Collaboration, Teamwork & Project Management

Why Collaboration Matters

- Combines diverse skills & ideas

- Builds trust and accountability

- Helps solve problems faster

Question: *Share one moment where teamwork helped you succeed.*

Key Elements of Teamwork

- Communication

- Defined roles

- Respect and empathy

- Shared goals

Question: *Which of these is hardest in your experience?*

Challenges in Collaboration

- Miscommunication

- Unequal participation

- Conflicts

- Weak leadership

Scenario: *Two teammates disagree strongly—what would you do?*

Made with GAMMA

Defining Project Scope & Objectives

- Scope = what's covered (and not covered)

- Objectives must be SMART

- Early clarity avoids scope creep

Question: *Give an example of a SMART objective.*

Team Roles & Responsibilities

- Project manager, team, sponsors, stakeholders

- Role clarity = accountability

- Everyone must know tasks & deadlines

Question: *How do you ensure role clarity?*

Risk Management in Projects

- Identify risks early → mitigation plans

- Risks can be threats or opportunities

- Review risks regularly for resilience

Question: *Name one risk you've faced in a project. How was it handled?*

Quality Control & Performance Tracking

- Maintain quality (Six Sigma, TQM, audits)

- Track with Gantt charts, dashboards, KPIs

- Continuous improvement through data

Question: *How do you measure success in your projects?*

Communication & Stakeholder Management

- Clear updates keep everyone aligned

- Regular meetings, transparent feedback

- Stakeholder buy-in improves results

Question: *How do you handle communication gaps in a team?*

Closing & Lessons Learned

- Review achievements & challenges

- Document lessons for future projects

- Celebrate wins and recognize contributions

Question: *What's one lesson you learned from a project?*

Project Management Phases (Overview)

1. Initiation

2. Planning

3. Execution

4. Monitoring & Control

5. Closure

Activity: *Think of a project—what phase did you rush through?*

Project Management Tools

- Trello / Notion → tasks

- Google Docs → collaboration

- Slack/Discord → communication

- GitHub → coding projects

Question: *Which tool would you try first for a group project?*

Leadership in Projects

- Assign roles by strengths

- Motivate & resolve conflicts

- Ensure fairness

- Leaders are accountable

Question: *Do you prefer leader or team player role? Why?*

Real-World Example

- Example: School tech club building a website

- Collaboration: designers + coders + writers

- PM: tasks & deadlines

- Teamwork: resolving conflicts

Question: *Which role would you take here?*

Soft Skills in Project Management

- Adaptability

- Critical thinking

- Time management

- Emotional intelligence

Question: *Which soft skill do you want to improve?*

Conflict Resolution

- Listen to both sides

- Focus on common goals

- Use mediator/leader if needed

Scenario: *If teammates stop talking to each other, what's your approach?*

Future of Project Management (AI & Tech)

- AI tools for scheduling & risk prediction

- Smart dashboards & auto-reporting

- Collaboration across time zones with AI assistants

Question: *How could AI make school/college projects easier?*

Commonly Used Terms in Software Industry

- **Escalation** – Raising an issue to a higher authority when it can't be solved at the current level.

- **Backlog** – List of tasks, features, or bugs waiting to be worked on.

- **Sprint** – A fixed time box (usually 2 weeks) in Agile where specific tasks are completed.

- **Burnout** – When a developer or team is overworked, productivity drops.

- **Technical Debt** – Shortcuts taken in code/design that will create problems later if not fixed.

- **Stand-up** – A short daily meeting (usually 15 mins) to discuss progress, blockers, and next steps.

- **Blocker** – A critical issue preventing progress until resolved.

- **Dependency** – When your task can't move forward until another team/member finishes theirs.

- **Root Cause Analysis (RCA)** – Digging into the real reason why a problem occurred.

- **On-call** – When a developer/engineer is responsible for monitoring and fixing production issues after hours.

- **Hotfix** – An urgent patch or code fix pushed directly to production.

- **Downtime** – When a system/service is unavailable.

- **Release Cycle** – The planned process and schedule for rolling out new features.

- **Scope Creep** – Unplanned expansion of project requirements beyond the agreed scope.

- **Code Freeze** – Period where no new features are added before a release, only bug fixes.

- **CI/CD (Continuous Integration/Continuous Deployment)** – Automated pipelines for testing and deploying software.

- **Rollback** – Reverting back to a previous version due to issues with the new release.

- **Load Testing / Stress Testing** – Testing how the system performs under heavy use.

- **Postmortem** – A report or meeting after a failure to discuss what went wrong and how to prevent it.

- **Stakeholders** – Anyone invested in the project (clients, managers, end users, etc.).