Sample DevOps for StudyBuddy

Plan:

- Sit down with StudyBuddy stakeholders and clarify what their goals for the app are
- Map out what will be passed onto the the developers; what it is they'll be coding

Code:

- Once the plan is set in motion the developers take the requirements (plan) and turn it into a viable product for the stakeholders
- All user stories will be taken into account when developing/coding the StudyBuddy app
- User stories that have to do with individual features such as searching for study groups, session scheduling, not-sharing, etc.

Build:

- The code written by the developers is then turned into a working/complete version of the product

Test:

- To ensure quality the product is tested for any bugs or anything else that would need to be fixed
- Test if users can join study groups, schedule sessions, share their notes without any errors, etc.

Integrate:

- All parts of the product (frontend, backend, API, database) are combined to be tested as a whole
- New features can be integrated into the existing codebase so that they work together, not individually

Deploy:

- After the product passes tests and runs smoothly, it gets deployed to where the target audience will be able to start using it
- Stakeholders can use StudyBuddy to see if it's the vision they wanted the project team to execute

Operate:

- Once the app is live the team manages the product by fixing bugs, making updates, and handling user requests
- Someone reports that their notes from a session wasn't saved, the team should be able to help them and fix their issue

Monitor:

- The team constantly checks the product to make sure everything is running smoothly in the sense of overall performance and errors
- Are there any errors when joining study groups? Is the performance speed fast and consistent?

Table

DevOps Steps	What Happens	What Gets Tested	What Gets Built	What Gets Deployed
Plan	Sit down with StudyBuddy stakeholders and clarify what their goals for the app are. Map out what will be passed onto the developers, what it is they'll be coding.	Nothing	Nothing	Nothing
Code	Developers take the plan and turn it into a real product. They code features based on user stories, like searching for study groups, session scheduling, and note-sharing.	Nothing	Core features such as group joining, sessions and everything alike, and note sharing	Nothing
Build	The code written by developers is turned into a working/complete version of the StudyBuddy app.	Basic syntax and errors to make sure that the code functions enough to be testes	A version of the app that can be tested	Nothing
Test	The product is tested for bugs or issues that would need to be fixed. Test if users can join groups, schedule sessions, and upload notes without any problems.	- Group joining - Session scheduling - Note sharing - Notifications	Nothing-it's just being tested	Nothing
Integrate	All parts of the product (frontend, backend, API, database) are brought together and tested as a full system. New features get added into the full codebase	App is tested as a whole to make sure all features work together smoothly.	Nothing much-now all features are combined	Code up until now gets deployed so that it can be tested by those wanting to add features

	so everything works together.			
Deploy	Once the app runs smoothly and passes all tests, it gets deployed to where students and advisors can start using it. Stakeholders get to see the vision brought to life.	This is where the final check takes place to make sure the app is ready to launch.	Final version that’s ready to be launched	Deployed to production
Operate	Now that the app is live, the team manages it. That means fixing bugs, helping users, and making updates when needed. If someone’s notes weren't saved the team looks into it.	Bugs or errors that users may run into.	Nothing-problems from the app are managed	Nothing - at this point it’s already in use
Monitor	The team keeps an eye on the app to make sure everything is running smoothly, checking for crashes or slow performance. Are users able to join study groups without issues? Is everything working fast?	Things such as performance speed and errors.	Nothing-current live version is watched and maintained	Already deployed