**Phase 1: Project Initialization and UI Design**

**1. Set Up Development Environment**

* **Task:** Establish coding standards and guidelines
  + **Labels:** Feature: Maintainability, Backend, Frontend, Documentation
* **Task:** Initialize Git repository and project structure
  + **Labels:** Feature: Maintainability, DevOps

**2. User Interface Design - Initial Setup**

* **Task:** Create UI mockups and wireframes
  + **Labels:** Feature: Usability, Frontend, Design, High Priority
* **Task:** Create minimum accessible webpage
  + **Labels:** Feature: Usability, Frontend, Design, High Priority

**3. Set Up Firebase**

* **Task:** Create Firebase project and configure authentication and Firestore
  + **Labels:** Feature: Integration, Backend, Firebase, High Priority
* **Task:** Securely manage environment variables and secrets
  + **Labels:** Feature: Deployment, Backend, Security

**Phase 2: Core Features Development**

**1. User Account Management - Basic Authentication**

* **Task:** Implement Email and Password Registration (Frontend & Backend)
  + **Labels:** Feature: User Account Management, Backend, Frontend, High Priority
* **Task:** Validate Password Complexity Requirements
  + **Labels:** Feature: User Account Management, Backend, Security
* **Task:** Send Verification Email Upon Registration
  + **Labels:** Feature: User Account Management, Backend, Email Service
* **Task:** Implement Email and Password Login (Frontend & Backend)
  + **Labels:** Feature: User Account Management, Backend, Frontend, High Priority

**2. Error Handling**

* **Task:** Implement global error handling in backend
  + **Labels:** Feature: Error Handling, Backend, High Priority
* **Task:** Implement error boundaries in React frontend
  + **Labels:** Feature: Error Handling, Frontend
* **Task:** Display user-friendly error messages
  + **Labels:** Feature: Error Handling, Frontend, UI/UX

**3. Implement Security Best Practices**

* **Task:** Ensure SSL/TLS is properly configured
  + **Labels:** Feature: Security, Backend, DevOps, High Priority
* **Task:** Use Helmet.js for Express security
  + **Labels:** Feature: Security, Backend

**Phase 3: Task Management**

**1. Task Management - Basic Functionality**

* **Task:** Implement task creation functionality
  + **Labels:** Feature: Task Management, Frontend, Backend, High Priority
* **Task:** Implement task editing functionality
  + **Labels:** Feature: Task Management, Frontend, Backend
* **Task:** Implement task deletion functionality
  + **Labels:** Feature: Task Management, Frontend, Backend
* **Task:** Validate task input fields
  + **Labels:** Feature: Task Management, Frontend, Backend, Validation

**2. User Account Management - Advanced Features**

* **Task:** Implement password reset via email
  + **Labels:** Feature: User Account Management, Backend, Frontend, Email Service
* **Task:** Design "Forgot Password" UI
  + **Labels:** Feature: User Account Management, Frontend, UI/UX

**3. Session Management**

* **Task:** Implement secure session handling
  + **Labels:** Feature: User Account Management, Backend, Security
* **Task:** Implement auto logout after 30 minutes of inactivity
  + **Labels:** Feature: User Account Management, Backend, Frontend, Security

**Phase 4: Enhancements and Additional Features**

**1. Task Management - Advanced Features**

* **Task:** Implement task lists/categories
  + **Labels:** Feature: Task Management, Frontend, Backend, Enhancement
* **Task:** Implement drag-and-drop task reordering
  + **Labels:** Feature: Task Management, Frontend, UI/UX
* **Task:** Implement task search functionality
  + **Labels:** Feature: Task Management, Frontend, Backend
* **Task:** Implement task filtering by attributes
  + **Labels:** Feature: Task Management, Frontend, Backend

**2. Profile Management**

* **Task:** Create user profile page
  + **Labels:** Feature: User Account Management, Frontend, UI/UX
* **Task:** Implement profile update functionality
  + **Labels:** Feature: User Account Management, Backend, Frontend
* **Task:** Allow users to upload profile pictures
  + **Labels:** Feature: User Account Management, Backend, Frontend, File Upload

**3. Collaboration Features**

* **Task:** Implement task sharing functionality
  + **Labels:** Feature: Collaboration, Frontend, Backend, High Priority
* **Task:** Set up permission levels (Viewer, Editor, Owner)
  + **Labels:** Feature: Collaboration, Backend, Security
* **Task:** Implement group creation and management
  + **Labels:** Feature: Collaboration, Frontend, Backend
* **Task:** Implement task commenting system
  + **Labels:** Feature: Collaboration, Frontend, Backend, Real-Time

**Phase 5: Reminders and Notifications**

**1. Implement Notifications**

* **Task:** Implement notification preferences settings
  + **Labels:** Feature: Notifications, Frontend, Backend, Settings
* **Task:** Allow users to choose notification channels
  + **Labels:** Feature: Notifications, Frontend, Backend, Email Service, SMS, Push Notifications
* **Task:** Integrate email notification service (e.g., SendGrid)
  + **Labels:** Feature: Notifications, Backend, Email Service

**2. Advanced Notification Features**

* **Task:** Implement push notifications for supported devices
  + **Labels:** Feature: Notifications, Frontend, Backend, Push Notifications
* **Task:** Allow users to set reminder times before task due dates
  + **Labels:** Feature: Notifications, Frontend, Backend
* **Task:** Implement recurring notifications for repeating tasks
  + **Labels:** Feature: Notifications, Frontend, Backend

**Phase 6: Real-Time Synchronization and Performance**

**1. Real-Time Synchronization**

* **Task:** Implement real-time database listeners with Firestore
  + **Labels:** Feature: Synchronization, Frontend, Backend, Real-Time
* **Task:** Ensure data consistency across devices
  + **Labels:** Feature: Synchronization, Backend, Testing

**2. Performance Optimization**

* **Task:** Implement code splitting and lazy loading in React
  + **Labels:** Feature: Performance, Frontend, Optimization
* **Task:** Minify and compress assets
  + **Labels:** Feature: Performance, Frontend, Build Process
* **Task:** Integrate performance monitoring tools (e.g., Google Lighthouse)
  + **Labels:** Feature: Performance, Frontend, Testing

**Phase 7: Accessibility and Compliance**

**1. Accessibility Compliance**

* **Task:** Ensure all images have alt text
  + **Labels:** Feature: Accessibility, Frontend, UI/UX
* **Task:** Provide keyboard navigation support
  + **Labels:** Feature: Accessibility, Frontend, UI/UX
* **Task:** Test compatibility with screen readers
  + **Labels:** Feature: Accessibility, Frontend, Testing

**2. Legal and Regulatory Compliance**

* **Task:** Draft privacy policy and terms of service
  + **Labels:** Feature: Compliance, Documentation, Legal, High Priority
* **Task:** Display privacy policy prominently in the application
  + **Labels:** Feature: Compliance, Frontend, UI/UX
* **Task:** Allow users to request data deletion or export
  + **Labels:** Feature: Compliance, Backend, Frontend, Security

**Phase 8: Security Enhancements**

**1. Security Assessments**

* **Task:** Protect against OWASP Top Ten vulnerabilities
  + **Labels:** Feature: Security, Backend, Testing
* **Task:** Conduct regular vulnerability scanning
  + **Labels:** Feature: Security, Backend, Testing, Recurring Task
* **Task:** Schedule and perform penetration testing
  + **Labels:** Feature: Security, Testing, External

**2. Implement Multi-Factor Authentication (MFA)**

* **Task:** Implement multi-factor authentication
  + **Labels:** Feature: User Account Management, Backend, Security, Enhancement

**Phase 9: Testing and Quality Assurance**

**1. Automated Testing**

* **Task:** Write unit tests for frontend components
  + **Labels:** Feature: Testing, Frontend, Testing
* **Task:** Write unit tests for backend functions
  + **Labels:** Feature: Testing, Backend, Testing
* **Task:** Implement integration tests for API endpoints
  + **Labels:** Feature: Testing, Backend, API

**2. End-to-End Testing**

* **Task:** Set up Cypress for end-to-end testing
  + **Labels:** Feature: Testing, Frontend, Testing
* **Task:** Write test cases simulating user interactions
  + **Labels:** Feature: Testing, Frontend, Testing

**3. Continuous Integration/Continuous Deployment (CI/CD)**

* **Task:** Configure GitHub Actions for CI/CD pipelines
  + **Labels:** Feature: Testing, DevOps, Automation
* **Task:** Automate testing and deployment processes
  + **Labels:** Feature: Testing, DevOps, Automation

**Phase 10: Deployment and Post-Launch Activities**

**1. Prepare for Deployment**

* **Task:** Set up production environment on hosting service
  + **Labels:** Feature: Deployment, DevOps, High Priority
* **Task:** Deploy frontend to hosting service (e.g., Firebase Hosting)
  + **Labels:** Feature: Deployment, DevOps
* **Task:** Deploy backend to cloud platform (e.g., Heroku)
  + **Labels:** Feature: Deployment, DevOps

**2. Post-Launch Monitoring**

* **Task:** Set up application performance monitoring tools
  + **Labels:** Feature: Deployment, DevOps, Monitoring
* **Task:** Configure alerts for critical failures
  + **Labels:** Feature: Deployment, DevOps, Monitoring

**Phase 11: Documentation and Training**

**1. Create User Documentation**

* **Task:** Write user guides and tutorials
  + **Labels:** Feature: Documentation, Content, High Priority
* **Task:** Include screenshots and step-by-step instructions
  + **Labels:** Feature: Documentation, Content

**2. Create Developer Documentation**

* **Task:** Document API endpoints and usage
  + **Labels:** Feature: Documentation, Backend, API, High Priority
* **Task:** Explain architecture and design decisions
  + **Labels:** Feature: Documentation, Backend, Frontend, Architecture

**Phase 12: Advanced Features and Integrations**

**1. Integration with External Services**

* **Task:** Integrate with Google Calendar API
  + **Labels:** Feature: Integration, Backend, API, Google
* **Task:** Integrate with Microsoft Outlook Calendar API
  + **Labels:** Feature: Integration, Backend, API, Microsoft
* **Task:** Integrate with Stripe for payment processing
  + **Labels:** Feature: Integration, Backend, Payment, Security

**Phase 13: Final Review and Project Closure**

**1. Final Testing and Quality Assurance**

* **Task:** Conduct final QA pass to ensure all features meet acceptance criteria
  + **Labels:** Feature: Closure, Testing, High Priority
* **Task:** Perform regression testing
  + **Labels:** Feature: Closure, Testing

**2. Project Documentation and Closure**

* **Task:** Compile project reports summarizing work done
  + **Labels:** Feature: Closure, Documentation, Content
* **Task:** Provide recommendations for future improvements
  + **Labels:** Feature: Closure, Documentation
* **Task:** Hold project retrospective meeting
  + **Labels:** Feature: Closure, Team, Meeting