

# EPICs & Sprint Plan

## DAU Campus Complaints Resolve

---

### Sprint 1 (1 Week) – User Authentication & Setup

**Epic:** User Authentication

**P.O.C for Implementation** - Rajdeep Patel (202301092)

- **Frontend:**
  - Build login/register UI with role selection (Student, Committee, Admin).
  - Redirect users to role-based dashboards after login.
- **Backend:**
  - Implement authentication APIs with JWT/session.
  - Role-based access control (RBAC).
  - Password hashing.
- **Database:**
  - Create Users & Roles tables.

Done when : Secure login works. LLM API connected with test prompt.

---

### Sprint 2 (1.5 Weeks) – Complaint Submission

**Epic:** Complaint Submission

- **Frontend:**
  - Complaint form UI (title, description, file upload, anonymous toggle).
  - Form validation + confirmation message.
- **Backend:**
  - API to accept complaint submissions.

- Handle file uploads (store file path in DB).
- Generate complaint ID.
- **AI :**
  - Set up API integration environment (Gemini).
  - Design prompt templates for classification.
  - Return JSON response: {category, priority}.
  - Integrate into backend pipeline → every new complaint calls LLM API.
  - Use similarity to assist classification (if complaint resembles a past one, reuse that category/priority).
- **Database:**
  - Create Complaints table (id, title, desc, user\_id, status, category, priority, file\_path).

Done when : Students can submit complaints, which get auto-tagged (category + priority) by LLM.

---

## Sprint 3 (1 Week) – Complaint Tracking & Dashboards

**Epic:** Complaint Tracking

- **Frontend:**
  - Student dashboard: list of submitted complaints with statuses.
  - Committee dashboard: assigned complaints with update buttons.
  - Admin dashboard: all complaints list.
- **Backend:**
  - GET APIs for complaint lists (student, committee, admin).
  - PUT API for status updates (Pending → In-Process → Resolved).
  - Integrate LLM classification output into complaint routing.
- **AI :**
  - Refine prompts with few-shot examples (eg, sample campus complaints).
  - Improve output reliability with formatting checks.
- **Database:**
  - Add Status column.

- Create Committee mapping table.

Done when : Complaints route automatically to the correct committee via LLM API.

---

## **Sprint 4 (1 Week) – Notifications & Escalations**

**Epic:** Notifications

- **Frontend:**
  - Notification panel (in-app alerts).
  - Complaint timeline visualization.
- **Backend:**
  - Email notifications on complaint submission & updates.
  - Escalation rules (auto-flag if overdue).
- **AI :**
  - Auto-escalate if text is strongly negative.
  - Add escalation flag in response.

Done when : Notifications work, escalations triggered automatically by deadlines or sentiment.

---

## **Sprint 5 (1 Week) – Analytics & Dashboards**

**Epic:** Analytics

- **Frontend:**
  - Analytics for committee dashboard (complaint count, resolution time).
  - Analytics for admin (complaints by category, priority, committee).
  - Student feedback form after resolution.
  - Filters (status, priority, committee).
- **Backend:**
  - APIs for analytics (counts, trends).
  - Feedback submission API.

- Search/filter endpoints.
- **AI :**
  - Add feedback loop → when committee/admin overrides LLM's classification, store override in DB.
  - For future calls, inject past overrides into prompt as context.

Done when : Dashboards functional with analytics; AI learns from overrides via prompt updates.

---

## Sprint 6 (1 Week) – Reports & Export

**Epic:** Reports

- **Frontend:**
  - Report export UI for Admin & Committees.
- **Backend:**
  - Report generation (CSV/PDF).
  - Export API.
  - Privacy handling (mask anonymous complaints).
- **Database:**
  - Reports table.

Done when : Reports downloadable. AI can leverage past complaints.

---

## Sprint 7 (1 Week) – Security & Developer Operations

**Epic:** Reliability & Security

- **Frontend:**
  - Responsive fixes.
  - Error handling & loading states.

- **Backend:**
  - Secure file storage.
  - Logging, monitoring, rate-limiting.
- **AI :**
  - Optimize API call usage.
  - Monitor LLM classification accuracy via stored feedback logs.
- **Database:**
  - Data encryption.
  - Performance optimization.

Done when : Secure, monitored system; LLM usage optimized.

---

## **Sprint 8 (1 Week) – Final Polish & Deployment**

**Epic:** Finalization

- **Frontend:**
  - UI polish with DAU branding.
  - Add Help & Support page.
- **Backend:**
  - Final bug fixes.
  - Deploy backend + LLM integration to server.
- **AI :**
  - End-to-end testing: Complaint → API classification → Routing → Dashboard.

Done when : Fully functional prototype deployed, demo-ready.

---