CODE:

```
class Timetable:
    def __init__(self):
        self.schedule = []  # Stores all scheduled classes

    def add_class(self, course, instructor, batch, section, day, start_time,
end_time, room):
        # Convert time to minutes for easier comparison
        def time_to_minutes(time_str):
            hours, minutes = map(int, time_str.split(":"))
            return hours * 60 + minutes

        start_time_mins = time_to_minutes(start_time)
        end_time_mins = time_to_minutes(end_time)

        # Define time constraints
        day_start = 540  # 09:00 AM
        day_end = 1050  # 05:30 PM
        lunch_break_start = 720  # 12:00 PM
        lunch_break_end = 780    # 01:00 PM

        # Ensure class is within working hours
        if start_time_mins < day_start or end_time_mins > day_end:
            print(f"Invalid time: {course} for {batch}-{section} must be scheduled
between 09:00 and 17:30.")
            return

        # Ensure lunch break is free
        if not (end_time_mins <= lunch_break_start or start_time_mins >=
lunch_break_end):
            print(f"Lunch break conflict: {course} for {batch}-{section} on {day}
cannot be scheduled between 12:00-1:00.")
```

```
            return

        # Check for conflicts
        for entry in self.schedule:
            if entry['day'] == day:
                entry_start = time_to_minutes(entry['start_time'])
                entry_end = time_to_minutes(entry['end_time'])

                # Section conflict check (No overlapping classes for the same section)
                if entry['batch'] == batch and entry['section'] == section:
                    if not (end_time_mins <= entry_start or start_time_mins >= entry_end):
                        print(f"Conflict: {batch}-{section} already has another class scheduled at this time. Please choose a different time.")
                        return

                # Room conflict check
                if entry['room'] == room and not (end_time_mins <= entry_start or start_time_mins >= entry_end):
                    print(f"Sorry..! The room {room} is not available for {course} on {day}. Please select another room.")
                    return

                # Instructor conflict check
                if entry['instructor'] == instructor and not (end_time_mins <= entry_start or start_time_mins >= entry_end):
                    print(f"Instructor {instructor} is not available for {course} on {day}.")
                    return

        # Add class to schedule
        self.schedule.append({
```

```
            'course': course,
            'instructor': instructor,
            'batch': batch,
            'section': section,
            'day': day,
            'start_time': start_time,
            'end_time': end_time,
            'room': room
        })
        print(f"Class {course} scheduled for {batch}-{section} in Room {room}
on {day} from {start_time} to {end_time}.")

    def display_schedule(self):
        def time_to_minutes(time_str):
            hours, minutes = map(int, time_str.split(":"))
            return hours * 60 + minutes

        print("\nComplete Timetable:")
        for entry in sorted(self.schedule, key=lambda x: (x['day'],
time_to_minutes(x['start_time']))):
            print(f"{entry['course']} | {entry['instructor']} |
{entry['batch']}-{entry['section']} | {entry['day']} | {entry['start_time']} -
{entry['end_time']} | Room {entry['room']}")

    def get_user_input(self):
        while True:
            course = input("Enter course name (or 'exit' to stop): ")
            if course.lower() == 'exit':
                break
            instructor = input("Enter instructor name: ")
            batch = input("Enter batch name: ")
            section = input("Enter section name: ")
            day = input("Enter day: ")
```

```
        start_time = input("Enter start time (HH:MM): ")
        end_time = input("Enter end time (HH:MM): ")
        room = input("Enter room number: ")
        self.add_class(course, instructor, batch, section, day, start_time,
end_time, room)


# Example Usage
timetable = Timetable()
timetable.get_user_input()
timetable.display_schedule()
```

OUTPUT:

```
Enter course name (or 'exit' to stop): dbms
Enter instructor name: sumalatha
Enter batch name: cse
Enter section name: e
Enter day: monday
Enter start time (HH:MM): 09:00
Enter end time (HH:MM): 10:00
Enter room number: 102
Class dbms scheduled for cse-e in Room 102 on monday from 09:00 to 10:00.
Enter course name (or 'exit' to stop): web
Enter instructor name: balvendra
Enter batch name: cse
Enter section name: h
Enter day: monday
Enter start time (HH:MM): 09:00
Enter end time (HH:MM): 11:00
Enter room number: 102
Sorry..! The room 102 is not available for web on monday. Please select
    another room.
Enter course name (or 'exit' to stop):
```

```
Enter course name (or 'exit' to stop): python
Enter instructor name: k.narayan
Enter batch name: cse
Enter section name: e
Enter day: monday
Enter start time (HH:MM): 09:00
Enter end time (HH:MM): 11:00
Enter room number: 101
Class python scheduled for cse-e in Room 101 on monday from 09:00 to 11:00.
Enter course name (or 'exit' to stop): web
Enter instructor name: balvendra
Enter batch name: cse
Enter section name: e
Enter day: monday
Enter start time (HH:MM): 09:00
Enter end time (HH:MM): 10:00
Enter room number: 102
Conflict: cse-e already has another class scheduled at this time. Please
    choose a different time.
Enter course name (or 'exit' to stop):
```

```
Enter course name (or 'exit' to stop): python
Enter instructor name: k,narayan
Enter batch name: cse
Enter section name: e
Enter day: monday
Enter start time (HH:MM): 13:00
Enter end time (HH:MM): 14:00
Enter room number: 611
Lunch break conflict: python for cse-e on monday cannot be scheduled between
    1:00-2:00.
Enter course name (or 'exit' to stop):
```

```
Enter course name (or 'exit' to stop): python
Enter instructor name: k.narayan
Enter batch name: cse
Enter section name: e
Enter day: monday
Enter start time (HH:MM): 15:00
Enter end time (HH:MM): 17:00
Enter room number: 611
Class python scheduled for cse-e in Room 611 on monday from 15:00 to 17:00.
Enter course name (or 'exit' to stop):
```