Phase 3: Core MVP Feature Implementation

Smart Course Registration
Chrome Extension

Karo Takhleeq 2026 Hackathon

Day 2 — 9:00 AM – 12:30 PM

Team Members:

Muhammad Ahmad – Lead Developer
Syed Mohammad Hussain Bukhari – Frontend & UX
Abdul Raffay Naeem – Research & Testing

January 2026

Contents

1 Executive Summary

Phase 3 focused on implementing the **core problem-solving functionality** of our Smart Course Registration extension. The MVP is now fully functional, allowing students to scan courses, set preferences, and generate optimized conflict-free timetables with a single click.

2 Objectives Achieved

Objective	Status
Core problem-solving feature implementation	Complete
End-to-end user flow working	Complete
Critical bugs fixed	Complete
MVP performs intended function	Complete

Table 1: Phase 3 Objectives Status

3 Core Features Implemented

3.1 Course Scanning System

- **DOM Scraping:** Content script extracts all course sections from the portal
- **Data Extraction:** Course code, name, section ID, status (open/closed), schedule
- **Real-time Detection:** Automatically identifies available sections

3.2 Smart Filtering System

Filter	Description
Preferred Days	Select which days you want classes (Mon-Fri)
Time Range	Set start/end time limits (8 AM - 8 PM)
Max Campus Days	Limit to 2, 3, 4, or 5 days
Max Gap / No Gap	Control gaps between classes

Table 2: Available Filtering Options

3.3 Optimization Algorithm

The optimization algorithm processes schedules through the following steps:

1. **Filter sections** by status and time/day preferences
2. **Generate combinations** (max 50,000)

3. **Detect and eliminate** conflicting schedules
4. **Score remaining schedules:**
   - Fewer days = Higher score
   - Fewer gaps = Higher score
   - Meeting preferences = Bonus points
5. **Output:** Top 10 ranked timetables

3.4 Gap Calculation & No-Gap Feature

- **Gap Detection:** Calculates total gap hours between consecutive classes
- **No Gap Option:** When selected, only shows schedules with back-to-back classes
- **Visual Indicator:** Result cards show gap hours (e.g., "0h gaps")

3.5 Schedule Application

- One-click enrollment of selected schedule
- Sequential section enrollment with confirmation
- Visual feedback on success/failure

4 Testing & Verification

4.1 Test Scenarios Completed

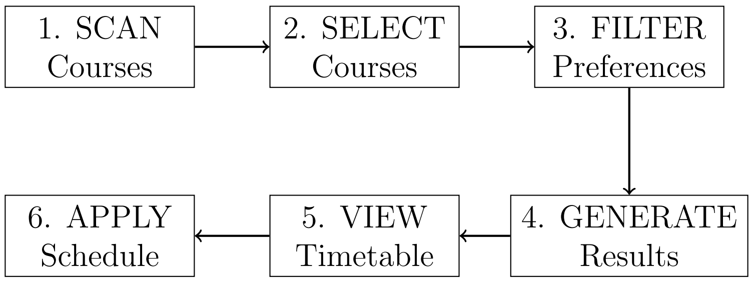
Test Case	**Expected Result**	**Status**
Scan courses from portal	All sections extracted	Pass
Select multiple courses	Courses added to selection	Pass
Generate with "No Gap"	Only gap-free schedules shown	Pass
Generate with time filter	Sections outside range excluded	Pass
Apply schedule	All sections enrolled	Pass
Conflict detection	Overlapping sections rejected	Pass

Table 3: Test Results

4.2 Test Data

- **Professional Practices** + **Parallel Computing**: Special no-gap sections created
- Back-to-back schedule: 8:00-10:00 → 10:00-12:00 on Mon/Wed

5 User Flow Diagram



6 Bugs Fixed

Bug	**Resolution**
Gap calculation returning 0 always	Implemented proper gap calculation algorithm
”No Gap” option not filtering	Added filter logic for maxGap === -1
Scores not reflecting gaps	Updated scoring to penalize gaps

Table 4: Bug Fixes

7 Performance Metrics

Metric	**Value**
Max combinations processed	50,000
Sections per course limit	8 (auto-reduced if exceeded)
Valid schedules returned	Top 10
Processing time	¡ 2 seconds

Table 5: Performance Metrics

8 MVP Stability Status

Criteria	**Assessment**
Core Function Works	Users can generate optimized schedules
No Critical Bugs	All major issues resolved
Demo Ready	Stable for demonstration
User Flow Complete	End-to-end functionality verified

Table 6: MVP Stability Assessment

9 Team Contributions

Member	**Phase 3 Contribution**
Muhammad Ahmad	Algorithm implementation, bug fixes
Syed Mohammad Hussain Bukhari	UI updates, gap feature implementation
Abdul Raffay Naeem	Testing, documentation

Table 7: Team Contributions

10 Next Steps (Phase 4)

1. UI polish and visual enhancements
2. Error handling improvements
3. Demo preparation and rehearsal
4. Final testing with demo portal

Phase 3 Completed Successfully
MVP is functional and ready for demonstration