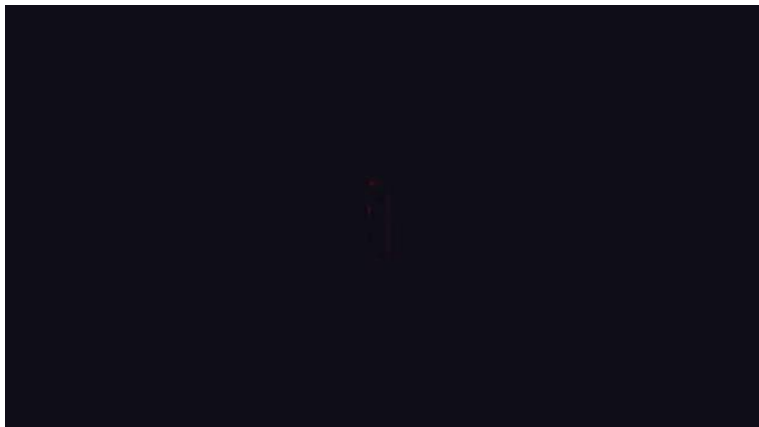


Ruby on Rails, a full stack MVC webframe where famous for powering some of the most successful internet startups of the last 20 years. It was created in 2004.



by David Hanemeyer-Hanson to give developers a way to build large-scale web applications quickly with the Ruby programming language. It exploded in popularity among the Silicon Valley tech...



and help launch Unicorn startups like Airbnb, Shopify, and GitHub just to name a few. Its underlying architecture is based on the Model View Controller Pattern. A model contains...



the business logic for your database. Rails has a built-in ORM called ActiveRecord that can easily model relationships in a SQL database. On the other end, we have the view for presentation



MODEL

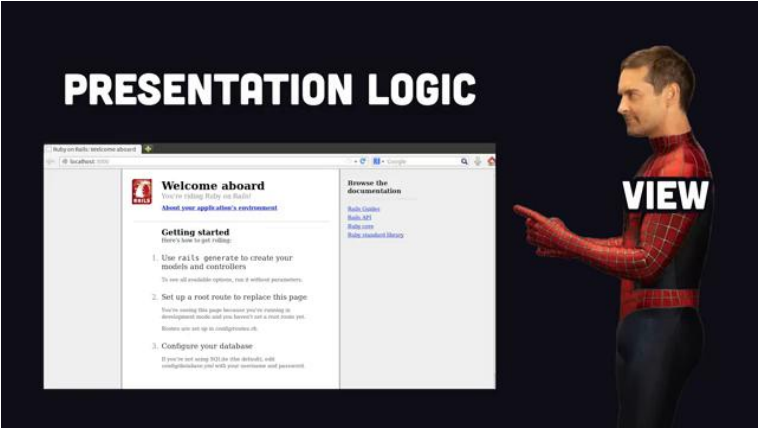
 **DATABASE STUFF**

```
app > models >  human.rb
class Human
  include ActiveRecord::API

  attr_accessor :name
  validates :name, presence: true

  def sleep
    "Zzz... dreaming of electric sheep."
  end
end
```

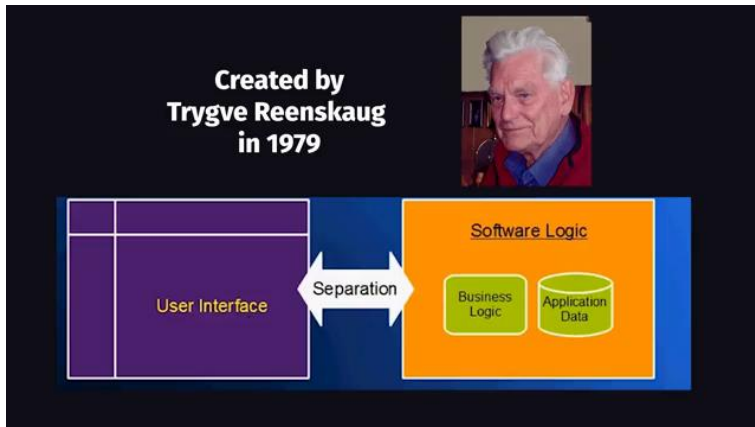
logic, which includes a mix of HTML, CSS, and embedded Ruby, which displays dynamic data queried from the back end. This data is made available by the controller, which sits in the middle.



It takes the incoming HTTP request from the user, the fetched data from the model, and finally responds with HTML or JSON. This pattern actually predates the web itself, but Rails makes it...



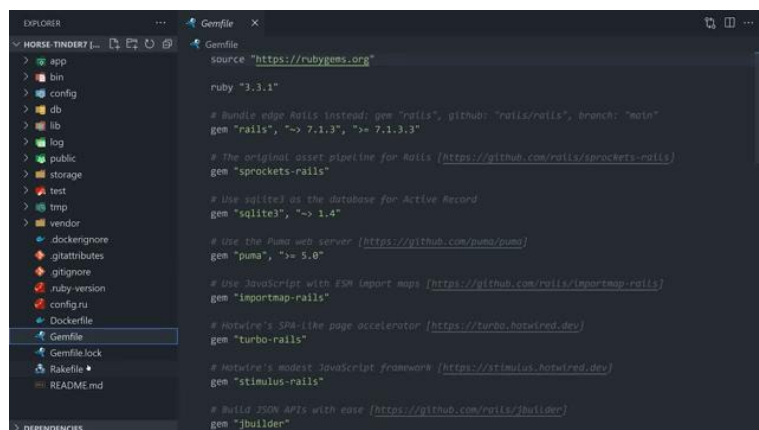
extremely easy to implement by following a highly opinionated and controversial doctrine, which includes 9 commandments like convention over configuration and exult beautiful code. Let's build a full



stack application right now by installing Ruby Rails and SQLite. Now run the Rails New command to scaffold out a new project. From here you'll notice a gem file to install dependencies in Ruby.



a rake file, which is like a make file in C to run a task, but the bulk of your code will be written in the app directory. In addition to models, views and controllers, you'll also notice mailers for transactions



The screenshot shows a code editor with a file explorer on the left and a Gemfile in the main editor. The file explorer lists various directories and files, including 'app', 'bin', 'config', 'db', 'lib', 'log', 'public', 'storage', 'tmp', 'vendor', '.dockerignore', '.gitattributes', '.gitignore', 'ruby-version', 'config.ru', 'Dockerfile', 'Gemfile', 'Gemfile.lock', 'Rakefile', and 'README.md'. The Gemfile in the main editor contains the following content:

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

ruby "3.3.1"

# Bundle edge Rails instead: gem "rails", github: "rails/rails", branch: "main"
gem "rails", "~> 7.1.3", ">= 7.1.3.3"

# The original asset pipeline for Rails [https://github.com/rails/sprockets-rails]
gem "sprockets-rails"

# Use SQLite3 as the database for Active Record
gem "sqlite3", "~> 1.4"

# Use the Puma web server [https://github.com/puma/puma]
gem "puma", ">= 5.0"

# Use JavaScript with ESM import maps [https://github.com/rails/importmap-rails]
gem "importmap-rails"

# Hotwire's SPA-like page accelerator [https://turbo.hotwired.dev]
gem "turbo-rails"

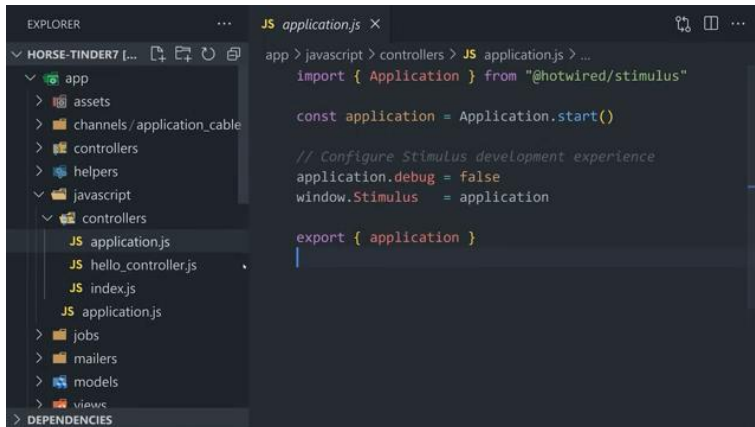
# Hotwire's modest JavaScript framework [https://stimulus.hotwired.dev]
gem "stimulus-rails"

# Build JSON APIs with ease [https://github.com/rails/jbuilder]
gem "jbuilder"
```

email, it jobs for background work, and channels for real-time connections via WebSockets. And we have a JavaScript directory configured with HotWire allowing you to build a highly interactive front-end.



without the unnecessary complexity of a JavaScript component framework. Rails also has an extremely powerful CLI. Generate will build your models using controllers, or do everything all at once.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the file structure of a project named 'HORSE-TINDER7'. The structure includes folders for 'assets', 'channels/application_cable', 'controllers', 'helpers', 'javascript', 'jobs', 'mailers', 'models', and 'views'. The 'javascript' folder is expanded, showing a sub-folder 'controllers' which contains three files: 'application.js', 'hello_controller.js', and 'index.js'. The 'application.js' file is selected. The main editor area on the right shows the content of 'application.js'. The code imports 'Application' from '@hotwired/stimulus', starts the application, configures the debug mode to false, and assigns the application to 'window.Stimulus'. The code is as follows:

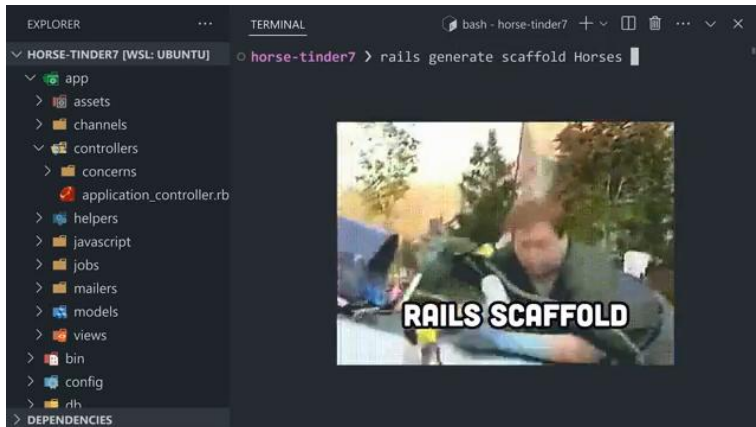
```
app > javascript > controllers > JS application.js > ...
import { Application } from "@hotwired/stimulus"

const application = Application.start()

// Configure Stimulus development experience
application.debug = false
window.Stimulus = application

export { application }
```

with scaffold. We give our model a name, which is plural by convention, then provide some fields and types we might use in the database. This command creates a bunch of files, but let's go ahead and run rake to migrate the



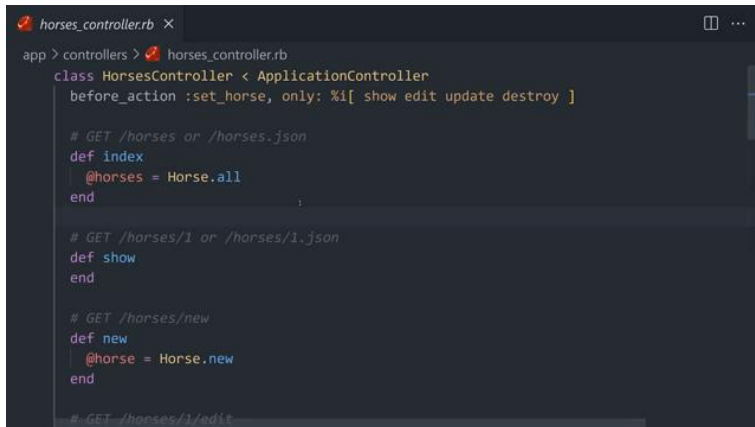
database, then rail server to run it locally. Congratulations, you just built a piece of crud. A full stack app that can create, read, update, and delete your database model in the browser. How did we do that?



so fast. At first, the routes are B file mapped a set of URLs to our controller. Automatically created by this resources method. When a user navigates to one of those routes, it triggers a controller action.



Like index, we'll fetch all the records from the database by calling the all method, which is provided automatically by the model. We could also perform data validations here, and do joins with other models.



```
horses_controller.rb X
app > controllers > horses_controller.rb
class HorsesController < ApplicationController
  before_action :set_horse, only: [:show, :edit, :update, :destroy]

  # GET /horses or /horses.json
  def index
    @horses = Horse.all
  end

  # GET /horses/1 or /horses/1.json
  def show
  end

  # GET /horses/new
  def new
    @horse = Horse.new
  end

  # GET /horses/1/edit
```

Now the controller created an instance variable, which we can access directly in the HTML.inbeddedRubyFile. Here we have each rune the database as a Ruby object, which we can loop over with e.

```
EXPLORER
HORSE-TINDER7 [WSL: UBUNTU]
  app
  channels
  controllers
    application_controller.rb
    horses_controller.rb
  concerns
  helpers
  javascript
  jobs
  mailers
  models
    application_record.rb
    horse.rb
  views
  DEPENDENCIES

horses_controller.rb
app > controllers > horses_controller.rb

class HorsesController < ApplicationController
  before_action :set_horse, only: %i[ show edit update destroy]

  # GET /horses or /horses.json
  def index
    @horses = Horse.all
  end

  # GET /horses/1 or /horses/1.json
  def show
  end

  # GET /horses/new
  def new
    @horse = Horse.new
  end

  # GET /horses/1/edit
```


to display it in the front end UI. This has been Rails in 100 seconds, but if you really want to learn how to build cool web apps, it's far more important to learn how to problem solve like a programmer. You can see

A screenshot of a code editor window titled 'index.html.erb'. The editor shows the following code:

```
app > views > horses > index.html.erb
<p style="color: green"><%= notice %></p>

<h1>Horses</h1>

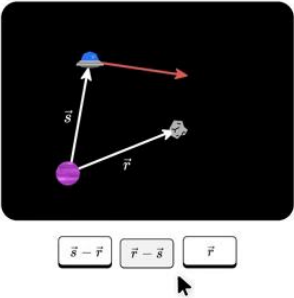
<div id="horses">
  <% @horses.each do |horse| %>
    <%= render horse %>
    <p>
      <%= link_to "Show this horse", horse %>
    </p>
  <% end %>
</div>

<%= link_to "New horse", new_horse_path %>
```

start making that happen for free thanks to this video sponsor, Brilliant. WebRamerix coming to go, but problem solving is a skill that you keep forever. Brilliant's platform will introduce you to essential programming.

Subtracting Vectors

Explore how to move from one vector's head to another when the tails are in the same position.



programming concepts, but most importantly, the hands-on exercises will develop your brain to recognize and solve complex problems that developers need to overcome on a daily basis.

Fly the Hypotenuse

Let's see how the **Pythagorean theorem** is related to the **shortest distance** between two points.

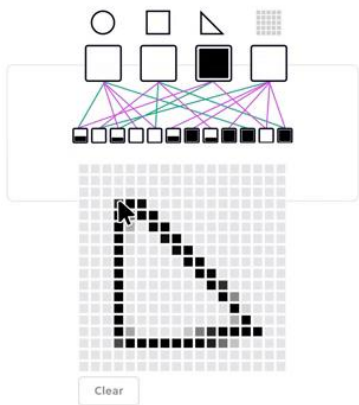
Distance: $\sqrt{89}$ km



Bike

Drone

of all, every lesson is concisely rewarding, but by investing just a few minutes each day, you'll develop habits that can level up your programming skills for the rest of your life. And you can do it anywhere, even...



What do you
neurons whe
prediction?

☐ Exactly 1

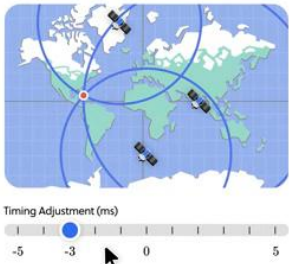
☐ One or more

Submit

your phone. To try everything Brilliant has to offer for free for 30 days, visit brilliant.org slash firehack or scan this QR code for 20% off their premium annual subscription.

Timing Corrections

Correct your phone's timing error to find your location.



Thanks for watching and I will see you in the next one.

