Curriculum draft

Web Application Development

Preparation (2-3 weeks)

Goal: Every student is ready to start the in-presence course with an understanding of how static web pages are created and what a web application is composed of.
This includes exercises on HTML/CSS and JavaScript and frontend frameworks like Bootstrap, as well as a first exposure to Ruby and Ruby on Rails applications.

Along the exercises, students start using the command line for basic commands and complete a basic setup of their development environment (code editor, version control, GitHub account). Assistance via email and Skype is available throughout the preparation phase.

KICKOFF WEEK

Week 1: Introduction + object-oriented design - basic algorithms, basic (and intermediate) Ruby and implementation in Ruby on Rails and Sinatra

Goal: Every student will have created a fully working web application at the end of week 1, including frontend design, backend code, server deployment, third-party API integration and a database connection.

Students will also familiarize themselves with the development environment, research the right implementation of specific features and be well versed in version control/ GitHub.

PHASE 1

Week 2: Ruby + web integration – HTTP, JSON & XML, third-party API integration, Web Scraping, REST

Goal: Students achieve a solid understanding of how web applications tie together backend code and the overall web infrastructure by continuing to write small web applications.

Week 3: Rails I – ActiveRecord, MVC, basic routing, Rails Forms, authentication (OAuth)

Goal: Students solve more complex and specific requirements and problems using the Ruby on Rails framework.

Weeks 2 and 3 build the basis for the advanced topics beginning in week 4. Depending on pace and content, central concepts of Rails like MVC and routing are introduced and applied using web integration and third-party data sources from week 2 on.

PHASE 2

Week 4: Test-driven development + JavaScript – jQuery, AJAX, Event-driven programming

Goal: Students switch to an efficient and effective work flow based on test-driven development. Additionally, client-side scripting is introduced, with JavaScript and jQuery used to build example projects.

Week 5: Rails II – Request-Response Lifecycle, Cookies + sessions, Capybara, asset pipeline

Goal: Students move on to advanced application features, while applying previously learned concepts.

PHASE 3

Week 6: Advanced Rails + SQL – Basic Querying, Complex Joins, Indexing

Goal: Students continue adding features to the applications being developed, and understand the way in which the Ruby on Rails framework extends the Ruby programming language and supports the development process of a web application. Database features are introduced as part of the continuing work on class projects.

Week 7: Advanced JavaScript + Rails API – Closures, Client-side MVC (advanced JSON Construction, single page web applications)

Goal: Students familiarize themselves with advanced concepts of Ruby on Rails application development and start developing (single page) web applications.

LAUNCHPAD

Week 8: Class Project I

Goal: Students apply previously learned material to work on a larger class project. Advanced concepts (e.g. HAML/ SCSS, CoffeeScript for frontend development; PostgreSQL as a database technology) are introduced as applicable.

Week 9: Class Project II, security, professional development

Goal: Students deepen the scope of their work and gain an understanding of the context in which their web applications can be deployed with a focus on security. Also includes preparation for hiring interviews and guest lectures focusing on professional development both as an employed and freelance developer.

PERSONAL PROJECT

Week 10-11: Finishing up + additional class projects

Goal: Students combine all previously learned concepts through a class assignment.

--