Smart Student Helpdesk – Full Explanation

Phase 1: Problem Understanding & Industry Analysis

Goal: Understand why we're building the Student Helpdesk and how it solves real academic problems.

1. **Requirement Gathering**
   - **Talk to stakeholders**: Students, Faculty, Exam Cell, Hostel Warden, Admin.
   - **Example requirements**:
     - Students can raise queries (Admissions, Exams, Fees, Hostel).
     - Faculty/Departments can resolve queries.
     - No query should remain unresolved for long.
     - Students should get AI-powered instant answers for FAQs.
2. **Stakeholder Analysis**
   - **Admin**: Manages system setup, monitoring.
   - **Faculty/Departments**: Resolve queries assigned to them.
   - **Students**: Submit queries, check status, get AI replies.
   - **Management**: Monitor reports & resolution efficiency.
3. **Business Process Mapping**
   - **Flow**: Student submits query → AI suggests instant answer → If unresolved, Case created → Assigned to department → Faculty resolves → Student notified → Dashboard updates.
4. **Education-Specific Use Case Analysis**
   - Colleges have thousands of queries daily (fees, admissions, exams, hostels).
   - Many queries are repetitive → **AI can auto-reply to FAQs**.
   - Escalation is needed if queries are pending.

5. **AppExchange Exploration**
    - Existing “Helpdesk” apps exist, but none tailored for **student-centric + AI FAQ**.
    - We’ll build our **own custom Smart Student Helpdesk**.

Phase 2: Org Setup & Configuration

Goal: Prepare Salesforce environment for Helpdesk.

1. **Salesforce Edition** – Developer Org (Free).
2. **Company Profile Setup** – Add College info, time zone, currency.
3. **Business Hours & Holidays** – College working hours (9AM–5PM), holidays.
4. **User Setup & Licenses** – Create users: Student, Faculty, Admin.
5. **Profiles** – Faculty Profile (restricted), Admin Profile (full access).
6. **Roles** – Admissions, Exams, Hostel, Accounts.
7. **Permission Sets** – For extra access like Reports.
8. **OWD (Org-Wide Defaults)** – Cases = Private.
9. **Sharing Rules** – Share queries with correct department.
10. **Login Access Policies** – Secure faculty logins.

Phase 3: Data Modeling & Relationships

Goal: Build student query data structure.

1. **Standard & Custom Objects**
    - **Standard:** Contact (Student), Case (Query).
    - **Custom:** FAQ__c (Knowledge Base).
2. **Fields**

- **Case**: Query Type, Priority, Resolution Notes.
- **FAQ**: Question, Answer, Category.

3. **Record Types**
   - **Cases**: Admissions, Exams, Hostel, Fees.
4. **Page Layouts**
   - Student sees query status + AI suggestions.
   - Faculty sees assigned cases with details.
5. **Relationships**
   - Case → Contact (Student).
   - Case → Department (optional custom).

Phase 4: Process Automation (Admin)

Goal: Automate query routing & notifications.

1. **Validation Rules** – Ensure all fields are filled.
2. **Flow Builder**
   - Auto-assign cases to correct department.
   - If query unresolved after 3 days → escalate to Admin.
   - AI Flow: Check FAQ database → Suggest AI answer instantly.
3. **Approval Process** – Urgent exam queries auto-escalated.
4. **Email Alerts** – Students notified on query creation/resolution.
5. **Custom Notifications** – Faculty notified on new case.

Phase 5: Apex Programming (Developer)

Goal: Add advanced query logic & AI.

1. **Triggers** – Auto-set priority based on query type.
2. **AI FAQ Handler Class** – Match student queries to FAQ answers.

```
public with sharing class StudentAIHandler {
    @AuraEnabled(cacheable=true)
    public static String getAIResponse(String studentQuery){
        List<FAQ__c> faqs = [SELECT Question__c, Answer__c FROM FAQ__c];
        for(FAQ__c f : faqs){
            if(studentQuery.toLowerCase().contains(f.Question__c.toLowerCase())){
                return f.Answer__c;
            }
        }
        return 'No direct answer found. Query forwarded to faculty.';
    }
}
```

3. **Scheduled Apex** – Daily unresolved queries report.
4. **Queueable Apex** – Reassign escalated cases.
5. **Test Classes** – Validate 75%+ coverage.

Phase 6: User Interface Development

Goal: Make it student-friendly.

1. **Lightning App Builder** – “Smart Student Helpdesk” App.
2. **Tabs** – Student Queries, FAQ, Reports.
3. **LWC: Student Query Form**
    - Enter Name, Email, Query Type, Description.
    - AI suggests answers instantly.
    - If not satisfied → Case created.
4. **LWC: Query History** – Students view past queries.
5. **Navigation Service** – Student redirected to their case page.

Phase 7: Integration & External Access

Goal: Expand access to students.

1. **Web-to-Case** – Embed query form in college website.
2. **Email-to-Case** – Students can email queries directly.
3. **External AI Integration (Optional)** – Einstein GPT or OpenAI API for smarter answers.
4. **Named Credentials** – Secure API keys.

Phase 8: Data Management & Deployment

Goal: Manage student data & queries.

1. **Data Import Wizard** – Import student details.
2. **Data Loader** – Bulk upload queries.
3. **Duplicate Rules** – Prevent duplicate cases.
4. **Data Export & Backup** – Weekly backup of cases & FAQs.
5. **Change Sets / SFDX** – Deploy configurations.

Phase 9: Reporting, Dashboards & Security

Goal: Monitor query trends.

1. **Reports**
   - Queries by Type, Dept, Resolution Time.
   - FAQ Effectiveness Report (auto-resolved vs manual).
2. **Dashboards**
   - Department-wise query load.
   - AI Response Success % Dashboard.
3. **Dynamic Dashboards** – Faculty see only their queries.
4. **Field Level Security** – Students can't see internal notes.
5. **Audit Trail** – Track faculty changes.

Phase 10: Final Presentation & Demo Day

Goal: Deliver like a real-world project.

1. **Pitch Presentation** – Problem → Solution → AI-powered Helpdesk.
2. **Demo Walkthrough** – Submit query → AI suggests answer → Case auto-created → Faculty resolves → Dashboard shows reports.
3. **Documentation** – User guide, setup steps.
4. **Portfolio Showcase** – Add to LinkedIn, GitHub.