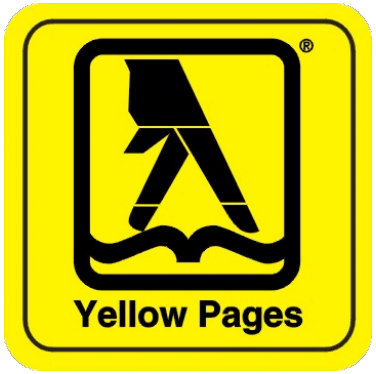
In this lecture, we will discuss…

- ✧ Brief Rails history
- ✧ Benefits of using Rails
- ✧ Model View Controller

History - Ruby on Rails (RoR)

- Framework for making **dynamic web applications**
- Created in 2004 - 2005 by David Heinemeier Hansson

Who is Using Rails?



Why Use Rails?

- **C**onvention **O**ver **C**onfiguration (COC)
  - Less code to write
    - Some code Rails automatically generates for you
    - Oftentimes, there is no need to write code at all
  - Learn it once - know what to expect the next time

Why Use Rails?

- ✧ Database Abstraction Layer
  - No need to deal with low-level DB details
  - No more SQL (*Almost*)
  - Important to understand the SQL generated!

Why Use Rails?

- ✧ Agile-friendly
- ✧ **D**on’t **R**epeat **Y**ourself (DRY) principle
- ✧ Cross-platform
- ✧ Open Source
- ✧ Modular

SQLite

- Rails uses SQLite for database by default
  - Self-contained, serverless, zero-configuration, transactional, relational SQL database engine.

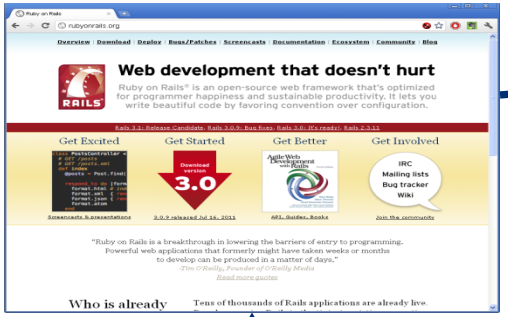
CLAIM: Most widely deployed SQL database engine in the world

MVC: Model View Controller

- Invented in 1979 by Trygve Reenskaug
- Well-established software pattern used by many web and desktop frameworks
- ***Model*** - represents the data the application is working with (and possibly business logic)
- ***View*** – (visual) representation of that data
- ***Controller*** - orchestrates interaction between the model and the view

MVC Cycle

1. Request sent
2. Controller ↔ Model
3. Controller invokes View
4. View renders data



Controller

View

Model

DB

Summary

- ✧ Rails is very good for **Rapid Prototyping**
- ✧ MVC and Convention over Configuration enable you to **“think less and do more”**

What’s Next

- ✧ Creating your first Rails app