**Phase 1: Initial Setup (0-2 Months)**

**Database**

* **Week 1-2:**
  + **Set up PostgreSQL database.**
  + **Implement basic tables: users, teams, team\_members.**
  + **Test relationships between users and teams.**

**Backend**

* **Week 3-4:**
  + **Set up Node.js and Express backend.**
  + **Implement authentication and authorization for users (JWT-based).**
  + **Implement API routes for users, teams, and team\_members.**
  + **Add basic CRUD operations for teams and user management.**

**Frontend**

* **Week 5-6:**
  + **Set up React app with Tailwind CSS for UI.**
  + **Implement user registration, login, and authentication screens.**
  + **Build team creation and user assignment UI.**

**Phase 2: Core Collaboration Features (2-4 Months)**

**Database**

* **Week 7-8:**
  + **Create tasks, task\_dependencies tables for task management.**
  + **Implement relationships between tasks and users (assigning tasks).**
  + **Add basic functionality for task creation and assignment.**
* **Week 9-10:**
  + **Create meetings table and relationship to teams.**
  + **Add meeting\_transcriptions table for storing AI-generated transcriptions.**
  + **Implement basic CRUD operations for meetings (create, update, delete).**

**Backend**

* **Week 7-8:**
  + **Implement task management API (CRUD operations for tasks, task assignments).**
  + **Implement task dependency handling.**
* **Week 9-10:**
  + **Create API routes for scheduling and managing meetings.**
  + **Implement AI-powered meeting transcription (connect to external AI service like OpenAI for transcription).**
  + **Implement basic AI features for action point extraction from meetings.**

**Frontend**

* **Week 7-8:**
  + **Build UI for task creation, viewing, and assigning.**
  + **Integrate task management UI with backend.**
* **Week 9-10:**
  + **Build meeting scheduling UI with agenda input.**
  + **Display AI-generated meeting transcriptions and action points in a user-friendly format.**

**Phase 3: Collaboration Tools and AI Features (4-6 Months)**

**Database**

* **Week 11-12:**
  + **Create chat\_messages table for team chat functionality.**
  + **Add files table for file sharing and versioning (store file metadata).**
* **Week 13-14:**
  + **Implement tables for AI-driven insights: task\_insights and sentiment\_feedback.**
  + **Link chat messages and files to tasks and meetings as needed.**

**Backend**

* **Week 11-12:**
  + **Implement real-time messaging functionality using WebSockets (for chat messages).**
  + **Integrate file upload functionality (store file metadata in files table, upload to server).**
* **Week 13-14:**
  + **Implement sentiment analysis and AI-driven task insights APIs.**
  + **Set up API routes for retrieving and displaying chat messages and files.**

**Frontend**

* **Week 11-12:**
  + **Build team chat interface with real-time updates (WebSocket integration).**
  + **Allow users to upload and download files related to tasks and meetings.**
* **Week 13-14:**
  + **Display AI-driven task insights and sentiment feedback in user dashboards.**
  + **Build a sentiment feedback form for users to submit team morale insights.**

**Phase 4: Advanced Features and Optimizations (6-9 Months)**

**Database**

* **Week 15-16:**
  + **Optimize database schema for scalability (consider indexing frequently queried columns).**
  + **Implement advanced querying for performance insights (e.g., workload distribution, bottleneck identification).**

**Backend**

* **Week 15-16:**
  + **Optimize backend for performance and scalability (consider Dockerizing the backend, autoscaling using AWS/Azure).**
  + **Implement advanced AI-driven workflows and insights for improving team productivity (e.g., task prioritization, task progress tracking).**

**Frontend**

* **Week 15-16:**
  + **Enhance UI/UX with performance dashboards showing team progress, workload distribution, and sentiment analysis.**
  + **Implement responsive design and mobile optimization for collaboration features.**

**Phase 5: Full Integration & Testing (9-12 Months)**

**Database**

* **Week 17-18:**
  + **Test and validate all database relationships, data integrity, and ensure consistency with business logic.**
  + **Set up backup and recovery strategies for the database.**

**Backend**

* **Week 17-18:**
  + **Complete integration of all backend APIs and AI features.**
  + **Final testing of backend functionality, including API endpoints for tasks, meetings, chat, file sharing, and AI-driven features.**

**Frontend**

* **Week 17-18:**
  + **Final integration of frontend with all backend services (e.g., task management, meetings, chat, file sharing).**
  + **Perform cross-browser and device testing to ensure frontend is fully responsive and user-friendly.**

**Phase 6: Deployment & Monitoring (12+ Months)**

**Backend & Database**

* **Week 19-20:**
  + **Deploy the app to a cloud platform (e.g., AWS, Azure).**
  + **Set up continuous integration and deployment (CI/CD) pipelines for seamless updates.**

**Frontend**

* **Week 19-20:**
  + **Deploy the frontend application to a CDN or hosting service (e.g., Vercel, Netlify).**
  + **Set up performance monitoring for frontend and backend (e.g., error tracking with Sentry, performance monitoring with New Relic).**

**Summary of Phases and Timeline**

1. **Phase 1 (0-2 Months): Database setup, user authentication, basic team management.**
2. **Phase 2 (2-4 Months): Core collaboration features: task management, meeting scheduling, and transcription.**
3. **Phase 3 (4-6 Months): Collaboration tools and AI features: real-time chat, file sharing, sentiment analysis.**
4. **Phase 4 (6-9 Months): Advanced features and optimizations: AI-driven task insights, performance dashboards.**
5. **Phase 5 (9-12 Months): Full integration and testing: final user acceptance testing, all features integrated.**
6. **Phase 6 (12+ Months): Deployment and monitoring: app deployment, CI/CD setup, performance monitoring.**