*

*Start: 3/27/2016 at 10:30 am (til 11:15 and then a break) Start again: 12:30*

Show User Info in Articles

Add Secure Password

New User Sign Up

Edit Users

Show User and Profile Image

*Finish: 3/27/2016 at 2:30 pm (2 hours 45 min) 39% complete*

*Start: 3/28/2016 at 9:15 pm*

Add Users Index

Add pagination

Add Login Form

Create and Destroy Sessions

*Finish: 3/27/2016 at 11:00 pm (1 hours 45 min) 42% complete*

*Start: 3/29/2016 at 9:00 pm*

Restrict Actions from UI

Restrict Actions in Controller

Sign in upon Signup

Add Admin User Functionality

Complete Admin User

*Finish: 3/29/2016 at 12:00 am (3 hours) 46% complete*

*Start: 3/30/2016 at 9:30 pm*

Added additional buttons, adjusted color and text and deployed for user testing

*Finish: 3/30/2016 at 11:30 am (2 hours)*

**Section 7: Many-to-Many Associations and Automated Testing – Integration, Functional, Unit**

*Start: 3/31/2016 at 10:00 pm*

Introduction to Section 7

Category Model and Unit Tests

Categories Controller and Test

Integration Test and Feature

Integration Test for Invalid Category

Integration Test and Feature: Listing Categories

*Finish: 4/1/2016 at 12:15 am (2 hours and 15 min) 49% complete*

*Start: 4/3/2016 at 8:15 pm*

Integration Test and Feature

Fix List Categories

Admin User Requirement Test

Update Test and Navbar

Build Article and Category Association  
  
*Finish: 4/3/2016 at 10:30 pm (2 hours and 15 min) 53% complete*

*Start: 4/4/2016 at 10:15 pm*

Add Associations from UI

Update Views

Edit Categories

Deploy to Production

*Finish: 4/5/2016 at 12:15 am (2 hours) 55% complete*

*Start: 4/20/2016 at 9:30 pm*

**Section 8 - Stock Track and Social Media App**

Assignment 1

Assignment 2

Devise and Bootstrap

*Finish: 4/20/2016 at 11:15 am (1 hour 45 min) 57% complete*

*Start: 4/22/2016 at 11:45 pm*

Stock Model

Stock Price Methods

*Finish: 4/23/2016 at 1:15 am (1 hour 30 min) 59% complete*

*Start: 4/27/2016 at 11:15 pm*

UI for Stocks

Form Submission and Ajax

Spinner and more Ajax

*Finish: 4/28/2016 at 12:30 am (1 hour 15 min) 62% complete*

*Start: 4/28/2016 at 11:10 pm*

Many to Many Association

Stock Restrictions

*Finish: 4/29/2016 at 12:10 am (1 hour) 63% complete*

*Start: 5/1/2016 at 3:15 pm*

Listing Stocks

User Details

*Finish: 5/1/2016 at 4:15 pm (1 hour) 64% complete*

*Start: 5/1/2016 at 10:30 pm*

UI Views for User Updates

Adding Friends

*Finish: 5/2/2016 at 12:30 am (2 hour) 66% complete*

*Start: 5/2/2016 at 10:15 pm*

Friendship Views

Search Friends from UI

Ajax for Search

Search Methods

Show and Remove Friend

*Finish: 5/2/2016 at 12:45 am (2 hours 30 minutes) 69% complete*

**Section 9 – Email, Custom Payment Functionality and File Uploads**

*Start: 5/4/2016 at 11:05 pm*

Start Photo App

Setup Authentication

Sending Email in Production

*Finish: 5/5/2016 at 12:50 am (1 hour 45 minutes) 72% complete*

*Start: 5/5/2016 at 9:30 pm*

Update Layout and Test Email in Production

Build Homepage

*Finish: 5/6/2016 at 12:45 am (3 hour 15 minutes) 73% complete*

*Start: 5/9/2016 at 10:15 pm*

Stripe and Payment Introduction

Payment Model

*Finish: 5/9/2016 at 11:45 am (1 hour 30 minutes) 76% complete*

*Start: 5/10/2016 at 11:00 pm*

Extend Devise Registrations Controller

Fix Conflict

*Finish: 5/9/2016 at 12:45 am (1 hour 45 minutes) 77% complete*

*Start: 5/12/2016 at 10:15 pm*

Image Upload

Image Size Validations

Complete Prod Image Upload

Final Student Projects

*Finish: 5/13/2016 at 12:15 am (2 hours) 81% complete*