

# Senior Design Server/Client Project

## Matching [Phase 2]

CLIENTS: AKHILESH TYAGI &  
JACOB GRUNDMEIER

SANJANA AMATYA, ALEC ELSBERND,  
HAYLEE LAWRENCE, MYTIEN KIEN

SDMAY23-18  
SPRING 2023

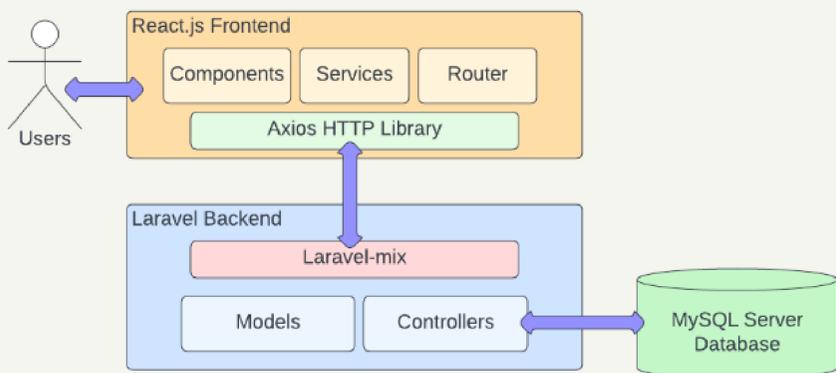
### INTRODUCTION & OVERVIEW

#### Problem Statement

Currently, matching students and projects by hand is time consuming and can lead to client and student dissatisfaction.

#### Our Solution

From the beginning, we aimed to design a website that would facilitate the senior design life cycle. This includes a dashboard for proposal submission, putting preferences, and automated project matching.



### METHODOLOGY

We chose these technologies because we were familiar with them (React.js & MySQL), and we wanted to stay consistent with other technologies used at ISU (Laravel).

For the algorithm, we based it on an implementation of Abraham et al's algorithm. We originally intended to utilize an auction algorithm, however, we were unable to find any existing implementations of such, and creating our own would have been too time-consuming.

### IMPLEMENTATION

#### Frontend

- Coded the front-end using React.js with Material UI Library components
- Unforeseen problems with the connectivity of React library and Laravel back-end, fixed by replacing the library with HTML and CSS

#### Backend

- Laravel connected to the MySQL database using an environment (.env) file
- Huge learning curve for Laravel that took a lot of time to overcome, fixed by lots of research, documentation, tutorials

#### Algorithm

- Based on Abraham's et al's algorithm used for project allocation
- Cycles through students and assigns projects based on top project and groupmate preferences

#### What makes it unique?

Project matching is an example of a classical assignment problem. Knowing this, we decided to create our own algorithm with our own heuristics based off of research and other similar projects.

#### Objectives

- Create a fully functional website that captures senior design lifecycle
- Create an efficient yet reliable matching algorithm that matches students to their preferred projects
- Increase student satisfaction and decrease instructor effort

### IMPACT

- Consolidating all senior design processes into one place
- Instructors will have more time and resources to review proposals
- Students and clients will be more satisfied with their project and team matching results

```
for si in all students
    Project pj = si's highest bid project preference
    Student lk = si's highest bid groupmate preference

    if si has no project preference and has a groupmate preference
        pj = a valid* project for si and lk
    if si already has a project pi
        if lk has a project pk
            sum = si's bid for pk
            sum += si's pid for pi
            if sum > si's bid for pj
                pj = pi
        assign si to pj

    while pj is not a valid* project for lk
        else choose lk = si's next highest groupmate bid

    while lk has a project pk
        total = lk's bid for si
        total += lk's bid for pj
        if total < si's bid for lk
            assign lk to pj
        else choose lk = si's next highest groupmate bid

    while pj has too many students
        sr = student with the lowest bid for pj
        remove sr from pj
for si in all students with no project
    if si has a highest bid project pb and it is valid*
        pj = pb
    else if si has a highest bid groupmate lk who has a valid* project pk
        pj = pk
    else
        pj = the first open project that is valid*
```

### RESULTS

In the end, we were able to create a fully functional front-end application with an effective matching algorithm. We created a solid framework for our project with room for future work to be done. Future iterations can focus on finishing the backend and adding more functionality overall.

### CONCLUSION

We aimed to create an efficient system to benefit our main users: students, clients, and instructors. Our goal was to make their senior design experience easier and less stressful while maximizing their ability to create an impactful product. Ultimately, we developed a solid basis for future projects to iterate upon a fully fleshed-out UI, an algorithm, and a running server.