

# User Manual

## Automated Time Table & Exam Scheduling System



**Developed by:** BumbleBee

### **Team Members:**

Raskin Verma

Udit Dadhich

Sindhu Telari

Vishwa D.

**Course:** Software Design Tools and Techniques

**Institution:** Indian Institute of Information Technology, Dharwad

**Date:** October 2025

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Folder Structure</b>	<b>2</b>
<b>3</b>	<b>Running the Schedulers</b>	<b>3</b>
3.1	Running the Academic Time Table Scheduler . . . . .	3
3.2	Running the Exam Scheduler . . . . .	4
<b>4</b>	<b>Adding New Data Files</b>	<b>4</b>
4.1	Adding a New Department (Academic Timetable) . . . . .	4
4.2	Adding a New Exam Dataset (Exam Timetable) . . . . .	5
4.3	Important Notes . . . . .	5
<b>5</b>	<b>Sample Input Files</b>	<b>5</b>
<b>6</b>	<b>FAQ</b>	<b>6</b>
<b>7</b>	<b>Future Enhancements</b>	<b>6</b>

## 1 Overview

The **Automated Scheduling System** for IIIT Dharwad contains two major modules:

1. **Academic Time Table Scheduler**
2. **Exam Time Table Scheduler**

Both systems read data from CSV files and generate well-formatted Excel timetables with conflict avoidance.

### Academic Time Table Features

- Automatic lecture + lab scheduling.
- Faculty conflict detection.
- Room availability constraints.
- First-Half / Second-Half semester logic.
- Color-coded Excel output.
- Faculty-wise consolidated timetable.

### Exam Scheduler Features

- Automatically generates clash-free exam timetables.
- Ensures one exam per branch per day.
- Supports multiple exam slots (Morning/Evening).
- Automatic room allocation.
- Invigilation assignment generation.
- Generates per-branch and combined exam sheets.

## 2 Folder Structure

The project folder structure (as in repository):

```
AUTOMATED - TIME - TABLE - SCHEDULING - FOR - IIIT - DHARWAD /  
|  
+-- data/  
|    +-- CSE_3_courses-A.csv  
|    +-- CSE_3_courses-B.csv  
|    +-- CSE_1_A_courses.csv  
|    +-- CSE_1_B_courses.csv  
|    +-- CSE_5_courses.csv  
|    +-- All_7_courses.csv  
|    +-- DSAI_3_courses.csv  
|    +-- ECE_3_courses.csv  
|    +-- rooms.csv  
|    +-- timeslots.csv  
|  
+-- data/exam_data/  
|    +-- CSE_1.csv  
|    +-- CSE_3.csv  
|    +-- DSAI_3.csv  
|    +-- ECE_3.csv  
|    +-- All_7_courses.csv  
|    +-- rooms.csv  
|    +-- timeslots.csv  
|    +-- faculty.csv  
|  
+-- timetable_automation/  
|    +-- main.py          # Academic Scheduler  
|    +-- exam.py          # Exam Scheduler  
|    +-- __init__.py  
|  
+-- docs/  
+-- tests/  
+-- README.md
```

### 3 Running the Schedulers

#### 3.1 Running the Academic Time Table Scheduler

From the project root directory:

```
python -m timetable_automation.main
```

This generates:

- <DEPT>\_timetable.xlsx
- faculty\_timetable.xlsx
- <DEPT>\_unscheduled\_courses.xlsx

### 3.2 Running the Exam Scheduler

From the project root directory:

```
python -m timetable_automation.exam
```

This produces:

- exam\_timetable.xlsx
- unscheduled\_exams.xlsx
- invigilation\_assignments.xlsx
- Branch-wise exam timetables

## 4 Adding New Data Files

### 4.1 Adding a New Department (Academic Timetable)

**Step 1:** Place the new CSV file inside:

```
data/
```

Example:

ECE\_5\_courses.csv

**Step 2:** Open `timetable_automation/main.py` and add it to the dictionary :

```
departments = {
    "CSE-3-A": "data/CSE_3_courses-A.csv",
    "CSE-3-B": "data/CSE_3_courses-B.csv",
    ...
    "ECE-5": "data/ECE_5_courses.csv",      # <-- Add this
}
```

**Step 3:** Run:

```
python -m timetable_automation.main
```

---

## 4.2 Adding a New Exam Dataset (Exam Timetable)

Step 1: Place the new file inside:

```
data/exam_data/
```

Example:

CSE\_5.csv

Step 2: Modify timetable<sub>automation</sub>/exam.py :

```
departments = {
    "CSE-3": "data/exam_data/CSE_3.csv",
    "CSE-5": "data/exam_data/CSE_5.csv",      # <-- Add new file
}
```

Step 3: Run:

```
python -m timetable_automation.exam
```

---

## 4.3 Important Notes

- File names can be anything, but keys should be meaningful: "CSE-5", "ECE-3" etc.
- The scheduler does not auto-detect new files | they must be added in code.
- Each dataset must include valid columns: Course\_Code, Course\_Title, Faculty.
- Exam scheduler requires: rooms.csv, faculty.csv, timeslots.csv in exam\_data/.

## 5 Sample Input Files

Academic: timeslots.csv

Start_Time	End_Time
07:30	09:00
09:00	10:00
10:00	10:30
10:30	12:00
14:00	15:30

## Exam: timeslots.csv

Start_Time	End_Time
09:00	12:00
14:00	17:00

## 6 FAQ

Q1. Why is a course/exam unscheduled?

Usually because:

- Room unavailable
- Faculty clash
- Slot clash
- Max exams per day limit

Q2. How to change break times?

Edit:

```
self.excluded_slots = [...]
```

Q3. How to add more exam slots?

Modify:

```
data/exam_data/timeslots.csv
```

## 7 Future Enhancements

- Automatic student clash matrix generation.
- Seating arrangement generator.
- Web UI for uploading CSV and downloading timetables.
- Google Calendar integration.