Team 2 Ruby on Rails Final Project

By Camryn Friedman, Ben Pearlstein, Matthew Rosivack, Aishwarya Arun, and Yifan Zhou

Initial Plan

- Alter web scraper to output the data we want
  - Put Autumn classes into a seed.rb file to populate our database
- Implement the interfaces
  - Change Rails scaffolding to make our application more professional
- Authentication via Devise
- Incorporate JavaScript to enhance user experience
- Work on presentation demonstrating our project

Additional Packages, Gems, and Extensions Used

- Devise gem

    - User authentication

- Mechanize gem

- SQLite Extensions in VSCode

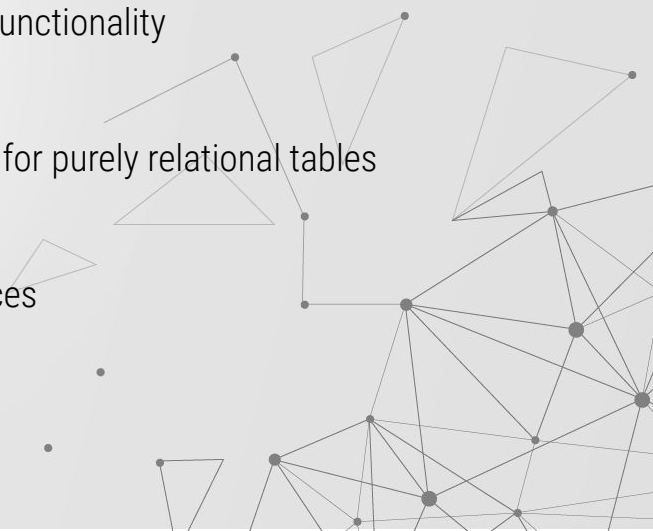   - Used to view database in comprehensive manner

Model Choices

- User

- Course

- Degree

- Course_Mapping

  - Maps between a degree and its courses

- Degree_In_Progress

  - Maps between a users and the degree(s) they are working towards

- Finished_Courses

  - Maps between a user and the courses that they have completed

Roadblocks & Solutions

1. Generally getting familiar with Rails and how each folder relates to one another

   a. **Solution:** This become easier once we started to program more in Rails

2. Making changes to our webscraper in order to correctly initialize the database

   a. **Solution:** We figured out the functionality of the seeds.rb file in the db folder

3. Getting familiar with Devise (sign out page caused issues at first)

   a. **Solution:** In class example greatly helped in figuring out functionality

4. Changing the database through updates made on the website

   a. Creating new instances to put in the database, especially for purely relational tables (course_mapping, degree_in_progress, finished_courses)

   b. **Solution:** Object.create(attributes) in the appropriate places

5. Passing object attributes between views

   a. **Solution:** passing parameters via link_to path

Key Features

- Student account vs. Admin account
  - Can create a new student account but not a Admin account
- Student Interface vs Admin Interface
  - Student: can view their incomplete courses and update them to complete
  - Admin: can view reports of students' completed courses and add required courses to the degree
- Login via Devise
  - Automatically updates the user database with new logins

Client-View Demo

Plans for Improvement

- Styling
  - Improve tables, fonts, and links
- Confirmation
  - Get confirmation before changing the database
- Ability for admin to remove courses from a degree if it is no longer required
- Implement JavaScript
  - Autocomplete when typing in courses
- Add additional features
  - Show courses not in major