Oregon State University

CS463 Senior Software Engineering Project: Spring Release

Prototype a web-based tool for creating and executing task-delineated, collaborative, AI-assisted assignments

Group 28:
Oliver Zhou, Ethan Lu, Trent Matsumura, Collin Kimball, Sai Anand

COLLEGE OF ENGINEERING

School of Electrical Engineering and Computer Science

Oregon State University
College of Engineering

Introduction

Our project release presentation will cover these 5 topics:

- Software Completeness
- Code Quality
- Technical Documentation
- End-User Documentation
- Future Work

Oregon State University
College of Engineering

Software Completeness

Completed Requirements/Features

- Organized Assignment and Task Creation
- Configure AI Tools
- Exporting Assignment Objects
- Metacognitive Bot
- Tracking and Analyzing Student Activity
- Deployment with student testing

Oregon State University
College of Engineering

Feature Demo

https://assisted-assignments.vercel.app/

Oregon State University
College of Engineering

Deployment

- When code is merged into the main branch, the frontend and backend will redeploy if any changes are made to them.
- Frontend, Database, and cache are hosted on Vercel

  assisted-assignments – Deployments – Vercel
- Backend is hosted on Render

  assisted-assignments · Web Service · Render Dashboard
- UptimeRobot to check availability of the web server

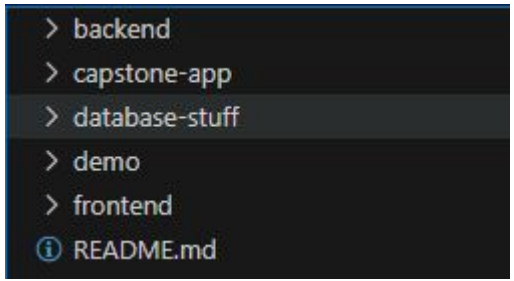
Oregon State University
College of Engineering

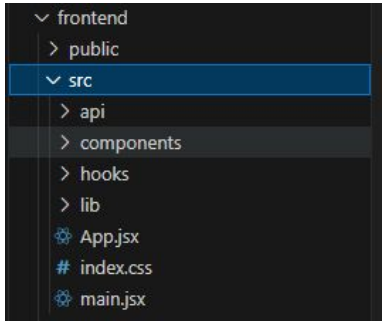
Code Quality

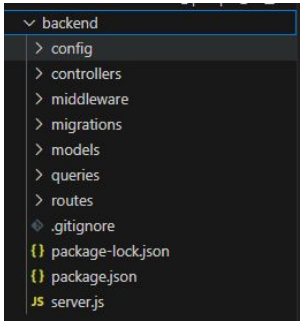
Codebase Organization:

Our code is organized in the format of a frontend and a backend folder that contain various contents of the application. Other folders include the demo, database-stuff, and capstone-app, which were all used in previous renditions of our application but not in our final release.

Furthermore within the frontend and backend folders, we have several folders that organize other features such as components, queries, or models.







Oregon State University
College of Engineering

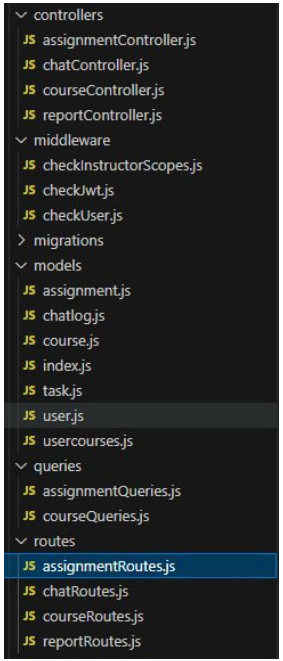
Code Quality Cont.

Key Architecture Decisions and Patterns:

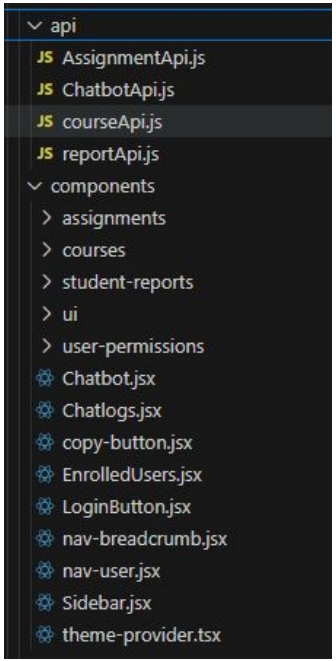
Folders in the frontend and backend are sorted by the feature they fulfill. Grouping the features like this creates a neater work environment that allows debugging and future features to be implemented easier.

Separation of Concerns:

The architecture allows for the separation of concerns to be shown. Many folders of various features show what is being worked on at any given time. This also means that unrelated files will not interfere with each other.



Backend



Frontend

Oregon State University
College of Engineering

Code Quality Cont.

Naming Conventions and Coding Standards:

- Our naming conventions and coding standards were reviewed via team meetings to ensure code quality was upheld. By checking important files we could make sure that the code was legible and formatted correctly as well as being filled with meaningful comments.

Maintenance and extension:

- By neatly storing our files within relevant folders, we can ensure that adding additional features is simple and won't affect other unrelated parts. This allows us to scale up our project as well as debug easier.

Oregon State University
College of Engineering

Technical Documentation

API Documentation

https://app.swaggerhub.com/apis-docs/assistedassignments/TD3A/1.0.0#/

Architectural/Design Documents

https://docs.google.com/document/d/1WjDXrTbgHQwpMtJEwI15nLf8iDnxUtgA8pC3BfwM6as/edit?tab=t.0

Oregon State University
College of Engineering

Technical Documentation (cont.)

Developer Guides

Download/Deployment Tutorials

https://github.com/Kiichigo-cc/assisted-assignments/blob/main/README.md

https://github.com/Kiichigo-cc/assisted-assignments/blob/main/Deployment.md

React Tutorial

https://www.w3schools.com/REACT/DEFAULT.ASP

Gemini Documentation

https://ai.google.dev/gemini-api/docs

Oregon State University
College of Engineering

Technical Documentation (cont.)

Code Comments

The majority of developed code has documentation in the form of code comments to help give clarity as to the purpose of our classes and functions.

Most code comments can be found next to code sections where specific clarity may be needed. Important developer information can be found throughout the code base.

Some functions which require additional context or explanation may have comments to explain the purpose to other developers.

```
1   import express from "express";
2   import {
3     chat,
4     getInstructorChatLogs,
5     getUserChatLogs,
6   } from "../controllers/chatController.js";
7   import { instructorScopes } from "../middleware/checkInstructorScopes.js";
8   import { checkJwt } from "../middleware/checkJwt.js";
9   import checkUser from "../middleware/checkUser.js";
10
11  const router = express.Router();
12
13  router.post("/", checkJwt, chat); // POST route for creating a new chat
14  router.get(
15    "/instructor/logs",
16    checkJwt,
17    instructorScopes,
18    checkUser,
19    getInstructorChatLogs
20  ); // GET route for fetching chat logs
21  router.get("/user/logs", checkJwt, checkUser, getUserChatLogs); // GET route for fetching chat logs
22
23  export default router;
```

Oregon State University
College of Engineering

Technical Documentation (cont.)

Code Reviews

Code reviews were conducted bi-weekly during our team meetings throughout the development process for quality assurance. This involved team discussion over all implemented changes to ensure full understanding of all code.

Where the team felt it was necessary, comments and documentation were added. This was done for the sake of clarity to ensure future developers could easily understand and add on to our code structure without direct training.

The majority of code was modified to have more modular structure to allow future developers to easily add new features and modifications to the software, which can be seen in the codebase organization.

Oregon State University
College of Engineering

End-User Documentation

User Guides are provided on the project GitHub repository. (TD3A GitHub)

Any issues regarding the deployment of the project are covered in the repository guides, as well as usage of the actual program.

The user guide in the README.md contains information on how to run the program locally for development purposes.

No common issues were encountered during the usage process, so issues regarding usage of the project will be in the repo user guides (navigation, assignment interaction, etc.)

Oregon State University
College of Engineering

End-User Documentation

Support resources come from the README.md documentation or the documentation in the team drive.

If further assistance is needed, users or developers can contact any of the team members for additional support.

Team Contact Info	Name	Email
	Collin Kimball	kimbacol@oregonstate.edu
	Oliver Zhou	oliveryzhou@gmail.com
	Ethan Lu	luet@oregonstate.edu
	Sai Meenakshisundaram	meenkass@oregonstate.edu
	Trent Matsumura	matsumut@oregonstate.edu

Oregon State University
College of Engineering

Future Work

Known Bugs

- As of now, each feature implemented into the project functions as intended, and no bugs have been discovered during user testing in real scenarios.
- Future implementations may require bug fixes but not at this time.

Limitations

- The project has limited support for additional context and comprehensive support for secondary features.
- AI assistance is limited in providing the best experience without more rigorous implementation of the AI model.

Oregon State University
College of Engineering

Future Work (cont.)

Architectural Improvements or Refactoring

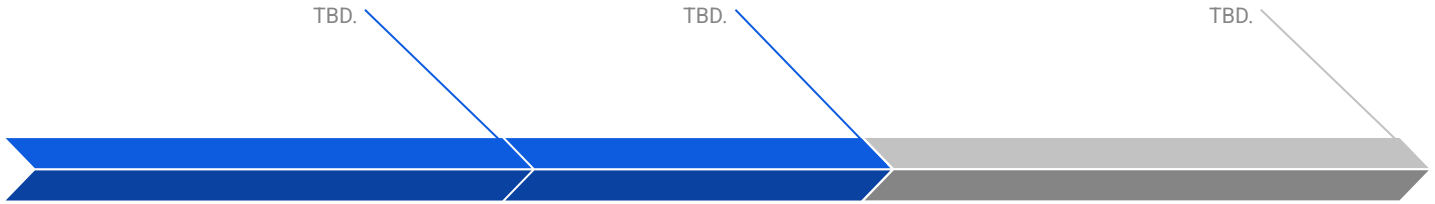
- Our project can improve its structure to remain organized and modular.
- Use additional web technologies which make development of future features easier, and help make things run smoothly.

Operations, Scaling, Maintenance

- Implement pipelines which make deployment of the project more efficient.
- Create school organizations which allow the app to be used in wider school systems.
- Have a developer or administrative accessible backend to manage data in this project.

Oregon State University
College of Engineering

Possible Future Roadmap



Content and File Types

Supporting addition file types such as pdf files to be interpreted as homework submissions.

Integration

Future improvements could be made for this tool to interact with other learning management systems or tools.

More Secondary Features

Other secondary features could include media support, groups, discussion boards, and more.

Oregon State University
College of Engineering

Conclusion

Our project demonstrates completed and deliverable features that were developed under organized standards which helped build our project to be comprehensive.

Additionally, our documentation has made usage of the project understandable for users, and the development process easy to understand for both our team and the future teams who may work on the project after us.

As we approach the project handoff, we will continue to clean up remaining bits of documentation as we see fit, for the handoff to the project mentor or future team.

Oregon State University

Thank You!