KARO TAKHLEEQ 2026

IEEE CS × IEC × TAKHLEEQ

PHASE 4 SUBMISSION

Supporting Features, UI/UX & Evaluation

Smart Course Registration Chrome Extension

MVP Complete   UI Polished   Demo Ready   Tested   Documented

Team Members

Muhammad Ahmad
Syed Mohammad Hussain Bukhari
Abdul Raffay Naeem

`github.com/Syed-Mohammad-Hussain-Bukhari/Hackathon`

University of Central Punjab
Faculty of Information Technology | January 2026

Contents

1. Executive Summary

Phase 4 focused on **polishing the UI/UX**, adding **supporting features**, and preparing for the **final demonstration**. Our Smart Course Registration extension is now a fully functional MVP that transforms the 45+ minute registration process into under 5 minutes.

Mission Accomplished

Before	→	**After**
45+ minutes of frustration		Under 5 minutes
Manual conflict checking		Automatic detection
5-day scattered schedule		Optimized 3-day timetable
No gap control		Zero-gap option available

2. Supporting Features Added (Phase 4)

★ Backtracking Search Algorithm

Replaced brute-force combination generation with an intelligent **backtracking algorithm** that:

- Prunes invalid branches early (conflict detection at each step)
- Finds valid schedules without generating all combinations
- Handles up to 500,000 search steps efficiently
- Returns up to 100 valid schedules in under 2 seconds

★ Conflict Detection & Resolution

When conflicts are detected, the system now:

- Shows **which courses conflict** with each other
- Displays the **exact day and time** of the clash
- Attempts **partial scheduling** by removing problematic courses
- Provides clear resolution suggestions

★ Side-by-Side Comparison View

Users can now compare top timetable options:

- View multiple mini-timetables on one screen
- Quick visual comparison of days and gaps
- Click any option to view full details

Enrollment Cart Integration

- View already enrolled courses in the extension
- Toggle between "All Courses" and "Enrollment Cart" views
- Visual indication of enrolled sections

Closed Sections Watcher

New feature to track full/closed sections:

- **View Closed** button shows all closed sections
- **Notify Me** button adds section to watchlist
- Content script receives `watch` action and stores section
- Visual feedback: button changes to “ Watching”
- Prepares for future auto-enrollment when section opens

3. UI/UX Improvements

3.1 Extension Popup Enhancements

Gradient headers with blue theme

Interactive chip selection for days

Stats row with course/section counts

Warning page for excluded courses

Comparison view for schedules

Result cards with days + gaps info

3.2 Portal Prototype Enhancements

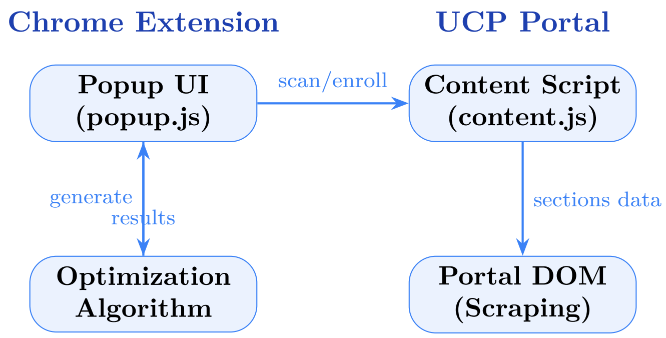
Timetable grid visualization

Filter buttons (Cart, Timetable, Closed)

Closed section watcher notifications

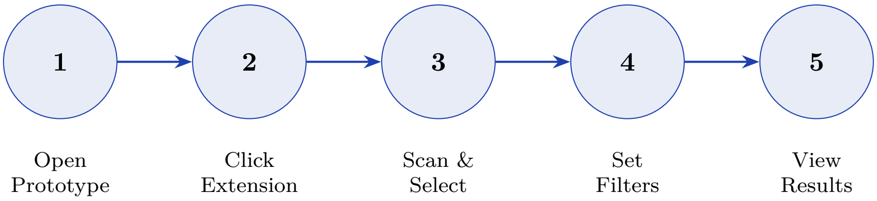
Color-coded course cells

4. Technical Architecture (Final)



4.1 Algorithm Flow

5. Demo Flow & Presentation Storyline



Demo Script:

1. **Problem Statement** (30s): "45+ minutes to register, conflicts, gaps..."
2. **Solution Overview** (30s): "Chrome extension that optimizes automatically"
3. **Live Demo** (3 min):
   - Open prototype portal
   - Click extension icon
   - Scan → Select 4-5 courses
   - Set preferences (3 days, no gap)
   - Show top results
   - View timetable grid
   - Compare options side-by-side
4. **Technical Highlights** (1 min): Backtracking algorithm, conflict resolution
5. **User Research** (30s): Survey results validation

6. Final Testing Results

Test Case	**Expected**	**Result**
Scan all courses from portal	Extract 18 courses, 270 sections	PASS
Generate with "No Gap" filter	Only gap-free schedules	PASS
Handle too many combinations	Auto-limit, still return results	PASS
Detect and show conflicts	Display conflicting courses	PASS
Partial scheduling on conflict	Remove problematic course	PASS
Side-by-side comparison	Show mini-grids	PASS
Apply schedule to cart	Click enroll buttons	PASS
View closed sections	Display all full/closed sections	PASS
Watch closed section	Add to watchlist, show confirmation	PASS

7. Repository Structure

GitHub Repository

```
github.com/Syed-Mohammad-Hussain-Bukhari/Hackathon/
|-- extension/
|   |-- manifest.json
|   |-- background.js
|   |-- popup/
|   |   |-- popup.html (225 lines)
|   |   |-- popup.css (520 lines)
|   |   |-- popup.js (978 lines)
|   |-- content/
|   |   |-- content.js
|   |   |-- content.css
|   |-- icons/
|-- prototype/
|   |-- index.html
|   |-- styles.css
|   |-- app.js (525 lines)
|   |-- data.js (18 courses, 270 sections)
|-- Phase1_Report/main.tex
|-- Phase2_Report/main.tex
|-- Phase3_Report/main.tex
|-- Phase4_Report/main.tex
|-- docs/
|   |-- USER_GUIDE.md
|   |-- TECHNICAL.md
|-- README.md
```

8. Evaluation Criteria Alignment

Criteria	Our Solution	Score
Depth of problem understanding	Survey of 30+ students, validated assumptions	
Core MVP functionality	Scanning, filtering, optimization, conflict detection	
User value & real-world relevance	45 min → 5 min, addresses real UCP problem	
Technical execution	Backtracking algorithm, 906+ lines of JS	
Quality of UI/UX	Polished popup, gradient design, comparison view	
Startup/incubation potential	Scalable to other universities	

9. Team Contributions (Phase 4)

Member	Contribution
Muhammad Ahmad	Backtracking algorithm, conflict detection, partial scheduling
Syed M. Hussain Bukhari	Comparison view, UI polish, enrollment cart, prototype filters
Abdul Raffay Naeem	Testing, documentation, demo preparation

10. Future Roadmap

Auto-enrollment (one-click)

Waitlist notifications

Multi-semester planning

Mobile companion app

Deploy to other universities

Phase 4 Complete

`github.com/Syed-Mohammad-Hussain-Bukhari/Hackathon`

Ready for Final Demonstration