**Consolidated Agile Sprint Plan**

**Sprint 1: Project Setup & Database (Week 1)**

**Objectives**

* Establish all development tools and repository.
* Design and create the initial database schema.

**Tasks**

1. **Development Environment Setup**
   * Install necessary software (Node.js, MySQL, Git).
   * Configure VS Code (or chosen IDE).
   * Initialize GitHub repository with a README and basic file structure.
2. **Database Schema Design**
   * Create UML diagrams for users and tasks tables (fields, relationships).
   * Finalize database schema attributes (user ID, email, hashed password, task ID, title, description, due date, priority, user ID, etc.).
3. **MySQL Setup**
   * Install & configure MySQL locally or on the chosen server.
   * Create the initial database, tables, and relationships based on the schema.

**Deliverables**

* Working development environment and GitHub repository.
* UML diagram & finalized MySQL database schema.
* (Optional) Database seeded with initial test data.

**Sprint 2: User Authentication & Task CRUD (Weeks 2–3)**

**Objectives**

* Implement core user registration and login functionality.
* Build essential Create/Read/Update/Delete (CRUD) operations for tasks.

**Tasks**

1. **User Authentication (Backend & Frontend)**
   * **POST /register**: Validate and store user data (email, hashed password).
   * **POST /login**: Verify credentials and return session/token.
   * **Frontend Forms**: Create registration/login pages and connect them to the REST endpoints.
2. **Task Management CRUD (Backend & Frontend)**
   * **Backend APIs**:
     + Create Task (POST /tasks),
     + Read Task(s) (GET /tasks),
     + Update Task (PUT /tasks/:id),
     + Delete Task (DELETE /tasks/:id).
   * Ensure only the task owner can modify or delete tasks.
   * **Frontend React Components**:
     + Build forms and pages to display, create, update, and delete tasks.
   * **Security/Validation**:
     + Basic checks (e.g., required fields, valid user session).

**Deliverables**

* Fully functional registration and login endpoints.
* Working frontend user authentication flows.
* CRUD APIs for tasks, integrated with frontend components.

**Sprint 3: UI Design & Security Enhancements (Weeks 4–5)**

**Objectives**

* Improve the overall look-and-feel of the application.
* Reinforce security measures throughout the stack.

**Tasks**

1. **UI/UX Design**
   * Apply consistent styling (CSS/SCSS) across all pages.
   * Enhance forms and layouts for better user experience.
2. **Security Implementation**
   * Add CSRF tokens for form submissions (if using cookies/sessions).
   * Implement server-side input validation to prevent SQL injection and XSS.
   * Review auth flows for any security loopholes and fix accordingly.

**Deliverables**

* Polished UI with a consistent design language.
* Hardened security (CSRF protection, input validation).
* Documentation of security measures taken.

**Sprint 4: Extra Features, Testing & Finalization (Weeks 6–7)**

**Objectives**

* Add advanced features (e.g., task priority, category).
* Conduct final testing (unit, integration, performance, security).
* Prepare for release/deployment.

**Tasks**

1. **Additional Feature Development**
   * Implement task priority levels and categories.
   * Update backend schemas/endpoints to support extra fields.
   * Update frontend to display and filter by priority/category.
2. **Testing & Stabilization**
   * **Unit Tests**: Verify each function/component (using Jest, Mocha, etc.).
   * **Security Tests**: Check for SQL injection, XSS vulnerabilities.
   * **Performance Tests**: Ensure the application can handle expected load.
3. **Deployment & Documentation**
   * Final review of all features.
   * Write project documentation and user guides.
   * Deploy to production or staging environment (if applicable).

**Deliverables**

* New features (priority, categories) integrated seamlessly.
* Comprehensive test results (unit, security, performance).
* Deployment-ready application and documentation.

**Overall Timeline**

1. **Week 1:** Project Setup & Database
2. **Weeks 2–3:** Auth & Task CRUD
3. **Weeks 4–5:** UI Design & Security
4. **Weeks 6–7:** Extra Features, Testing & Finalization