Client Wishlist and Requirements

Site-wide Experience

- Static landing page
    - Informational page about the site

Instructor Experience

- Register an account
    - Name (first and last)
    - Email Address (verify .edu, maybe?)
- Sign into account
    - Email/Password
    - MFA (?, see additional ideas)
    - Password reset ability
- Sign out
- Experience
    - Create courses
    - Load students to courses or send code to allow students to 'enroll' in course
    - Create teams
    - Create reviews
        - CRUD questions for the review
        - Question types: Likert, multiple choice, or short answer
    - Schedule reviews
    - Review reviews
        - Private review
        - Public review
    - Reports
        - Student average (converted to 100pt scale)
            - Sortable
        - Class metrics

Student Experience

- Register an account
    - Name
    - Email Address
    - Contact details
        - Mobile
        - Discord
        - Teams
- Sign into account
    - Email/Password
    - MFA (?, see additional ideas)
    - Password reset ability
- Sign out

- Experience
  - Complete pending reviews
  - Reviews should have both public and private feedback options
  - Review score
  - Review public feedback from peers
  - Reports
    - Combined feedback
    - Overall scores

Constraints

- Email addresses must be valid
- Passwords must be complex NIST compliance
- Data must be stored in relational database
- Must use technology covered in class
  - HTML
  - CSS
    - CSS libraries are encouraged including Bootstrap or Tailwind
  - Javascript
    - jQuery is okay, but not preferred
    - Frontend frameworks are allowed but only if we get to that portion of the material in class
    -
  - Backend must use nodeJS and express
- Code should be organized similar to the charts example
- Must be a single page application (SPA)
- Must include manifest for 'Add to Home Screen' capability
- Must use Github repository w/ branching
- Must meet accessibility standards
- Passwords can NOT be stored in plain text in database
- Secure backend routing
- Must be responsively designed to look appropriate on mobile or desktop devices while being mobile first.

Additional ideas

- Host the application on GCP
  - Custom URL or DuckDNS URL
  - HTTPS
- CI/CD Pipelines
  - Automated testing with Github Actions
  - Webhook integration for automatic deployments
- Multi-factor authentication
  - Using a TOTP or email or SMS

- - Passport.js may help
- Integrate with MariaDB or Postgres
- Use of JS libraries to improve professionalism of application
- Include an automated application walk through (IntroJS)

Prohibitions

- Although Generative AI is allowed (and encouraged), you may NOT use it to develop a full solution.
- Existing content management systems (Wordpress/Drupal/Joomla/etc) are strongly prohibited