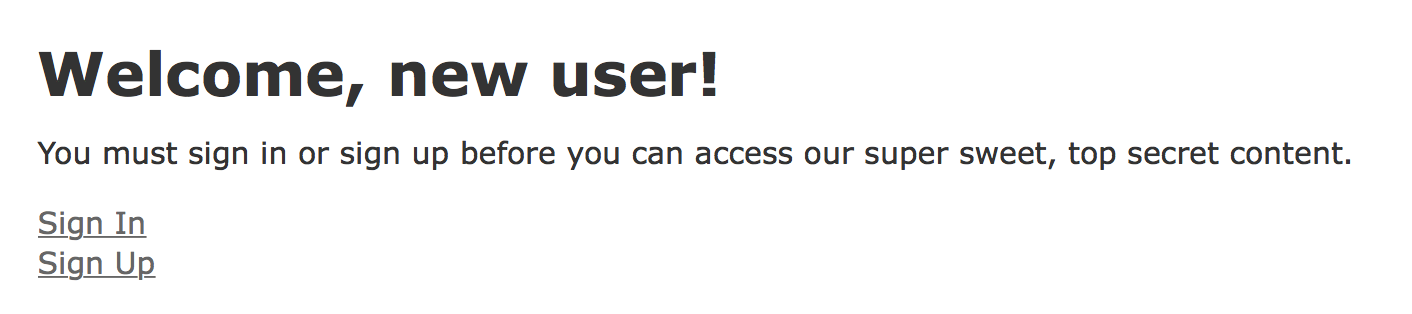
General description of the app:

Users can login and view that they are on the list. They can view programs and courses. If they see anything not on the list they will contact the admin. The admin can add, update, and delete.

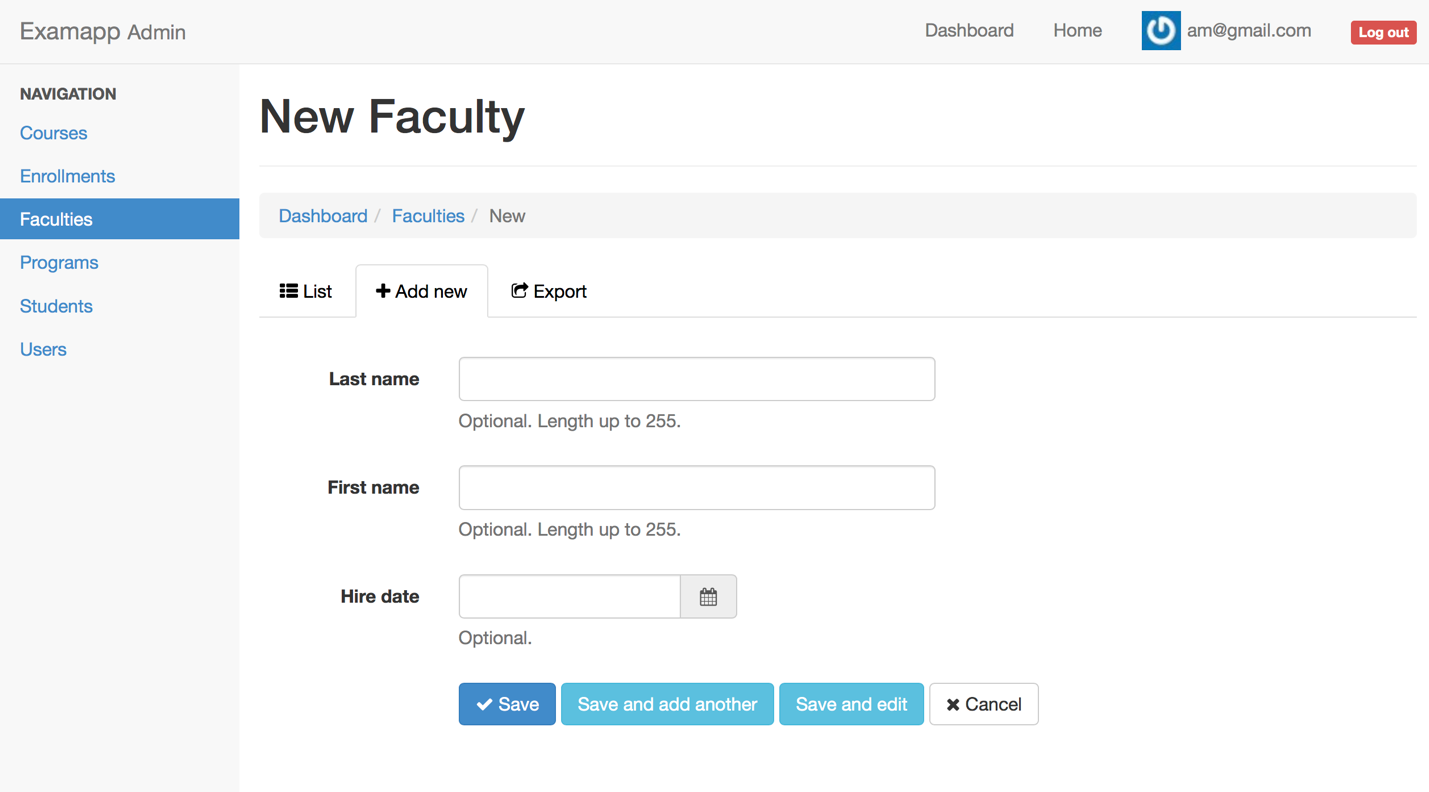
Github: https://github.com/arimunoz/examapp.git

Exam ScreenShot of Project:

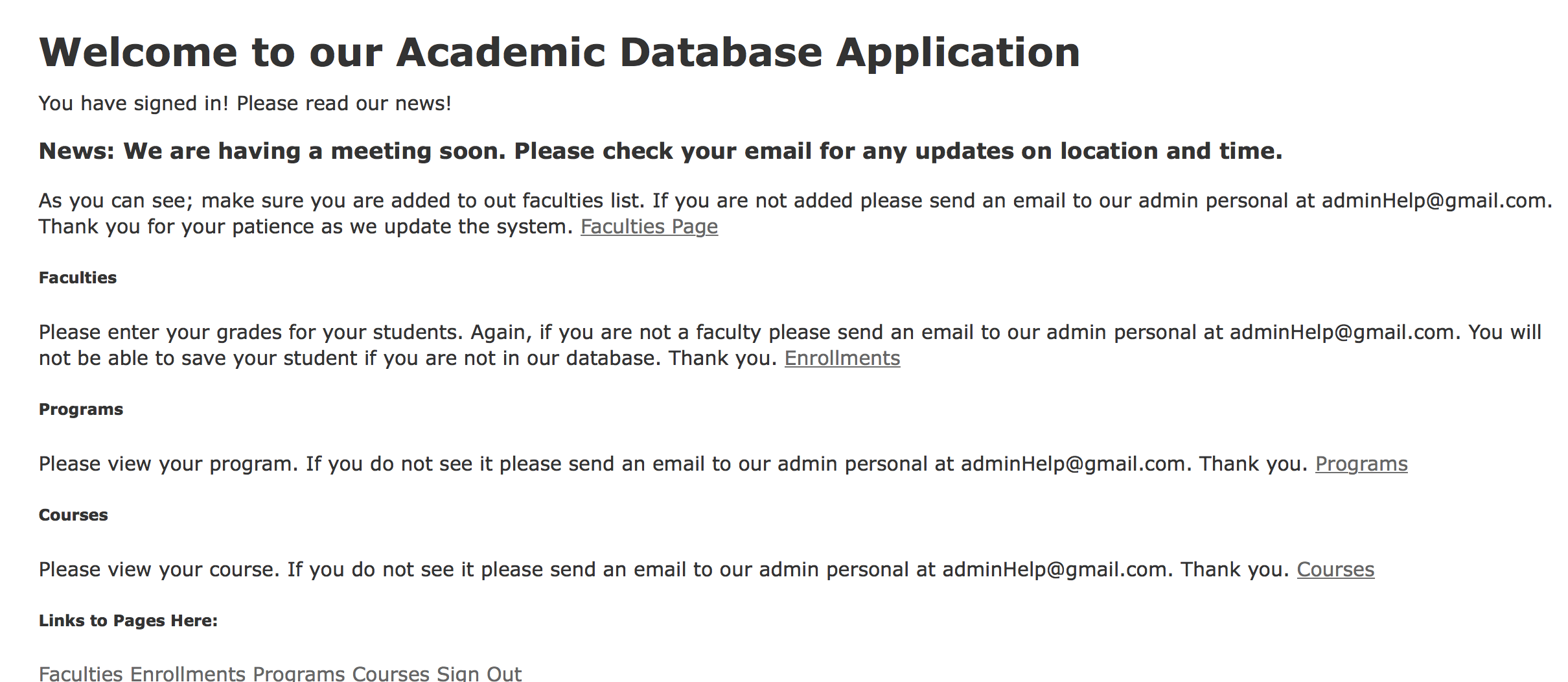
Login/Signup Page:



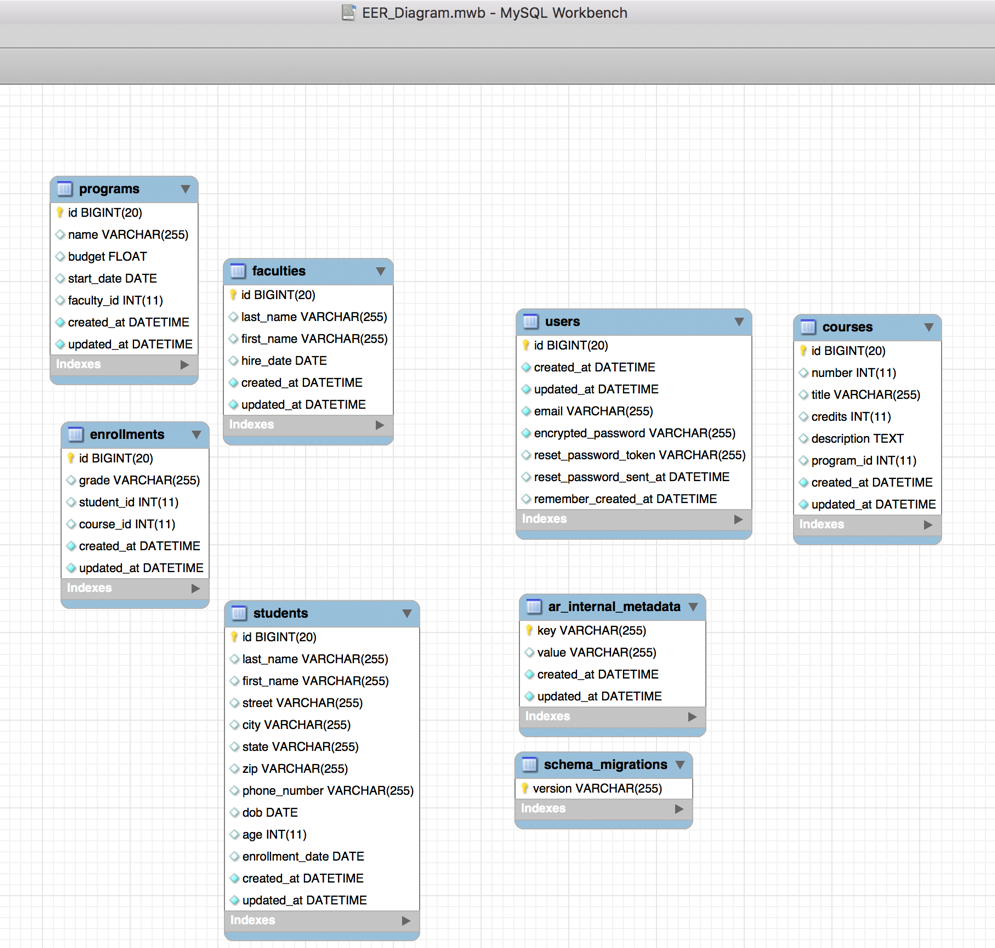
Login Admin: Note! If you type in localhost:3000/admin it will go straight to admin. In admin Page you can update, edit tables. There is a home tab that takes you to a regular user view.



Home Page: Type in localhost:3000. It will make you sign in and then route you to home page. You can click on links and it will take you to different pages. You can only view data in some pages but the user can add data to enrollments page.



E/R diagram from mysql workbench:



The student should go through the following steps in completing this project:

1. Design the database, following an E/R approach; then go through the normalization process to come up with a collection of tables that are in (3NF) third normal forms.

2. Describe some realistic constraints such as primary keys, foreign keys, check constraints, and not null constraints, for the tables and attributes, etc.

A: Constraints would be if a certain faculty is not there then program can’t be created because it is assigned to a faculty. Same with enrollments; it needs a faculty to be there for it to go through.

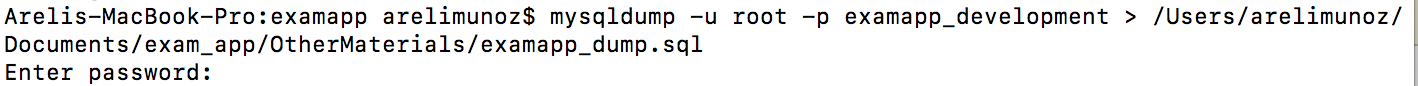
This was all tested before I changed the outlook where the users can only view the program and faculty.

When the admin creates a faculty; it has a primary key and that communicates to program and course.

3. Use MySQL to create the normalized tables (ERD).

The student has to come up with a list of print-outs of the table structure and the content, similar to those that were created in the class demos. (Could use mysql-dump)

Did mysql dump and it is in OtherMaterials folder.



5. Populate the database with at least 15 records by using the seed.rb file or the faker gem in your application.

Check: It is in seeds.rb. Here is the records:

Faculty.create!(last\_name: 'Neet', first\_name: 'Mary', hire\_date: Date.new(1998,1,15))

Faculty.create!(last\_name: 'Blue', first\_name: 'Kreg', hire\_date: Date.new(1990,2,5))

Faculty.create!(last\_name: 'James', first\_name: 'Ted', hire\_date: Date.new(1995,10,12))

Student.create!(last\_name: 'Boy', first\_name: 'Dann', street: '78654', city: 'Chicago', state: 'IL', zip: '09753', phone\_number: '708-345-2834', dob: Date.new(1998,2,20), age: 20, enrollment\_date: Date.new(2008,1,12), )

Student.create!(last\_name: 'Dader', first\_name: 'Tom', street: '36456', city: 'Melrose', state: 'IL', zip: '38495', phone\_number: '208-395-3899', dob: Date.new(1990,5,20), age: 28, enrollment\_date: Date.new(2016,9,12), )

Student.create!(last\_name: 'Olts', first\_name: 'Susy', street: '23476', city: 'Bellwood', state: 'IL', zip: '9843', phone\_number: '708-345-2312', dob: Date.new(1998,9,25), age: 20, enrollment\_date: Date.new(2018,8,12), )

Program.create!(name: 'AU NightCoders', budget: 1250, start\_date: Date.new(2002,1,1), faculty\_id: Faculty.last.id)

Program.create!(name: 'AU Jam', budget: 2000, start\_date: Date.new(2005,1,1), faculty\_id: Faculty.last.id)

Program.create!(name: 'AU Art', budget: 1250, start\_date: Date.new(2010,1,1), faculty\_id: Faculty.last.id)

Course.create!(number: 5454, title: 'Code Jam', credits: 3, description: 'Place where you code and jam!' , program\_id: Program.last.id)

Course.create!(number: 5454, title: 'Art', credits: 3, description: 'Place where you are creative!' , program\_id: Program.last.id)

Course.create!(number: 5454, title: 'Math', credits: 3, description: 'Place where you think hard!' , program\_id: Program.last.id)

Enrollment.create!(grade: 'A', student\_id: Student.last.id, course\_id: Course.last.id)

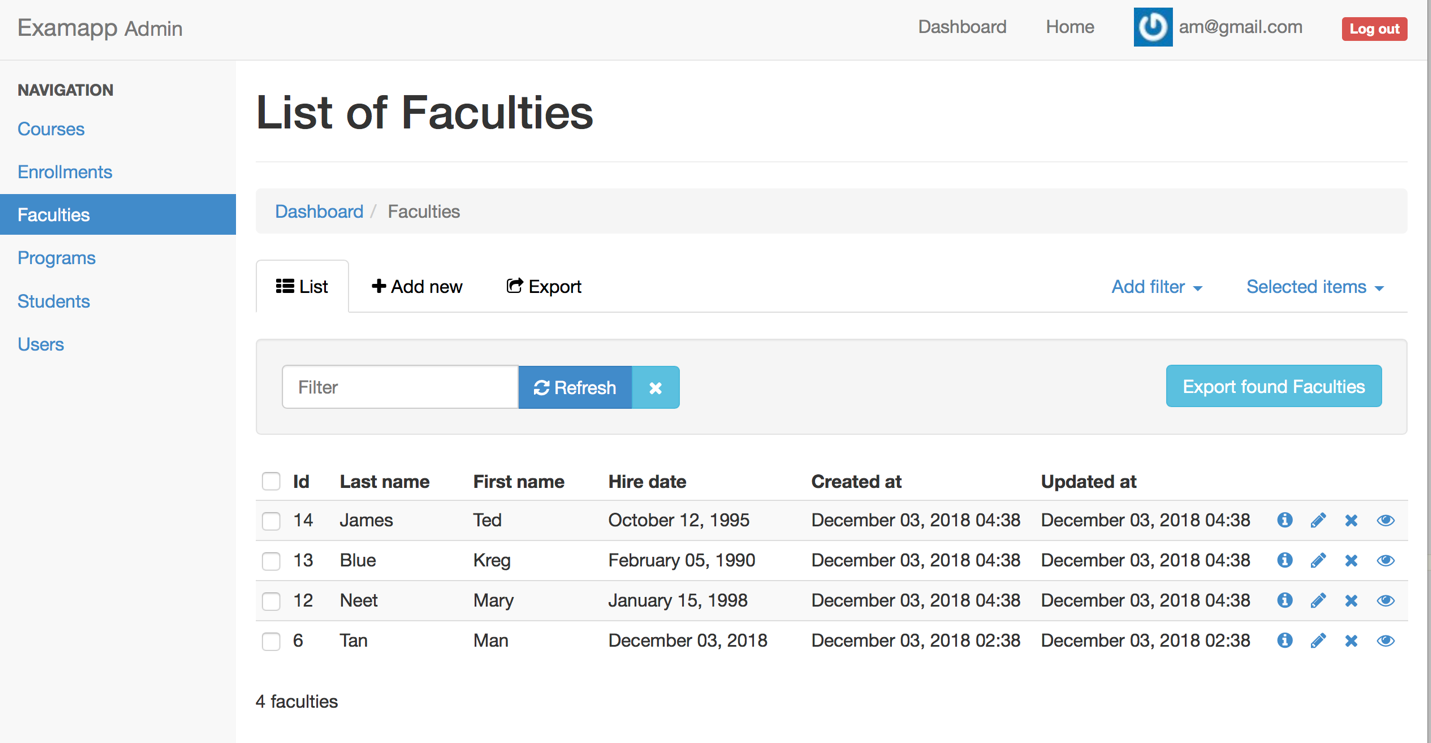
Enrollment.create!(grade: 'B', student\_id: Student.last.id, course\_id: Course.last.id)

Enrollment.create!(grade: 'C', student\_id: Student.last.id, course\_id: Course.last.id)

6. Write user interface, with embedded ruby on rails script(s):- the Information page should contain something like general announcements, news information, etc..

7. The administrative page should include such functions as add/delete/update faculty, form/dismiss/update committees, as well as add/delete/update programs, etc... (NB. Use devise and real-admin gems)

It does:



Submit:

Answers to the above questions plus; the general description of the app, the database design, interface design, the source code (a link to your github), as well as the database tables, and any other supporting material.