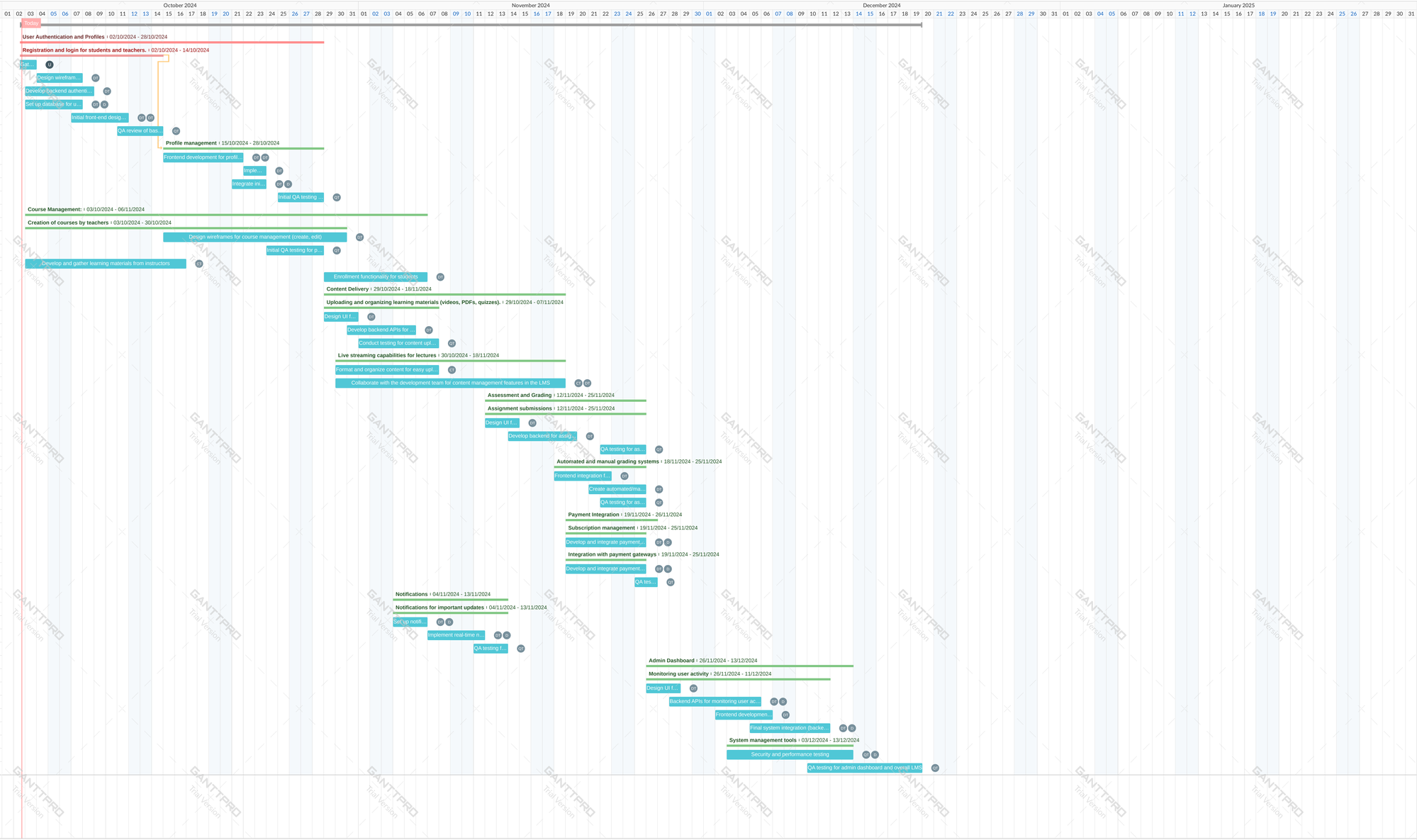
My Team | web based LMS

| Task name | Start date | Duration | Status | Assigned | End date |
|---|---|---|---|---|---|
| | 02/10/202 | 452h | | | 19/12/202 |
| 1 User Authentication and Profiles | 02/10/202 | 148h | | | 28/10/202 |
| 1.1 Registration and login for students and teachers. | 02/10/202 | 68h | | | 14/10/202 |
| 1.1.1 Gather project requirements | 02/10/202 | 12h | Open | Umaimah Abdullah Khan | 03/10/202 |
| 1.1.2 Design wireframes for login, registration, and ... | 04/10/202 | 16h | Open | Design Team | 07/10/202 |
| 1.1.3 Develop backend authentication (OAuth/JWT) | 03/10/202 | 32h | Open | Developers Team | 08/10/202 |
| 1.1.4 Set up database for user roles and profiles | 03/10/202 | 24h | Open | | 07/10/202 |
| 1.1.5 Initial front-end design for login and registrati... | 07/10/202 | 40h | Open | | 11/10/202 |
| 1.1.6 QA review of basic authentication features | 11/10/202 | 16h | Open | QA Team | 14/10/202 |
| 1.2 Profile management | 15/10/202 | 80h | | | 28/10/202 |
| 1.2.1 Frontend development for profile management | 15/10/202 | 40h | Open | | 21/10/202 |
| 1.2.2 Implement course enrollment and profile upd... | 22/10/202 | 16h | Open | Developers Team | 23/10/202 |
| 1.2.3 Integrate initial profile management function... | 21/10/202 | 24h | Open | | 23/10/202 |
| 1.2.4 Initial QA testing for profile and course mana... | 25/10/202 | 16h | Open | QA Team | 28/10/202 |
| 2 Course Management: | 03/10/202 | 200h | | | 06/11/202 |
| 2.1 Creation of courses by teachers | 03/10/202 | 160h | | | 30/10/202 |
| 2.1.1 Design wireframes for course management (cr... | 15/10/202 | 96h | Open | Design Team | 30/10/202 |
| 2.1.2 Initial QA testing for profile and course mana... | 24/10/202 | 24h | Open | QA Team | 28/10/202 |
| 2.1.3 Develop and gather learning materials from in... | 03/10/202 | 80h | Open | Content Team | 16/10/202 |
| 2.2 Enrollment functionality for students | 29/10/202 | 56h | Open | Developers Team | 06/11/202 |
| 3 Content Delivery | 29/10/202 | 120h | | | 18/11/202 |
| 3.1 Uploading and organizing learning materials (vid... | 29/10/202 | 64h | | | 07/11/202 |
| 3.1.1 Design UI for content delivery (upload materi... | 29/10/202 | 24h | Open | Design Team | 31/10/202 |
| 3.1.2 Develop backend APIs for file uploads (video... | 31/10/202 | 32h | Open | Developers Team | 05/11/202 |
| 3.1.3 Conduct testing for content uploads and noti... | 01/11/202 | 40h | Open | QA Team | 07/11/202 |
| 3.2 Live streaming capabilities for lectures | 30/10/202 | 112h | | | 18/11/202 |
| 3.2.1 Format and organize content for easy upload | 30/10/202 | 56h | Open | Content Team | 07/11/202 |
| 3.2.2 Collaborate with the development team for c... | 30/10/202 | 112h | Open | | 18/11/202 |
| 4 Assessment and Grading | 12/11/202 | 80h | | | 25/11/202 |
| 4.1 Assignment submissions | 12/11/202 | 80h | | | 25/11/202 |
| 4.1.1 Design UI for assignment submissions | 12/11/202 | 24h | Open | Design Team | 14/11/202 |
| 4.1.2 Develop backend for assignment submissions | 14/11/202 | 32h | Open | Developers Team | 19/11/202 |
| 4.1.3 QA testing for assignment and payment syste... | 22/11/202 | 16h | Open | QA Team | 25/11/202 |
| 4.2 Automated and manual grading systems | 18/11/202 | 48h | | | 25/11/202 |
| 4.2.1 Frontend integration for grading systems | 18/11/202 | 40h | Open | Developers Team | 22/11/202 |
| 4.2.2 Create automated/manual grading functional... | 21/11/202 | 24h | Open | Developers Team | 25/11/202 |
| 4.2.3 QA testing for assignment and payment syste... | 22/11/202 | 16h | Open | QA Team | 25/11/202 |
| 5 Payment Integration | 19/11/202 | 48h | | | 26/11/202 |
| 5.1 Subscription management | 19/11/202 | 40h | | | 25/11/202 |
| 5.1.1 Develop and integrate payment gateway (sub... | 19/11/202 | 40h | Open | | 25/11/202 |
| 5.2 Integration with payment gateways | 19/11/202 | 40h | | | 25/11/202 |
| 5.2.1 Develop and integrate payment gateway (sub... | 19/11/202 | 40h | Open | | 25/11/202 |
| 5.3 QA testing for assignment and payment systems | 25/11/202 | 16h | Open | QA Team | 26/11/202 |
| 6 Notifications | 04/11/202 | 64h | | | 13/11/202 |
| 6.1 Notifications for important updates | 04/11/202 | 64h | | | 13/11/202 |
| 6.1.1 Set up notification system (email/SMS for up... | 04/11/202 | 24h | Open | | 06/11/202 |
| 6.1.2 Implement real-time notifications | 07/11/202 | 24h | Open | | 11/11/202 |
| 6.1.3 QA testing for content upload and notificatio... | 11/11/202 | 24h | Open | QA Team | 13/11/202 |
| 7 Admin Dashboard | 26/11/202 | 112h | | | 13/12/202 |
| 7.1 Monitoring user activity | 26/11/202 | 96h | | | 11/12/202 |
| 7.1.1 Design UI for admin dashboard | 26/11/202 | 24h | Open | Design Team | 28/11/202 |
| 7.1.2 Backend APIs for monitoring user activity | 28/11/202 | 48h | Open | | 05/12/202 |
| 7.1.3 Frontend development for admin system man... | 02/12/202 | 40h | Open | Developers Team | 06/12/202 |
| 7.1.4 Final system integration (backend, frontend, ... | 05/12/202 | 40h | Open | | 11/12/202 |
| 7.2 System management tools | 03/12/202 | 72h | | | 13/12/202 |
| 7.2.1 Security and performance testing | 03/12/202 | 72h | Open | | 13/12/202 |
| 8 QA testing for admin dashboard and overall LMS | 10/12/202 | 64h | Open | QA Team | 19/12/202 |



**Task 2: Meeting Agenda: LMS Project Kick-Off Meeting**

Meeting Title: LMS Project Kick-Off Meeting

Date: 23/09/2024

Time: 9:00 AM

Duration: 1 hour

Location: Virtual (Zoom/Teams)

**Meeting Objectives:**

**1. Introduction of team members and their roles:**

- Emily, Jeff, Roy - *Frontend Developer*
- Raj, Kristin, Steve - *Backend Developer*
- Lisa - *UI/UI Designer*
- Carlos - *QA Engineer*
- Sophia - *DevOps Engineer*

**2. Scope, Goals and Key deliverables of the project:**

Scope:

We're developing a Web-Based Learning Management System (LMS) for an education client. This platform will handle everything from user management (students, instructors, admins) and course creation to grading, content delivery, and reporting. It will also integrate with third-party tools, ensure data privacy, and be scalable for future growth.

Goals:

- Create an easy-to-use platform for managing courses and users.

- Make learning engaging with multimedia and interactive tools.

- Track student performance with detailed reports.

- Prioritize data security and compliance with privacy regulations.

Key Deliverables:

- A full-featured user and course management system.

- Tools for grading and assessments.

- Multimedia content delivery for courses.

- Detailed performance reports.

- Messaging tools for smooth communication.

- Security features to protect user data.

- Ongoing support and maintenance.

**3. Discuss project timelines, milestones, and Agile sprint structure.**

**4. Identify potential risks and challenges.**

- Changing Project Scope

- Risk: The client might request new features or changes, which could lead to delays and higher costs.

- How to Handle: We'll clearly define the project requirements upfront and manage any changes through a structured process to stay on track.

- Tight Deadlines

- Risk: Rushing to meet deadlines could compromise the quality of the final product.

- How to Handle: We'll set a realistic timeline with some buffer for unexpected delays and make sure the team sticks to it.

- Integration Challenges

- Risk: Integrating third-party tools like video conferencing or databases could lead to compatibility issues.

- How to Handle: We'll test everything thoroughly during integration and keep the setup flexible in case adjustments are needed.

- User Resistance

- Risk: Instructors or students might be hesitant to switch to a new system, slowing down adoption.

- How to Handle: We'll offer training, user guides, and ongoing support to make the transition as smooth as possible.

- Security Risks

- Risk: There's always a chance of security threats, which could compromise user data.

- How to Handle: We'll implement strong security measures from the start and ensure compliance with privacy regulations.

- System Performance

- Risk: The LMS might struggle under heavy use or be hard to scale as more users join.

- How to Handle: We'll optimize the system for performance from day one and plan for future scaling.

- Team Availability

- Risk: If key team members aren't available when needed, it could slow down the project.

- How to Handle: We'll carefully plan who's doing what and have backups ready for critical roles.

By keeping these risks in mind, we can be proactive and make sure things run smoothly!

**5. Establish communication protocols and team collaboration tools.**

**6. Assign initial tasks and clarify responsibilities.**

**Frontend Developers**

- Implement login and registration forms

- Develop profile management UI

- Integrate frontend with backend APIs

- Handle form validation and error handling

**Backend Developers**

- Implement user authentication (OAuth/JWT)

- Develop APIs for login, registration, and profiles

- Handle database schema design and password encryption

- Implement security features

**UI/UX Designers**

- Design wireframes for user authentication and profiles

- Ensure responsive and accessible designs

- Collaborate with frontend developers for a seamless user experience

**QA Engineers**

- Create test plans and execute functional, performance, and security testing

- Perform API testing with Postman

- Conduct security audits (penetration testing) and bug reporting

**DevOps Engineers**

- Set up cloud hosting and manage CI/CD pipelines

- Configure security and scalability (SSL, load balancing)

- Integrate email services for notifications (MFA, password recovery)

**Content Team (optional)**

- Develop and gather learning materials from instructors

- Format and organize content for easy upload

- Collaborate with the development team for content management features in the LMS

---

**Agenda Items:**

1. Welcome and Introductions (5 mins)

- Welcome remarks by Project Manager
- Team introductions and roles

2. Project Overview (10 mins)
- Brief review of project goals and key deliverables
- Overview of the LMS features to be developed

3. Timeline and Sprint Planning (15 mins)
- Presentation of overall project timeline with the help of WBS and Gantt Chart.
- Discuss sprint planning and iteration schedule
- Assign responsibilities for the first sprint (authentication system setup)

4. Potential Risks and Challenges (10 mins)
- Identify any potential technical or operational challenges
- Discuss mitigation strategies

5. Communication and Collaboration Tools (10 mins)
- Overview of communication protocols (Slack, email, etc.)
- Tools for collaboration and code management (Jira, GitHub, etc.)

6. Task Assignments and Responsibilities (5 mins)
- Assign initial tasks to team members for Sprint 1
- Clarify individual responsibilities and collaboration points

7. Next Steps and Q&A (5 mins)
- Open floor for questions and clarifications

End of Meeting

—------

**Task 3: Addressing Adaptive Learning Algorithm Feature Request from the client**

**Ambiguity Identification**

The client's request for implementing "adaptive learning algorithms" raises a few questions that we need to clear up to ensure we're on the same page. Here's a quick rundown of what's a bit unclear and why we should chat about it:

*Unclear Aspects:*

1. Who are we helping?

- Why we need to know: Are we looking to personalize the experience for all students or just certain groups (like beginners or advanced learners)? Understanding this helps us focus our efforts and narrow down our end customer for additional features.

2. What does personalization look like?

- Why we need to know: Should the algorithm tailor things based on how fast students learn, what type of content they like, or something else? This info will shape how we build the system.

3. What kind of data are we talking about?

- Why we need to know: Knowing what data we can use (like quiz scores, time spent on activities, etc.) helps us design the adaptive features.

4. How much automation do you want?

- Why we need to know: Should the algorithm run completely on its own, or do teachers need the ability to step in and make changes?

5. How quickly should it adapt?

- Why we need to know: Is it crucial for the system to adjust in real-time, or is it okay if it adapts at certain checkpoints?

6. What content will we adapt?

- Why do we need to know: Are we changing quizzes, videos, or other types of materials? This affects how we structure the algorithm.

7. Will it integrate with other features?

- Why we need to know: Do we want this adaptive feature to work alongside grading and reporting tools in the LMS?

8. How many students are we expecting?

- Why we need to know: Understanding the scale helps us decide on the technical requirements for the system.

**Questions to Ask the Client:**

1. Who exactly is the adaptive learning system aimed at?
2. What do you want to personalize?
3. What data will we collect?
4. How automated should the process be?
5. How quickly should the system adapt?
6. What types of content should the algorithm modify?
7. Do you expect this feature to work with existing grading and reporting tools?
8. What is the desired scale for the system?
9. How will we know if the personalized experience is successful?
10. Do you have any preferred AI models in mind for this feature?

**Research Plan:**

To better understand adaptive learning algorithms and how to implement them, here's what we'll look into:

Topics to Research:

1. Different Adaptive Learning Algorithms

Examples include adaptive simulated annealing, adaptive coordinate descent, adaptive quadrature, AdaBoost, Adagrad, Adadelta, RMSprop, and Adam.

2. Best Practices in the Industry

- **whatfix** analyzes user interactions in real time and delivers relevant, just-in-time information, walkthroughs, and tips to guide users through complex processes or tasks. This adaptability ensures that users receive the right level of support precisely when they need it, making the learning process more efficient and effective.
- **EdApp** offers a library of customizable, ready-made training courses as well as templates that allow L&D team members to build courses in minutes. This solution also includes a Brain Boost feature, a spaced learning tool that helps learners retain new information by delivering automatic, personalized follow-up tests
- **Adaptemy's** pre-built adaptive learning strategies include adaptive assessments, guided learning reinforcement, and detailed analytic tools to help L&D teams cater learning experiences directly to the needs of employee learners.
- **Knewton Alta's** adaptive learning technology adjusts learning experiences based on learner proficiency levels with every interaction, providing employees with a personalized path to continent mastery.

3. Data Collection and Privacy

It's important to understand how we can collect and use student data while respecting privacy standards.

4. Impact of Personalization

We'll look into how personalization can affect learning outcomes and student engagement.

5. Scalability Considerations

Understanding how adaptive algorithms can handle growth is key.

Resources to Consult:

1. Academic Research

https://www.mdpi.com/2071-1050/16/3/1347

https://link.springer.com/article/10.1007/s40692-022-00250-y#Sec7

2. Industry Articles

Some prominent examples of real-world adaptive learning systems often discussed on eLearning and EdTech blogs include: Khan Academy, Duolingo, DreamBox Learning, Lexia PowerUp Literacy, Rosetta Stone, Smart Sparrow, and Knewton; all of which utilize algorithms to personalize learning paths based on individual student performance, adjusting content difficulty and providing targeted feedback to address specific learning needs.

3. Requirement Refinement

Based on our research and some logical assumptions, here are some potential ways we could implement the adaptive learning feature:

Possible Implementation Approaches:

1. Reinforcement Learning Model:

- This approach could use reinforcement learning to adapt content based on how well students are doing. If a student struggles, we can gradually introduce easier material before moving on to tougher stuff.

2. Collaborative Filtering:

- Similar to recommendation systems, this could suggest content based on what works well for other students with similar performance profiles.

3. Real-Time Data Use:

- The system could collect performance metrics in real-time and adjust content on the fly to better fit each learner's needs.

4. Periodic Adaptation:

- Instead of adjusting everything in real-time, we could reassess students' performance at the end of each module and adapt accordingly.

5. Teacher-Driven Adjustments:

- Give teachers insights from the adaptive system, allowing them to adjust the learning path manually or leverage automated recommendations.

---

**Tasks 4 : Provide detailed answers to the following scenarios:**

**1. Underperforming Team Member:**

*Scenario: Carlos, the QA Engineer, has missed several critical bugs during testing, which has led to issues in production*.

*Question:* How would you address this situation to improve quality?

*Answer:* immediately investigate the root cause of the missed bugs, openly communicate the situation with the team, implement process improvements to enhance test coverage, prioritize training for Carlos on critical areas, and establish a culture of continuous learning and feedback to prevent similar issues in the future; while also taking corrective actions to mitigate the current production issues and fix the identified bugs promptly.

**2. Tight Deadlines:**

*Scenario: The client has moved up the project deadline by two weeks due to market pressures*.

*Question:* What strategies would you employ to ensure the team meets the new deadline without compromising on quality?

*Answer:* First of all I will design a gantt chart which will keep some spaces for such last minute changes or unseen circumstances. So we will have some time margin to this last minute change but if the time change is huge and doesn't fit our schedule I will try to communicate with the client as well as my team. Explain the situation, and review the tasks. We'd focus on the most important things for launch and push non-essential features to after launch. I'd encourage everyone to work in parallel, so frontend and backend tasks happen at the same time, and ensure roles collaborate to move faster. DevOps can start setting up while development finishes. We'd also leave enough time

for testing to maintain quality. I'd keep the client informed about anything that will be delivered later to manage expectations.

**3. Conflict Resolution**:

*Scenario: Emily (Frontend Developer) and Raj (Backend Developer) disagree on the API design, causing delays.*

*Question:* How would you mediate this conflict to reach a timely and effective resolution?

Answer: First, I will try to understand where, when and how a front end developer and a back end developer disagree so that when I hear both sides I know how I have to bring them both on the same page. Then I'd bring Emily and Raj into a meeting to have a transparent conversation about the API design issue. The goal is not to point fingers but to get to the root of the disagreement.

Usually, when front-end and back-end developers clash over a web feature, it's often due to differing priorities. Front-end developers may prioritize the look and immediate functionality, while back-end developers might focus on the scalability and security of the feature.

To bring them both on the same page, I will have to make them both understand that their teamwork will make the project reach its success and will make them understand their way of work and the challenges they have to face in their tasks so that they understand each other's positions and then continue with the project.

The key here is facilitating open communication and making sure the resolution is quick, so we avoid further delays. Conflict should lead to a better solution, not slow down the project.

4. Team Motivation:

Question: Describe specific techniques you use to keep your team motivated during challenging phases of a project.

Answer: **Steps I'd Take**:

- **Acknowledge Small Wins**: Celebrate each small victory along the way. Whether it's completing a sprint or solving a critical bug, recognition goes a long way in keeping morale up.
- **Clear Communication**: Ensure everyone knows their work matters. When deadlines are tight, people need to feel that their contribution is essential to the bigger picture.
- **Provide Support**: Check in regularly to see if anyone is feeling overwhelmed. If needed, I'd adjust workloads or offer additional resources.
- **Create a Vision**: Remind the team of the end goal and how their efforts will positively impact the project. Connecting daily tasks to the bigger picture keeps people motivated.

- **Flexibility and Breaks**: Encourage taking breaks when needed to prevent burnout. A short breather can often lead to better productivity and creativity.

In tough phases, maintaining transparency, offering support, and celebrating progress helps keep everyone motivated and focused.

**Task 5: Meeting Facilitation and Summarization Skills**

**Meeting Minutes:**

**■ Key Discussion Points:**

The YouTube video "Monthly Release Kickoff (Public Livestream)" discusses updates and new features released for a particular platform or service, offering insights on development progress, upcoming improvements, and community engagement. It features presenters explaining changes, roadmap updates, and answering audience questions about functionality. This session gives users a chance to stay informed and ask questions in real-time about ongoing developments.

**Taylor McCaslin** (group manager of product)

**Josh lambert** (core platforms teams)

**Derek ferguson** (devon monitor)

**Tim rizzy** (cicd section)

**Sam white** (sec team)

**Talked about 4 themes**

1. **(theme 1)Enable ai ml efficiency across dev SEC ops lifecycle,**
2. **(theme 2)drive use case adoption to help customers fully realize value,**
3. **(theme 3)differentiation on dev te ops platforms,** and
4. **(theme 4)strengthen our SAAS deployments and capabilities.**

**■ Decisions Made:**

1. The one thing that is done is work on rolling out organizations which is this boundary which is in place around the customer's data. Hen it has this topology service which has all the

data which will decide which router to go to, bootstrapping for this is all done and building for any initial support to the database.

2. Import integrated group is helping in user contribution mapping which will help us in mapping the user from an imported content to the system. Then you can manage and approve them too from an imported system.
3. the source code group will be rolling out the MVC of our consolidated programme, also adding the ability to add roles for owners, maintainers and developers so that they can assign the task to the required person when they want to.
4. New feature they are working on is the time constrained workspaces where the workers when will be working the workspace will be shut down and saved immediately so the work is not lost either the track of time is also maintaing as well as any additional time spent on a work done is monitored as well.
5. The editor extension group is rolling out this new chat feature which will help the developers to open chat directly in the file. They are also planning to add this duo chat feature in visual studio.
6. For code creation team we are moving our repository from current CI pipeline job to the rails monolith moving the CI job to background job
7. Functionality of our policies will have filters in them with which the customers can actually filter them group wise and apply accordingly.

■ **Action Items:**

1. Gitdata storing on kubernetes needs fixing and **Josh Lambert the core platform team** is working on it. And if it works well after deployment then they will go to the production stage then. All the past challenges have been worked on properly and now will be tested if working properly or not.
2. Then there are these multiple teams working on sales projects to make gitlab scale in the future. Furthermore they are working on the movers which help in the gitlab to move the organization one from the other. Next they are working on the geo replication so there is no physical work in physical replication.

3. Working to support unified backup story to support all the data types and make one c interface to make it easy to backup the entire data of gitlab and restore it. Also working on this milestone of CL native hybrid to support other type of deployment, in cell side of things there is this tenant scale group, working on a routing layer, which will reach even minor cells of gitlab and the work is still going.

4. Shifting to the theme enabling AI and efficiency, the **cloud connector team** is working on enhancing the user experience of the health check and letting the customers know if the configuration is correct or not. Team is also working on making it more scalable for additional solutions.

5. **the personal productivity team** is working on overall experience and notifications. And one of the foundational pieces of this is to convert the to-do list current architecture into a view app. Also the team is working on the design token for themes so that in the product themes can be changed more holistically. Like dark mode or better color ratios

6. **Developer group** is working on the enabling the vs code where the maintainers can install and use their choice of extension whenever they want to.

7. Team is also wokring on the pseudo access of the workspaces where the workers can use the containers within the workspace.

8. For the final step of this feature the **code review team** is handling the enhanced merge reuest where the maintainers will be able to assign the correct reviewer for the code. And any feedback will be helping the system to work more efficiently.

9. **CICD section** supports developers and platform engineers in completing their automation configuration and operation tasks after submitting their code in review. This stage is mainly focused on improving workflow.

10. **Pipeline execution team** will work on AI assistant focusing on providing on more targeted responses

11. **SEC dev ops team** in this section the team is working on an architecture which allows the users to finally add the gained permissions to their tokens. This is an enhancement to the current allow list that exists.

12. On the vulnerability management side we plan on bringing credential inventory but is only available for self manage only and will be available for users later

■ **Risks and Issues:**

1. Gitdata storing on kubernetes needs fixing and Josh Lambert the core platform team is working on it.
2. Second area to be looked at is the mitigating of lock contention within the posters database in gitlab on higher scale deployment. This will be addressed in a few different ways to make sure it is not a problem anymore.
3. The challenge they are facing in it is needed to move the entire database from gitlab.com to different organizations' databases whereas only selective data needs to be moved so it is

solved by doing logic moving of data which will only move selected parts of the database to the requested destination

4. there are some (challenges) like how the design tokens will work in CSS.

**Analysis and Recommendations:**

○ Meeting Effectiveness:

■ Briefly evaluate the effectiveness of the meeting.

One thing that i noticed and always appreciated about project management is it keeps the whole team on track and enhance the team's productivity and keep them motivated when their is continuos accountability.

■ Suggest any improvements or best practices that could enhance future meetings.

A headstart of the project and the workings in the team would have been helpful to understand the task and their success parameters.