Peergrader

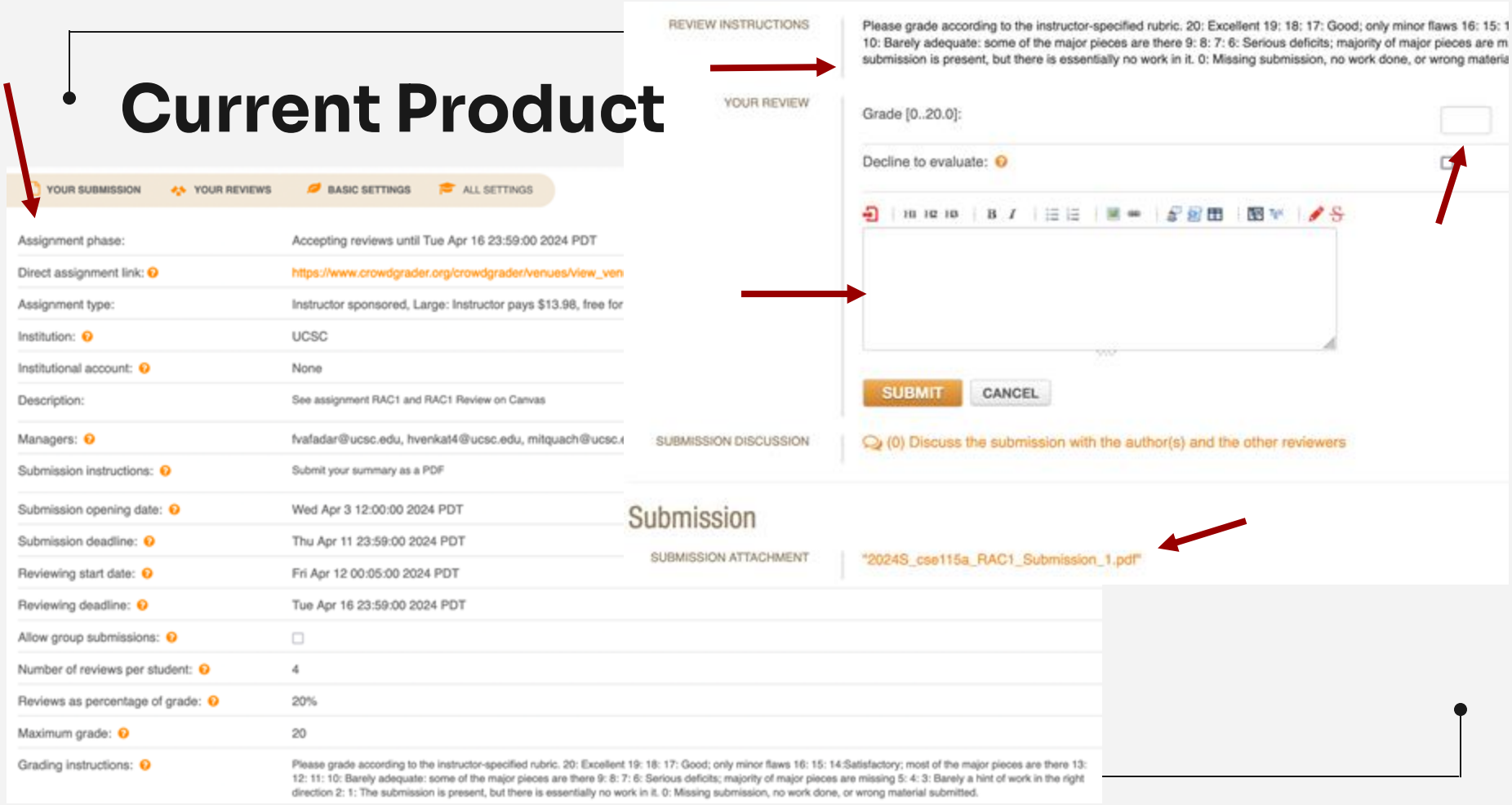
June 5, 2024

Christian Knab, Aastha Verma, Eric Chuang, Eliah Reeves

01

What’s Peergrader?

Current Product



Our Solution

Professors

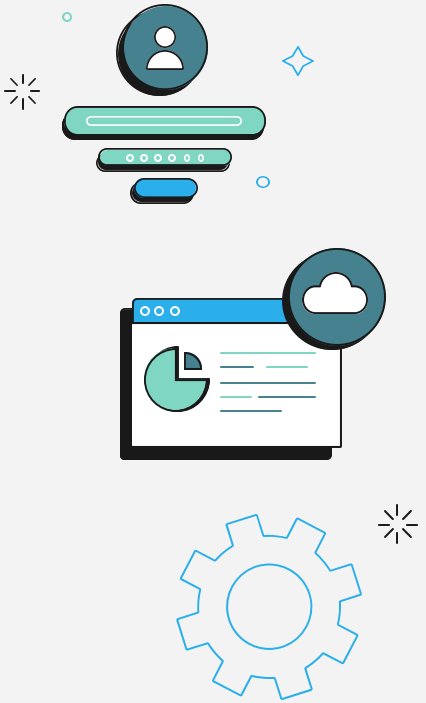
Professors don’t have the bandwidth to grade hundreds of students

Provide a cheap, modern design, with integration to Canvas gradebook for easy grade publishing

Students

Students don’t like the current tool, Crowdgrader

A more intuitive interface increases usability and confidence about submissions & grading



High Level Goals

Original

Students
- join class
- submit assignments
- grade peers
- receive grade

Instructors
- create & customize course
- invite students & restrict enrollment
- create & customize assignments
- sync grades to Canvas

Clean and intuitive design

Achieved

Students
- join class
- submit assignments
- grade peers
- receive grade

Instructors
- create & customize course
- invite students ~~& restrict enrollment~~
- create & customize assignments
- ~~sync grades to Canvas~~

Clean and intuitive design

Biggest Challenges

Technical:

- Learning Next.js
  - Best practices
  - How to write clean code
- Efficiency
  - Caching data
- Security

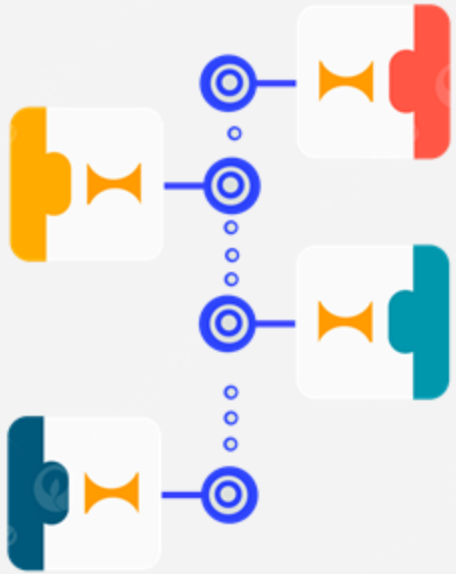
Logistic:

- Task management
  - Used Github issues & Google doc
- Working with conflicting priorities (other classes)

02

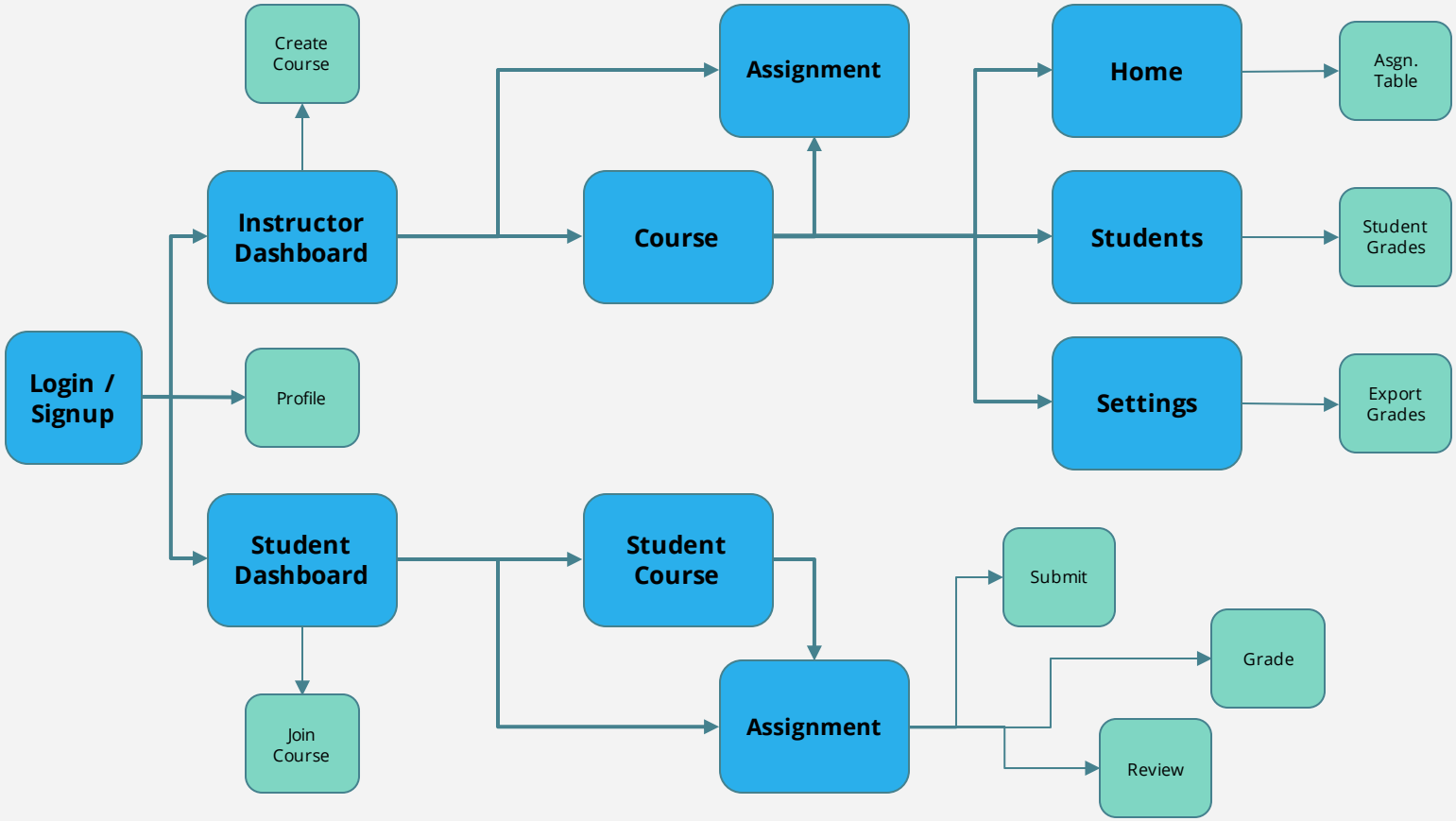
Demo

PeerGrader / Grape Studies
Grape Studies

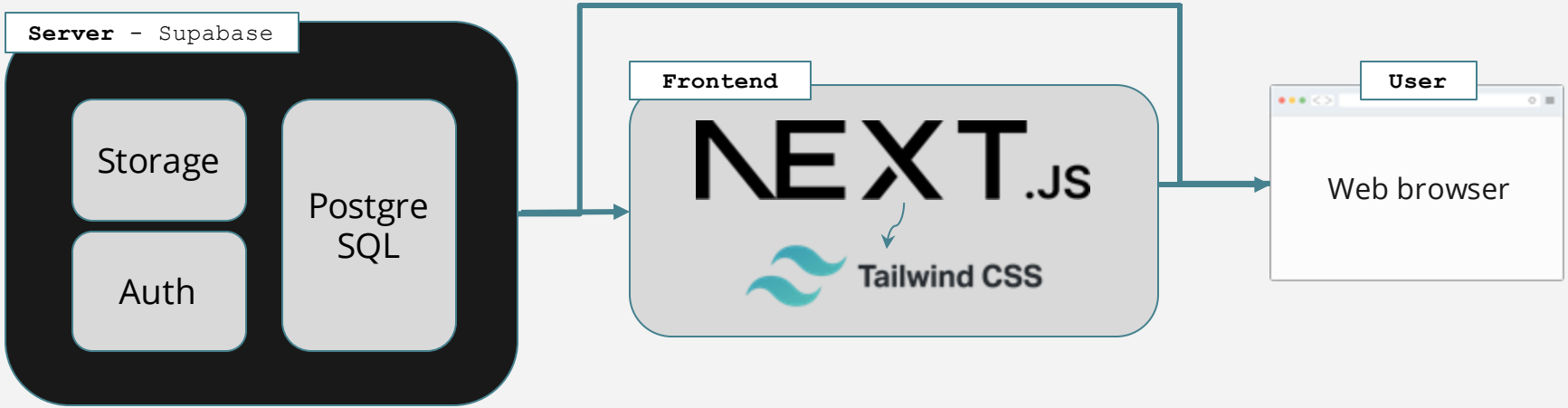


03

Website Outline


Create Course
Assignment
Home
Asgn. Table
Instructor Dashboard
Course
Students
Student Grades
Login / Signup
Profile
Settings
Export Grades
Student Dashboard
Student Course
Submit
Grade
Assignment
Review
Join Course

Technologies



04

Looking Back...

Project Management Techniques

GitHub Issues

Track bugs found

Request changes we wanted someone else to make

ToDo List

List all tasks to accomplish

Keep track of other member's progress

Group Chat

Ask quick questions about bugs, UI changes, ask to look at something

Things we...

Enjoyed

Gaining web development experience

Learning how to use widely used technologies

Experience working in a group

Did Not Enjoy

Merging conflicts

Working around other class priorities

Learning curve of a new technology and rewriting old, bad code

Resolving deployment issues not present in dev builds

Lessons Learned

Keep Doing

Split work across different tasks that do not conflict

Split work between frontend and backend

Consistently updating each other about our next tasks to avoid overlap

Didn't Work

Not prioritizing core functionality

Prioritizing completing user stories over writing quality code

Not fully completing features

Wished we did

Documented code as we wrote it

Wrote more detailed task lists to remember unfinished work

More UI designs so implementation is easier

Created a style guide