3. Project Plan

3.1 Project Management/Tracking Procedures

To manage and track progress, we are using the following tools:

- Git: We use Git for version control to manage and integrate code across all team members collaboratively. Each feature or bug fix is handled on a separate branch and merged into the main branch after review.
- Google Docs: We use shared Google Docs to write our reports, maintain to-do lists, and record changes. This helps keep everyone up-to-date and provides a central location for documentation.
- Weekly Meetings: We meet once a week with our client to discuss progress, receive feedback, and set goals.

This setup keeps our team organized and allows us to adapt and refine our work as the project progresses collaboratively.

3.2 Task Decomposition

To build Smartclass, we decomposed the overall problem into key functional areas and corresponding subtasks. Each task represents a component of the final product and can be aligned with Agile sprints of iterative development.

1. Authentication and User Management
    - Create user registration and login functionality
    - Implement user roles (Student, Teacher, TA)
    - Secure user data (password hashing)
    - Enable profile management (password resets, etc)
2. Questions and Answer System
    - Build UI for students to post questions (text, image, etc)
    - Enable anonymous question posting
    - Allow replies/comments for each question
    - Implement media upload and playback support
    - Add categorization by context (lecture, homework, quiz, etc)
3. Polling and Quizzes Features
    - UI for instructors to create and launch polls/quizzes
    - Student response interface (real-time updates)
    - Store and track participation data
    - Export participation results for grading
4. Archiving and Retrieval
    - Store all Q&A and polls persistently in a database
    - Implement filtering/searching by category or date
    - Provide instructor and student views of the archive
5. Statistics and Leaderboard (Stretch Features)
    - Track the number of contributions per user
    - Display leaderboard of top 10 student contributors
    - Display TA contribution stats
6. Backend Development
    - Design and implement database schema (SQL)
    - Set up API endpoints
    - Ensure secure handling of media files
    - Implement data validation and error handling

7. Frontend Development
    - Set up a responsive UI framework (React)
    - Build navigation between views (auth, questions, polls, stats)
    - Integrate with backend APIs
    - Support mobile-friendly design
8. Testing
    - Unit tests for frontend and backend components
    - End-to-end system testing
    - Set up CI/CD pipeline

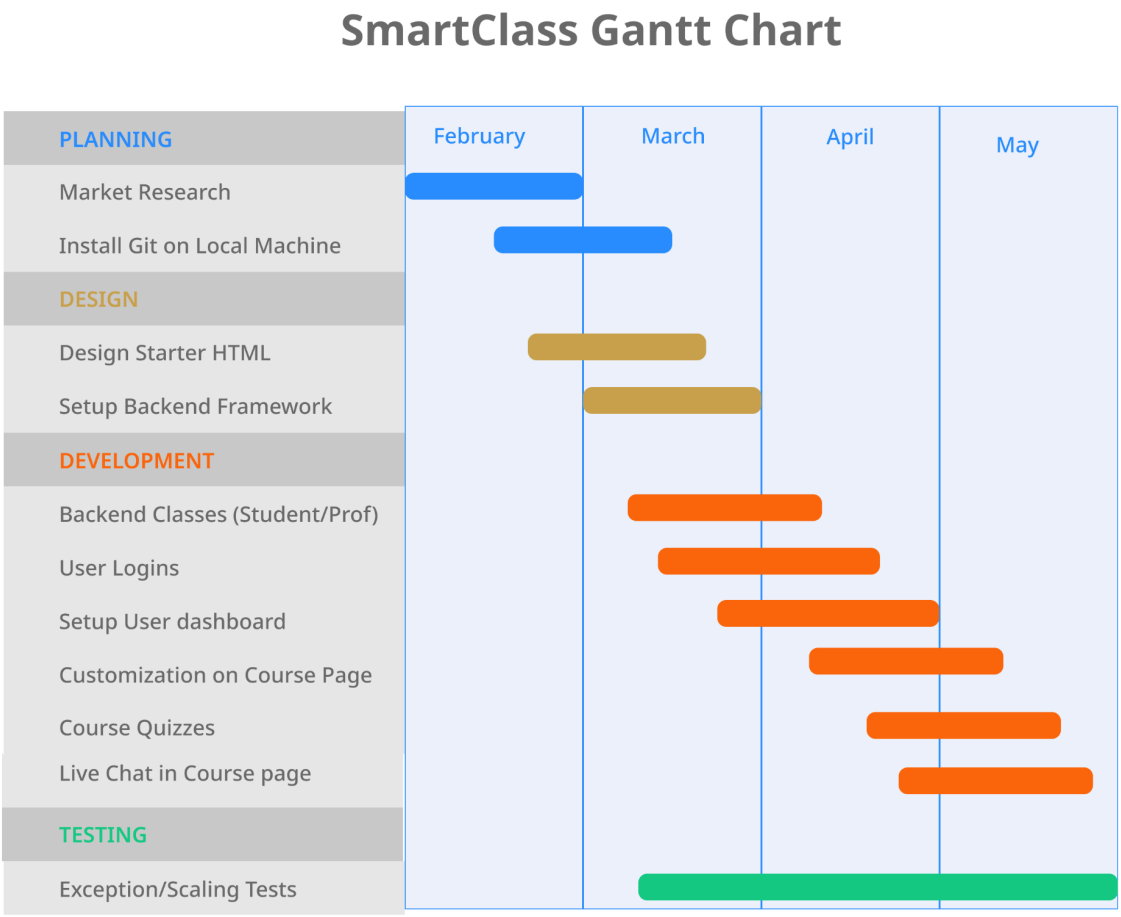
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

We have established key project milestones and corresponding metrics to ensure timely and measurable progress on the SmartClass platform. These milestones are structured around the major components of the application and aligned with our sprint schedule. Each milestone includes specific evaluation criteria to determine success.

1. Authentication & User Management
    - User login and registration completed.
      Metric: Users can sign up, and log in, and roles (Student, TA, Instructor) are correctly stored in the database.
    - Role-based access control implemented
      Metric: Students, TAs, and Instructors see different views and permissions.
2. Question & Answer System
    - Basic Q&A with text implemented
      Metric: Students can post and reply to text questions; data is stored and displayed correctly.
    - Media support enabled
      Metric: Students can upload and playback image, audio, and video content in posts or replies.
    - Anonymous posting and categorization functional
      Metric: Students can choose anonymity; posts are sorted by type (e.g., lecture, lab, homework).
3. Polling and Quizzing
    - Poll creation and launch system completed.
      Metric: Instructors can create and launch polls using text/images.
    - Student responses and results displayed implemented
      Metric: Real-time response tracking; instructors can download a CSV of participation.
4. Archiving and Retrieval
    - Archive system and basic search functional
      Metric: Q&A and polls are stored by date/category; can be retrieved and filtered by users.
5. Statistics and Leaderboard
    - Participation data tracking is complete
      Metric: Number of contributions logged per user.
    - Leaderboard for students and TAs implemented
      Metric: Top contributors displayed by rank and number of posts.
6. Frontend UI and UX
    - Responsive layout and navigation completed
      Metric: Pages render cleanly on different devices; navigation is smooth.
    - Client-approved UI/UX design polish
      Metric: Positive client feedback on usability and layout.
7. Testing and Deployment
    - The system passes unit and integration tests.
      Metric: All core features pass test cases without breaking existing functionality.

3.4 Project Timeline/Schedule

3.4.1 Gantt Chart



3.4.2 Detailed Schedule Breakdown By Date

- **February 15 - 28** – Market Research
- **February 15 - March 15** – Install Git on Local Machine
- **February 20 - March 20** – Design Starter HTML
- **March 1 - March 31** – Setup Backend Framework
- **March 10 - April 10** –Backend Classes (Student/Prof)
- **March 15 - April 15** – User Logins
- **March 20 - April 30** – Setup User Dashboard
- **April 10 - May 5** – Customization on Course Page
- **April 15 - May 15** – Course Quizzes
- **April 20 - May 20** – Live Chat on Course Page
- **March 10 - May 31** – Exception/Scaling Tests

3.5 Risks and Risk Management/Mitigation

For each major task, we identified potential risks that could impact project success. Each risk is assessed based on its probability (Low < 0.3, Medium ≈ 0.5, High > 0.7) and severity (Low, Medium, High). We outline a mitigation plan for any risk with a probability greater than 0.5 or high severity.

1. Authentication & User Management
    - Risk: Security vulnerabilities in the login system (e.g., weak encryption, session hijacking)
      Probability: Medium (0.5)
      Severity: High
      Mitigation: Use industry-standard libraries (e.g., bcrypt for hashing), enforce secure login flows, and conduct security testing.
2. Question & Answer System
    - Risk: Media upload/storage issues (e.g., large file sizes, unsupported formats)
      Probability: High (0.7)
      Severity: Medium
      Mitigation: Limit upload sizes, validate file types, and use third-party cloud storage for scalability.
    - Risk: Poor UI/UX leads to underuse by students
      Probability: Medium (0.5)
      Severity: High
      Mitigation: Regular feedback from clients and students; perform usability tests and iterate on design early.
3. Polling and Quizzing Features
    - Risk: Real-time response delays or data inconsistency
      Probability: Medium (0.4)
      Severity: Medium
      Mitigation: Use efficient frameworks (e.g., socket.io or polling fallback), test under load, and simulate classroom-size users.
4. Archiving and Retrieval
    - Risk: Search or filter performance degrades with large datasets
      Probability: Medium (0.5)
      Severity: Medium
      Mitigation: Index database fields correctly; use pagination and caching to optimize frontend performance.
5. Statistics and Leaderboards
    - Risk: Inaccurate or unfair contribution tracking
      Probability: Medium (0.4)
      Severity: Medium
      Mitigation: Clearly define contribution rules (e.g., minimum content length), test metric logic, and allow instructor override if needed.
6. Testing and Deployment
    - Risk: Incomplete test coverage leads to bugs in production
      Probability: Medium (0.5)
      Severity: High
      Mitigation: Enforce testing as part of the development cycle (e.g., require tests for pull requests), and use automated test tools.
    - Risk: Deployment issues or downtime near the demo
      Probability: High (0.6)
      Severity: High
      Mitigation: Deploy early on a staging server, conduct trial runs, and have a backup environment prepared.
7. Team-Level Risks
    - Risk: Uneven workload or member availability
      Probability: Medium (0.5)
      Severity: Medium
      Mitigation: Regular check-ins, shared documentation, rotating responsibilities, and pairing for critical tasks.
    - Risk: Delayed client feedback stalls progress
      Probability: Low (0.3)
      Severity: Medium
      Mitigation: Propose agenda in advance of meetings, and prioritize independent development items between feedback cycles.

3.6 Personnel Effort Requirements

Task	Subtask	Est. Hours
Authentication & User Management	User login/registration & role setup Profile management and access control Security implementation	10
Q&A System	Text-based Q&A functionality Media upload and playback Anonymous posting and categorization	15
Polling & Quizzes	Poll creation Student response interface and live updates Exporting participation data	10
Archiving & Retrieval	Data persistence and organization Search, filter, and view by category	10
Statistics & Leaderboard	User activity tracking Displaying leaderboards for students/TAs	10
Frontend UI/UX	Basic layout and responsive design Navigation and state management UI Polish and client feedback integration	25
Testing & Debugging	Unit and integration testing System testing and bug fixing	20
Deployment & Documentation	Hosting and deployment Final presentation and written documentation	30

3.7 Other Resource Requirements

In addition to time and effort from team members, the SmartClass project will require the following non-financial resources to support development:

1. Development tools & Platforms
    a. Code Editor/IDE: Visual Studio Code
    b. Version Control: Git
2. Hosting & Backend Services
    a. Database: MySQL
3. Communication & Collaboration Tools
    a. Google Docs: For documentation and planning
    b. Email & meetings: Weekly client meetings for reports and feedback