

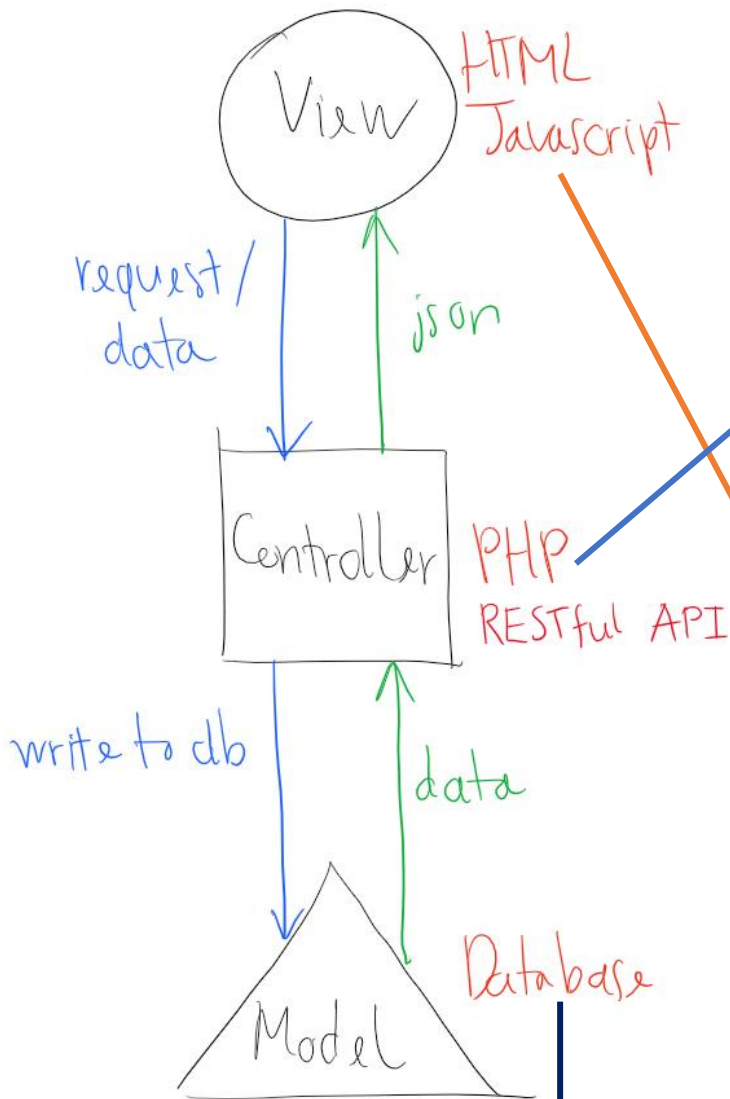
EASTERN WASHINGTON UNIVERSITY
ADVANCEMENT PROGRAMMING EXAM
WEBSITE

PROJECT OVERVIEW

Senior Capstone Class 2017

Instructor: Tom Capaul

Team: Tu Nguyen, Aaron Griffis, Andy Robinson



▲ APE

- .vscode
- account
- ape
- auth
- category
- cron_job
- grade
- homepage
- location
- settings
- sso
- **util**

▲ view

- category
- css
- exam
- grade
- home
- img
- includes
- js
- location
- settings
- user

🐘 index.php

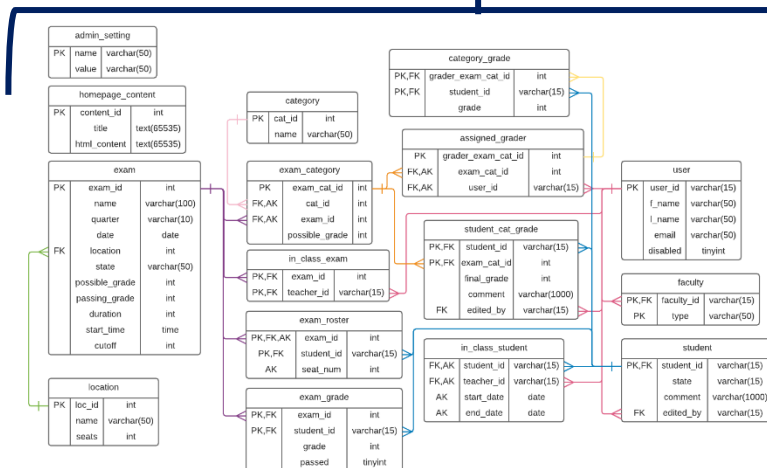
📄 README.md

🐘 tabbedTable.php

🐘 table.php

🐘 pdoconfig.php

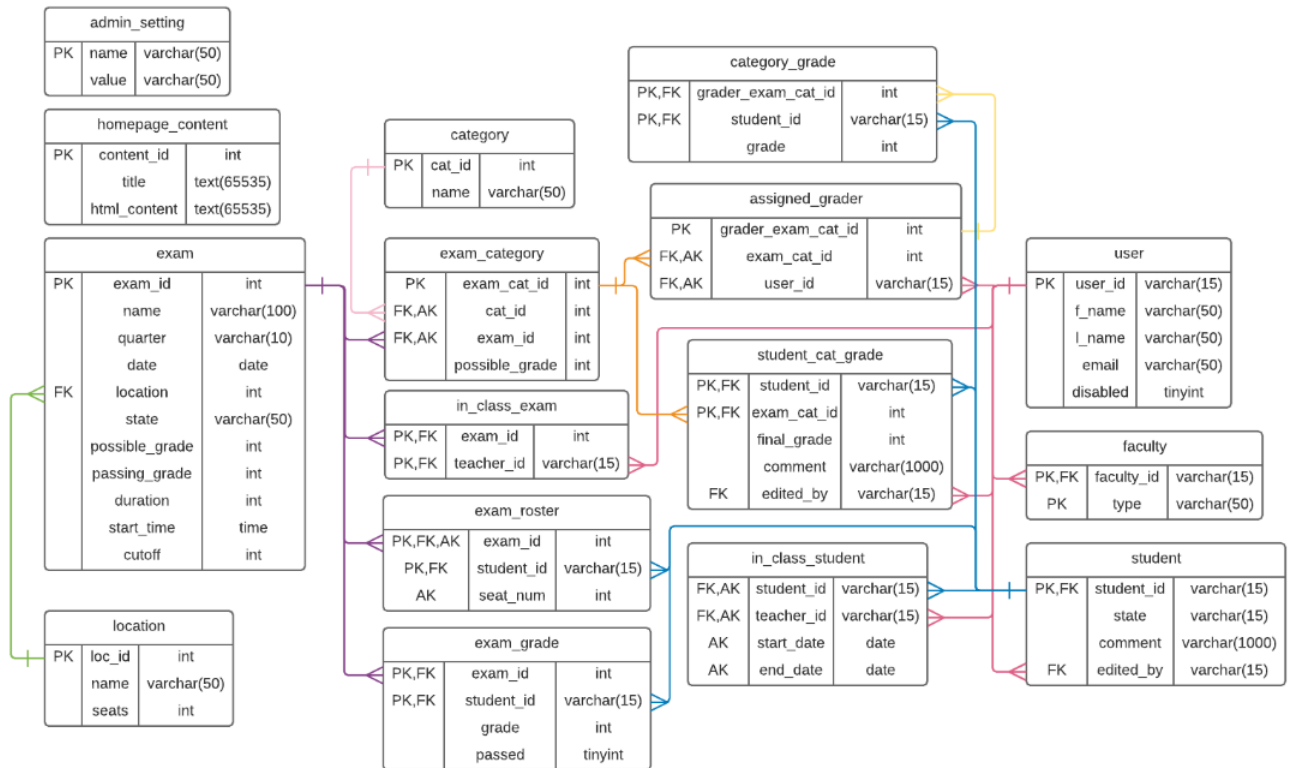
M



Model

The database. Learn more in [APE Database Documentation](#).

/pdoconfig.php: configurations for accessing database. Uses PHP PDO.



View

Uses:

- jQuery
- Bootstrap
- Bootstrap Notify
- Summernote editor

Views are designed according to website components: Exam, User, Category, Location, Grade. There are no separate pages for each user type. Instead, the page index.php uses user type info to render a correct view. Additionally, the page's JavaScript is used to hide or show elements depends on user type. For example, User page for Admin and Student page for Teacher is rendered by /user/index.php but User page for Admin has all account type tabs while Student page for Teacher has only Student tab and import function.

All pages, except Category and Location, need a "page" URL param to indicate what view to render according to the user type.

For example:

`http://localhost/ape/view/user/?page=admin_user`

`http://localhost/ape/view/user/?page=teacher_user`

A User page is rendered follow this flow:

URL -> /view/user/index.php -> /view/index.php

/view/user/index.php: checks user type to add/remove user tabs

/view/index.php: master index file that loads all CSS and JavaScript, plus rendering the info from /view/user/index.php

Bootstrap table and tabbed table are refactored to /view/table.php and /view/tabbedTable.php in order to reduce duplicate code.

Master components

Frequently used components are refactored to master files to reduce duplicate code and for easy code maintenance.

/view/index.php: master index file that loads all CSS and JavaScript, plus rendering the info from individual index.php

/view/js/script.js: master JavaScript that has frequently used functions

/view/table.php: renders a table

/view/tabbedTable.php: renders a tabbed table

/view/js/build_table.js: function that builds data rows to a table

/view/css/custom.css: master CSS file applied site-wide

/view/includes/tabbedModal.php: renders a tabbed modal

Modal

Bootstrap modal is heavily used to display detail information of an item. There are regular modal and tabbed modal. Tabbed modal is used in Exam page. Each modal is its own HTML file within the view directory.

Learn More

To learn more about the view structure, see in-line code comments and examples.

Controller – API

The REST API is the heart of the website that functions as the interface between the database and the view. The view makes an Ajax call to the API, the API processes it, then returns in JSON format.

Learn more about each API endpoint in [APE API Documentation](#).

Security

API Back-end Security

To call an API endpoint, a client must have at least these three pieces of information:

- User ID
- User type
- User session id

Each API endpoint has its own allowed type(s) to prevent unauthorized actions. For example, `create_ape.php` has allowed types of [“Admin”, “Teacher”] so that only admin and teacher accounts can create an exam. If a student tries to create an exam, the endpoint will return a 401 HTTP code.

These are the steps of authorizing and verifying for each API call:

1. Authorize user
 - a. Check user exists
 - b. Check user id matches user type
 - c. Check session id matched
 - d. Check user type against the endpoint allowed types
2. Sanitize sent-in data (remove HTML tags, script tags)
3. Validate sent-in data in correct format (text, number, etc.)

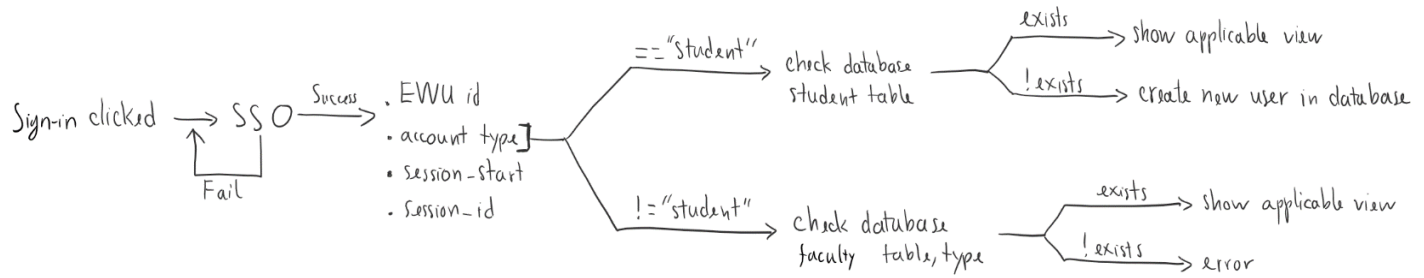
Special Accounts

User Id	User Type	Purpose
999999	System	Used for internal calls between API functions if necessary.
000	000	Used solely for getting open exams to show in public homepage when no user is signed in.

Front-end Security

All input fields are validated upon submitting.

Sign-in Flow



When a student signs in for the first time, the website automatically creates an account for the student.

An Admin or Teacher can create a student account beforehand.

An Admin or Teacher or Grader account must be created before the this user signs in, otherwise the website will throw error.

Team Contact Information

Tu Nguyen (Lead): hoangtu@live.com

Aaron Griffis: aarongriffis@eagles.ewu.edu

Andy Robinson: ydnabyrd@eagles.ewu.edu