Building a comprehensive project management website with all the features you've listed involves several steps. Below is a basic outline using HTML, CSS, JavaScript, PHP, and MySQL. Keep in mind that this is just a starting point, and you may need to expand, optimize, and secure the code for a production environment.

**1. Database Setup:**

); -- Other necessary tables can be added as needed

**2. Backend (PHP):**

2.1. User Authentication (login.php, register.php):

Implement registration and login functionality.

2.2. Project CRUD (projects.php, project\_details.php):

Create, Read, Update, and Delete operations for projects.

2.3. Comment System (comments.php):

Allow users to post comments on projects.

2.4. File Upload (upload.php):

Enable users to upload project files.

2.5. Integration with ICT Portal (integration.php):

Integrate with the ICT portal using appropriate APIs.

2.6. Admin Panel (admin.php):

Implement an admin panel to manage users, projects, and other functionalities.

**3. Frontend (HTML, CSS, JavaScript):**

3.1. Layout:

Design the layout with HTML and CSS. Utilize a responsive framework like Bootstrap.

3.2. Navigation Bar (navbar.html):

Create a navigation bar with links to different sections.

3.3. Project Feed (feed.html):

Display projects with options for sorting and filtering.

3.4. Student Portfolio (portfolio.html):

Showcase a student's projects and uploaded files.

3.5. Faculty Review (faculty\_review.html):

Enable faculty members to review and comment on projects.

3.6. User Dashboard (dashboard.html):

Display user-specific information, activities, and project status.

3.7. Calendar (calendar.html):

Integrate a calendar for scheduling project timelines.

3.8. Interactive Elements (comments, likes, shares):

Implement JavaScript for interactive features like comments, likes, and shares.

3.9. Embedded API Viewer (api\_viewer.html):

Integrate an API viewer for external content.

3.10. Privilege Status (privilege\_status.html):

Display user privilege status.

3.11. Live Stream/Project Fair (live\_stream.html):

Implement features for live streaming and project fairs.

**4. Additional Features:**

4.1. Chat Bot (chat\_bot.php):

Implement a virtual chat bot for assistance.

4.2. Time-Monitored Proposal (proposal.php):

Enforce a 15-day window for faculty approval and student input.

4.3. User/Viewer Statistics (statistics.php):

Track and display user and viewer statistics.

4.4. Outreach/Newsletter (newsletter.php):

Implement an outreach and newsletter system.

**5. Security:**

5.1. Input Validation and Sanitization:

Ensure all user inputs are validated and sanitized to prevent SQL injection and XSS attacks.

5.2. Session Management:

Secure user sessions and implement proper session management.

5.3. Password Hashing:

Use password hashing for storing passwords in the database.

5.4. HTTPS:

Ensure your website is served over HTTPS for secure communication.

**6. Deployment:**

Deploy the website on a secure server and configure necessary server settings.

This is a high-level outline, and you'll need to dive into each section for detailed implementation. Be aware of security considerations and always validate and sanitize user inputs to prevent common vulnerabilities.