NURSE ROSTERING PROBLEM

A PROJECT

submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Akhil Goel (141357)

Under the supervision of

Dr. Pradeep Kumar Singh

To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology

Waknaghat, Solan-173234,

Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **" NURSE ROSTERING PROBLEM"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2017 to May 2018 under the supervision of **Dr. Pradeep Kumar Singh** (Assistant Professor (Senior Grade),Computer Science Department).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Akhil Goel (141357)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pradeep Kumar Singh

Assistant Professor (Senior Grade)

Computer Science & Engineering

Dated:

ACKNOWLEDGEMENT

We owe our profound gratitude to our project supervisor **Dr. Pradeep Kumar Singh**, who took keen interest and guided us all along in the project work titled – "**Nurse Scheduling Problem**". We take this opportunity to express deep regards to our guide for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The in-time facilities provided by the Computer Science department throughout the project development are also equally acknowledgeable. At the end I would like to express my sincere thanks to all my friends and others who helped me directly or indirectly during this project work. The project development helped us in research and we got to know a lot of new things in the domain.

We are really thankful to him.

Akhil Goel, 141357

CONTENTS

Abstract

The Nurse scheduling problem (NSP) is to the problem for studying and analysing the scheduling process in propose and practice heuristics and models to improve both the operation and the quality of the resulting schedule. Nurses must get the benefit from this of getting some high-quality schedules and moreover administrators must get minimum total cost so that he can enjoy the benefits of optimization tools in order to solve the problems related healthcare,

The objective is to maximize the decency of the schedule by optimizing the whole process. It represents a hard class of Multi-objective optimization problems comprising of several snooping objectives between the individual nurses and hospitals. The objective of this is to search difficulties that occur during the process of finding the solution of NSP.

CHAPTER 1

INTRODUCTION

1.1 Introduction

It is a combinatorial optimization problem. Solving a real-world NRP manually **typically needs your high amount of time and price**. For the hospital administration, due to its motivation and complex nature to deduce automated nurse scheduling system, many researches in this area are performed over the years. Burke et al, Cheang et al. and Den Bergh Jorne have conducted surveys on **numerous strategies** that **are used** for **finding** the NRP. Some of these methods are specifically used to solve real-world NSPs. It is noted from the literature that the precise solutions haven't been achieved by these strategies because of the combinatorial nature of this problem. Clearly, the analysis during this domain continues to be open associate degreed presenting an economical technique to tackle this drawback are often thought of a big accomplishment.

Numerous real-world datasets investigated by the researchers to check their strategies are adopted from completely different hospitals. Some of them are sampled from Australia; UK Hospital; Riyadh Al- Kharj hospital, Vienna hospital, Saudi Arabia; Canada hospitals; Chinese hospitals; Auckland hospitals, New Zealand; Malaysia hospitals; US hospitals; German hospitals; ‘‘ORTEC’’ dataset from Dutch Hospital, Netherlands; Queen’s Medical Centre University Hospital Trust (QMC) Nottingham, UK; York Hospital in York, Lisbon hospital, Portugal and ‘‘NSPLib’’ dataset that comes from Belgian hospital. However, as a result of totally different constraints and mathematical formulations of the real-world datasets, it's nearly impossible for the researchers within the domain to check the results of their methods. Accordingly, the INRC2010 has gone to the fore.

In order to standardize the NRP dataset, the CODeS analysis cluster at SINTEF in Norway and also the University of Udine, Katholieke Universities Leuven in Belgium in Italy organized the primary international nurse scheduling competition (INRC2010). The dataset of this competition consists of sixty-nine instances which may reflect many cases of the real-world scenario that are varied in complexity and size. This dataset is chosen for the research because of its adoption by the nurse scheduling researchers to compare and evaluate their methods.

1.2 Problem Statement

The **nurse scheduling problem** is the best method for finding a way to assign nurses to their respective shifts, generally with a set of soft and hard constraints that all valid solutions pursue, and which explains the relative quality of all valid solutions.

The soft and hard constraints are as follows:

Hard constraints

- All demanded shifts allocated to a nurse.
- A nurse can work with only a single shift per day.
- The least number of nurses required for the shift.
- The total no. of working days for the nurse should be between the maximum and minimum range.
- A day shift after the night shift on the previous day is not allowed.

Soft constraints

- The maximum number of shifts assigned to each nurse.
- The minimum number of shifts assigned to each nurse.
- The maximum number of consecutive working days assigned to each nurse.
- The minimum number of consecutive working days assigned to each nurse.
- The maximum number of consecutive working days assigned to each nurse on which no shift is allotted.
- The minimum number of consecutive working days assigned to each nurse on which no shift is allotted.
- The maximum number of consecutive working weekends with at least one shift assigned to each nurse.
- The minimum number of consecutive working weekends with at least one shift assigned to each nurse.
- The maximum number of weekends with at least one shift assigned to each nurse.
- Specific working day.
- Requested day off.
- Specific shift on.
- Specific shift off.
- Nurse not working on the unwanted pattern.

1.3 Objective

NSP involves the development of best shift assignments for nursing employees for a particular interval that is typically for one month. The schedule should offer a particular range of personnel for the shift forms of daily, so as to be considered as possible. Shift type is the term used for representing a time slot for which a nurse with specific skill is needed. In addition, a schedule should avoid introducing difficult to follow shift patterns and should respect as much as possible the work contract of each employee. By work contract we tend to sit down with the agreement signed between the nurse and also the hospital that has requests for days or shifts on and off, work on weekends, most acceptable consecutive work days etc. Some work contract parts will be seen as instantiations of legal necessities enforced by the state. beneath the work contracts notion it's worthy to think about the fairness of the matter that means that contract violations ought to be equally distributed to any or all offered nursing personnel. This side of the matter is typically unnoticed. Comprehensive surveys of NRP are often consulted in Cheang et al. (2003), Burke et al. (2004) and in Ernst et al. (2004a).

Approaches across numerous disciplines, and generally combining them, are projected so as to resolve the NRP. Mathematical programming was used to address simplified models of the problem. Later, constraint programming, network improvement techniques and linear and number Programming strategies were attempted. Several metaheuristics i.e. Tabu Search, simulated annealing and Evolutionary Algorithms were the strategies of selection for many researchers in their plan to address the problem. Often, these problems are hybridized with different either approximate or actual strategies. Recently revealed papers utilize techniques like branch and value, scatter search and hybrid approaches whereas the multi-objective nature of the problem is usually investigated. Finally, a variable neighbourhood search approach that compares results with those of Burke et al.,2008. As a result of the various attempts to resolve the problem variety of formulations for it do exist. So, comparisons between totally different resolution approaches area unit inherently troublesome. for instance, shift kind needs can be four expressed as one worth or as a try of values pertaining to the minimum and also the most well-liked range of nurses per shift kind. Another example of variation are often found in Burke et al. (2006) wherever no specific shift kind needs area best-known in advance and work requirements are outlined in terms of your time intervals that everyone includes a mix of potential shift sorts.

1.4 Methodology

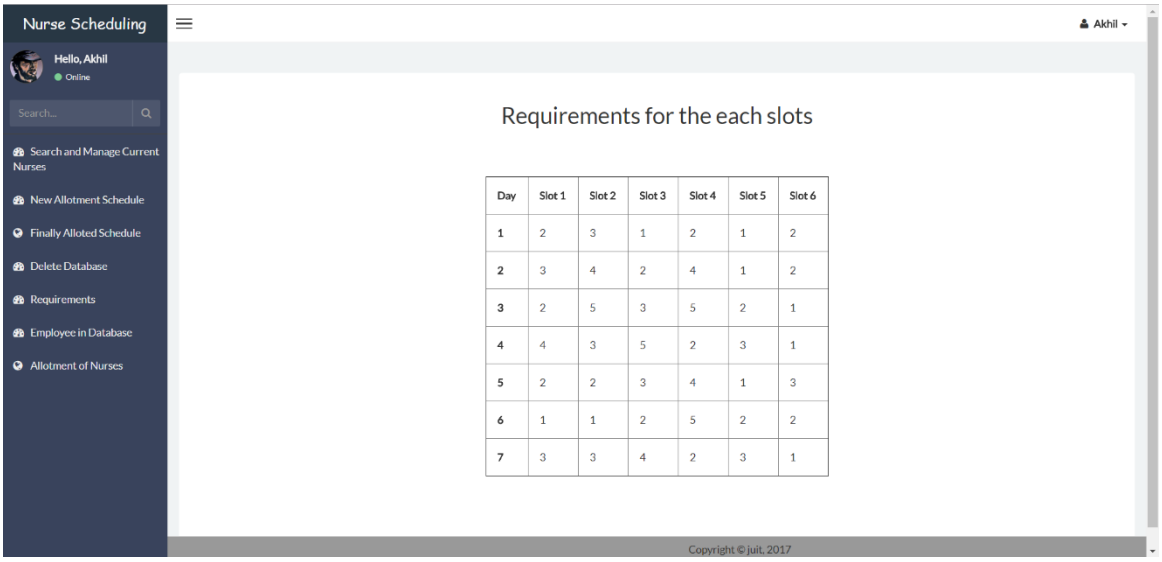
The main objective of Nurse Scheduling problem is to assign shifts to the nurses i.e schedule nurses on the basis of their availability and other constraints as described above.

For the scheduling purpose following data should be given:

- A list of nurses with their available work slots in a day, and wages
- e.g The blank cells in availability column shows that there is no particular specification for time from the employee (i.e nurse) side.
- A prescribed minimum number of staff/nurses required to be at work at a given slot (less staff are needed at night, more number of staff is needed during peak hours)

	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Isha	1	1	125	1
Edit Copy Delete	4	Shubham	1	1	130	1
Edit Copy Delete	5	Akhil	1	1	150	1
Edit Copy Delete	6	Nehan	1	1	180	1
Edit Copy Delete	7	Varchas	1	2	200	1
Edit Copy Delete	8	Akshit	1	2	200	1
Edit Copy Delete	9	Yukta	1	1	250	1
Edit Copy Delete	10	Aman	1	2	300	1
Edit Copy Delete	11	Sankalp	1	1	300	1
Edit Copy Delete	12	Mehak	1	1	320	1
Edit Copy Delete	13	Kriti	1	1	390	1
Edit Copy Delete	14	Parima	1	1	430	1
Edit Copy Delete	15	Jatin	1	1	450	1
Edit Copy Delete	16	Arpit	1	1	500	1
Edit Copy Delete	17	Nanda	1	2	190	1
Edit Copy Delete	18	Aditya	1	2	120	1

From the above data we have to find a schedule which minimizes total daily wages of the employer and meet the minimum staff requirement criteria.



1.5 Organization

Chapter 1 Explains the importance of the topic, scope, methodology of data collection and analysis.

Chapter 2 It consists of all the papers and the literature that are reviewed in detail.

Chapter 3 It covers the algorithm used for Nurse scheduling problem.

Chapter 4 Performance analysis & implementation are covered under this.

Chapter 5 gives the conclusions, recommendations, future scope & action plan.

CHAPTER 2

LITERATURE SURVEY

Literature on nurse scheduling & rostering is very huge. Several studies are employed for optimization strategies to find the solution for the NSP, like constraint, goal programming, mixed integer or integer programming or linear programming. Most of the current paper handle the Nurse scheduling problem with strategies like simulation, genetic algorithms or tab search. Alternative approaches, like meta- heuristics or heuristics are less accessible, and will be time-taking. Hence our contribution, associated with existing approaches, is concentrated on the applied mathematics drawback, that seeks to satisfy the demand coverage whereas minimizing the regular payment value and increasing the nurses' preferences moreover as team balance. Totally different objectives are studied during this literature to decrease manual programming, to extend demand covering in terms of work force size however additionally per needed skills, to get equity between the schedules.

2.1 Hybridization of harmony search with hill climbing for highly constrained nurse rostering problem

In HHSA, to empower the exploitation capability, the hill climbing optimizer is used to hybridize the harmony search algorithm. Moreover, random selection scheme is able to replace the global best concept of particle swarm optimization to change and modify the memory consideration operator and the rate of convergence.

In HHSA, the quality dataset revealed within the 1st international nurse rostering competition 2010 (INRC2010) was utilised to judge it. The experimental results additionally reveal that the hybrid approach is ready to get smart ends up in terms of answer quality and time necessities. the long run work is to check the effectiveness of various combination of neighbourhood structures within the hill rise optimizer for NRP, to adapt the projected HHSA for the IInd international nurse scheduling competition 2015, wherever the dataset is on the market at INRC-II web site

2.2 Directed Bee Colony Optimization Algorithm to Solve the Nurse Rostering Problem

Nelder-Mead methodology is used in this paper to propose a metaheuristic technique which is known as Directed Bee Colony optimisation rule for finding the NSP. To resolve the

NSP, the authors used a multi-objective mathematical programming model and planned a strategy for the variation of a Multi objective Directed Bee Colony optimisation (MODBCO). For optimizing the scheduling issues and for finding the multi objective drawback, it is successfully used.

This MODBCO is associate degree integration of settled native search, multiagent particle system surroundings, and honey bee decision-making method. The performance of the rule is assessed mistreatment the quality dataset INRC2010.

2.3 Variable neighbourhood search accelerated column generation for the nurse rostering problem

In order to obtain feasible solutions, Variable Neighbourhood Search accelerate Column Generation procedure for the NRP in addition to a Relax-and-fix Heuristic. This technique can handle a large integer programming formulation for the Nurse Rostering Problem. The algorithm that is proposed is not only able to provide strong lower bounds, but was also proved to be the best-known solutions for all 29 hidden instances with four week scheduling horizons. For the final computance, duality gap was less than 8% for all the instances. The main component of this method is heuristic based on Variable Neighbourhood Search Procedure which significantly speeds up the convergence of the column generation procedure.

2.4 A tensor based hyper-heuristic for nurse rostering

While solving the computationally hard problems, Hyper-heuristics have come out to be as a general search methodology that manage and mix a predefined set of low level heuristics. The approach that was proposed finally evaluated on a well-known nurse scheduling benchmark composed of a diverse collection of instances obtained from different hospitals across the world.

It either generates new low-level heuristics from existing components or selects from a set of available low-level heuristics to solve a problem, leading to a distinction between generation and selection hyper-heuristics, respectively.

The constraints in the nurse scheduling problem can be grouped into two categories:

(i) Two or more than two nurses are linked

(ii) Only applicable to a single nurse.

Constraints that fall into the 1st category comprises of the cover (sometimes known as demand) constraints. The second group of constraints model the requirements on each nurse's individual schedule. For example, the minimum and maximum number of hours worked, permissible shifts.

2.5 A bi-objective integrated approach of building surgical teams and nurse schedule rosters to minimize nurse's idle time & maximize surgical team affinities and.

In this paper, we have reviewed that the impact of using scrub and circulating nurses or using polyvalent nurses. In all instances and settings, the empathy between the surgical team members were more sensitive to differences in the idle time. Furthermore, the use of polyvalent nurse's results in rostering with reduction in idle time and can have better surgical team member affinities.

A bi-objective integer programming (BOIP) problem is achieved by this weekly assignment of the nurses to surgical operations. The nurse allocations are based on the weekly assumptions that several nurses are available, and that overtime may be required and on elective surgical schedule.

2.6 A constraint programming and hybrid integer approach to solve nurse scheduling problems

This paper presents a hybrid algorithm, which combines Integer Programming and Constraint Programming to fully solve the highly-constrained Nurse Scheduling Problem. For finding an optimal solution and obtaining lower-bounds the power of Constraint Programming is used in a co-operative manner and for this the strength of Integer Programming is exploited. In order to improve the performance and for obtaining high-quality solutions as well as strong lower-bounds for a relatively short time from the algorithm, many innovative ways were applied to extract useful information. For example, the computational difficulty of constraints and instances to adaptively set the search parameters.

This algorithm utilizes the strengths of Constraint Programming to aid the IP solver to achieve better solutions. Due to the high-level hybridization of Constraint and Integer Programming, an significant aspect of the proposed hybrid algorithm is its straightforward adaptability to practical circumstances in terms of implementation, that makes it more advantageous when compared to a sophisticated customized methodology from the literature that are difficult to implement.

2.7 A two-phase adaptive variable neighbourhood approach for nurse

For solving the highly constraint nurse scheduling problem, a two-phase variable neighbourhood search algorithm was used. To show its efficiency, this algorithm is applied to all the instances proposed in the 1st International Nurse Scheduling Competition (INRC-2010). Finally, the results of the proposed algorithm have been compared with five INRC-2010 competition finalists on the 60 competition instances. As Compared to the winner of INRC-2010 competition who achieves the best-known result in 36 out of 60 instances, the proposed algorithm achieves the best-known result in 50 out of 60 instances. In future, the main issue for proposed algorithm will be to verify it against the newly published nurse rostering instances.

2.8 Adaptive neighbourhood search for nurse scheduling

ANS uses the two distinct neighbourhood moves simultaneously and adapts switches in the company of three diversification and intensification search strategies according to the search history. Computational results assessed on the three sets of sixty competition instances that shows that ANS has improved the best-known results for twelve instances while matching the best bounds for thirty-nine other instances.

In this paper consists of allocating shifts to nurses in accordance with a given set of constraints. Usually, two types of constraints are defined: those which must be strictly satisfied under any circumstances (hard constraints) and the constraints which are not necessarily satisfied but whose contravention should be desirably minimized (soft constraints). A schedule that can satisfy all the hard constraints is known as feasible assignment.

2.9 A Hybrid Evolutionary Approach to the Nurse Scheduling Problem

In this paper, we marked the real nurse scheduling problem faced by a large U.K. hospital. The problem is represented here for completeness. The formulation employed in those two

studies that represents a generic nurse scheduling problem and has been used in many other studies. Weekly schedules are made in this problem for about thirty nurses. Each days' schedule comprises of a day shift and a night shift, and for each shift a practicable solution has to allocate sufficient nurses to cover the actual demands that are subject to change throughout the week.

2.10 Constraint Programming for Nurse Scheduling

It has only recently become available, being situated at the intersection of artificial intelligence and operational research. It combines logic programming and operational research techniques.

Constraint Programming has allowed successful solution of some complex, combined problems such as, scheduling, planning, resources allocation, car sequencing problem, etc. More formally, constraint programming aims at solving a class of problems called the constraint satisfaction problem (CSP). In the case of a CSP dealing with finite domains, several techniques can be combined to find a solution to the problem. The techniques generally used are: generate and test, standard backtracking, forward checking, and looking ahead.

2.11 A Hybrid Artificial Bee Colony for a Nurse Scheduling Problem

Artificial bee colony algorithm proposed, is recently emerged as one of most popular nature-inspired algorithms which has been utilized for tackling practical optimization problems [24, 25]. This algorithm carries out a search by evolving a population of food sources (i.e. solutions) with the aid of nondeterministic operators (e.g., employed, onlooker and scout), by iteratively improving the solutions using these operators until some criterion of convergence has been achieved.

2.12 A systematic two-phase approach for the nurse scheduling problem

This algorithm uses two phases for producing feasible and high-quality schedules. The day assignment problem is solved at the 1st phase. This problem comprises of defining rest and work days of all nurses for the specified planning horizon. On completion of this phase each nurse has a day only schedule that is to be followed by having no shift specific information.

CHAPTER 3

SYSTEM DEVELOPMENT

It consists of Algorithm used to solve Nurse Scheduling problem. It is implemented in PHP using MYSQL.

Problem Statement

Obviously the "Nurse Scheduling Problem" is not only limited to "nurses" as an occupation, so we will just use the generic term "employee" here. The customer's problem statement was as follows.

Given:

- A list of employees having available work hours, and hourly salaries
- A prescribed minimum number of staff that are required to be at work at a given hour (fewer staff are needed at night, more staff are needed during peak hours)

Find a plan which MINIMIZES:

- The wages the employer must payout daily.

1. Read in the requirements and employee data from the Excel sheet

The staff information and scheduling requirements are in an CSV file, so we will first need to import the data in the MSQL Database.

1. **Form Schedule on the website**

- Administrator must fill the dates in the admin panel. These are dates for which form will be active on the website.
- During this duration every employee must fill his/her preference according to the availability.
- Nurses must also fill their wages.

2. **Generate and prepare the Excel Sheet**

Excel sheet of the requirement of the nurses in four particular slots is prepared by the admin.

Now this excel sheet is imported to MYSQL server where database of employee with their availability and wages is already present.

3. Algorithm for scheduling employee

Generating a formulation of a problem involves expressing the minimization objective and constraints.

In this Algorithm, one slot is taken at once. One employee is scheduled & then moved to the second slot. And the process continues further.

For the allocation in the 1st slot, first of all number of requirements is checked from the requirement table. Then it goes to employee table to fetch the availability of the employee. In the employee table, all the employees are sorted in ascending order according to their wages and then numbering is given to them according to their respective rank. Now their can be many constraints in assigning the employee in the slots. They are as follows: -

- Minimum and maximum employee required by the hospital can vary.
- Employee can change their availability.
- They can also change their wages after seven days.
- Minimum requirement for each employee that he/she must work in at least one slot.
- Employee if available can be assigned multiple shifts on the same day.

4. Gather and display the results

All the results are gathered and displayed in a human-friendly form. Employee can check their schedule by respective login on the website. They can also send their request to admin to change their shifts.

5. Conclusion

The problem outlined is not a hard one to solve. To be sure, it takes experience & a bit of study to be able to know how to easily convert a real-world problem into its equivalent algorithm.

But for problems like this, especially at larger problem sizes, attempting to solve it using other methods such trial-and-error or genetic algorithms will usually take longer, and generally not yield as good results.

CHAPTER 4

PERFORMANCE ANALYSIS

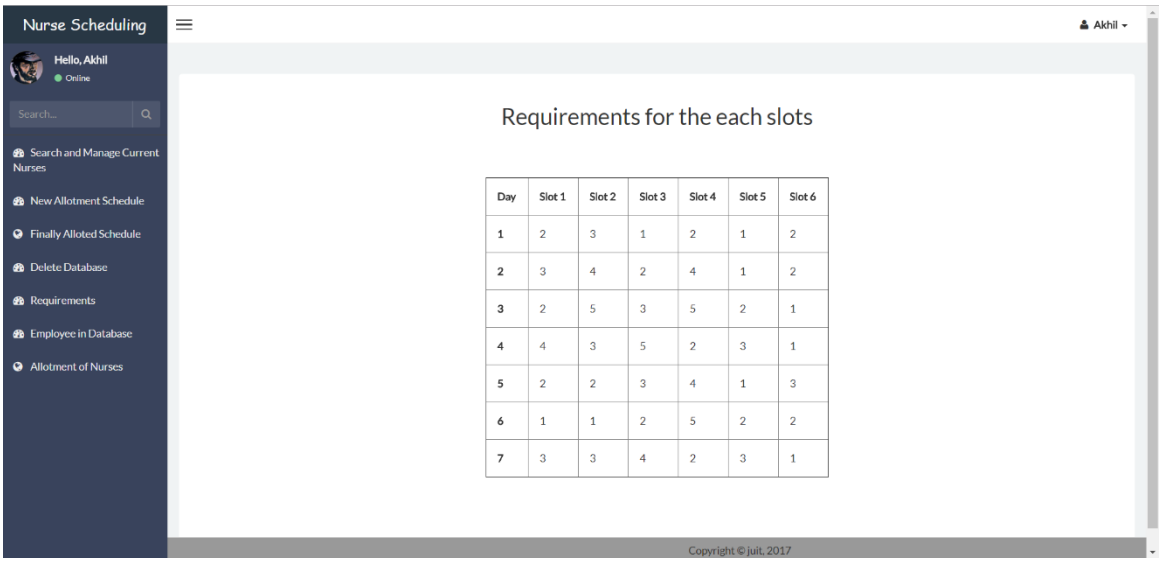
For the sake of analysing the performance of our algorithm we have considered different datasets and presented its output as follows: -

Dataset 1

Input:

- A list of employees with their available work hours, and hourly salaries

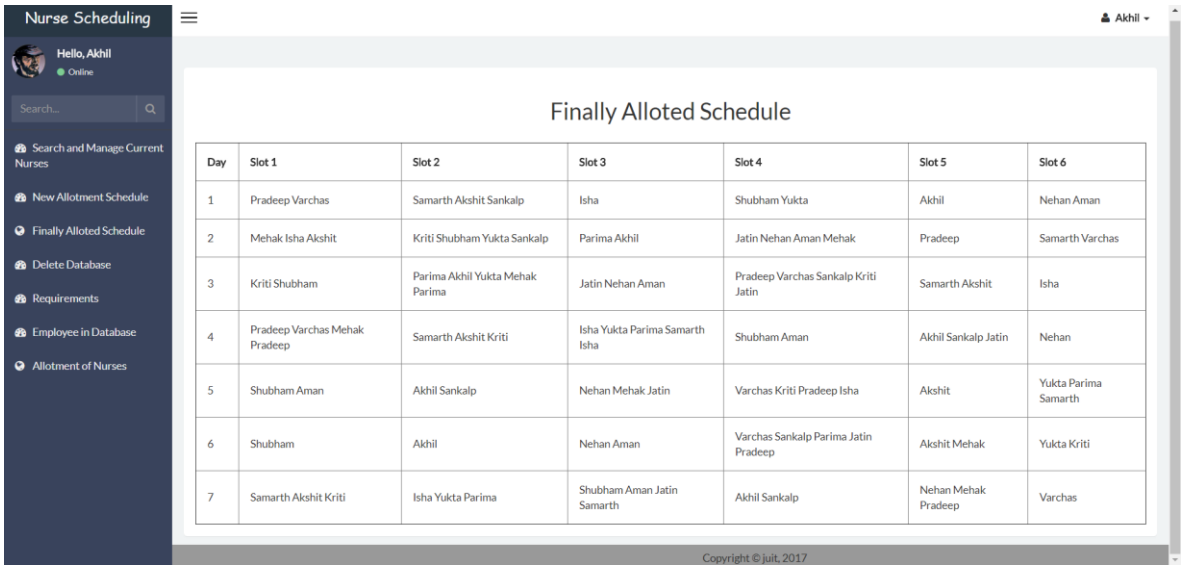
	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Isha	1	1	125	1
Edit Copy Delete	4	Shubham	1	1	130	1
Edit Copy Delete	5	Akhil	1	1	150	1
Edit Copy Delete	6	Nehan	1	1	180	1
Edit Copy Delete	7	Varchas	1	2	200	1
Edit Copy Delete	8	Akshit	1	2	200	1
Edit Copy Delete	9	Yukta	1	1	250	1
Edit Copy Delete	10	Aman	1	2	300	1
Edit Copy Delete	11	Sankalp	1	1	300	1
Edit Copy Delete	12	Mehak	1	1	320	1
Edit Copy Delete	13	Kriti	1	1	390	1
Edit Copy Delete	14	Parima	1	1	430	1
Edit Copy Delete	15	Jatin	1	1	450	1
Edit Copy Delete	16	Arpit	1	1	500	1
Edit Copy Delete	17	Nanda	1	2	190	1
Edit Copy Delete	18	Aditya	1	2	120	1



Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	2	3	1	2	1	2
2	3	4	2	4	1	2
3	2	5	3	5	2	1
4	4	3	5	2	3	1
5	2	2	3	4	1	3
6	1	1	2	5	2	2
7	3	3	4	2	3	1

Output:

- When No Optimization is done.
- When there is no constraint on every nurse must work atleast once in a day if there is a requirement.



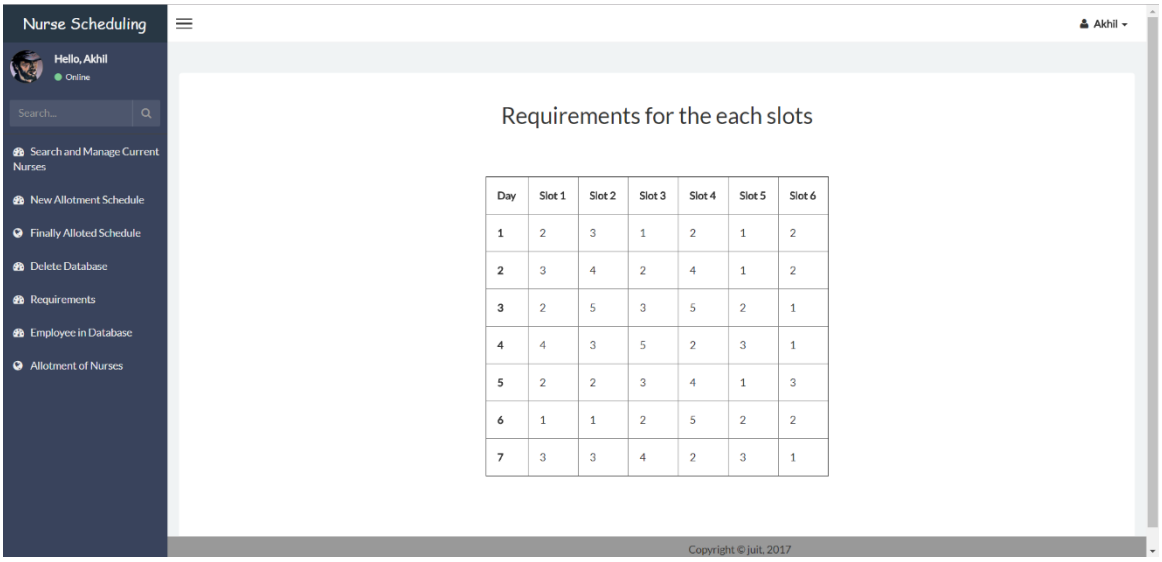
Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	Pradeep Varchas	Samarth Akshit Sankalp	Isha	Shubham Yukta	Akhil	Nehan Aman
2	Mehak Isha Akshit	Kriti Shubham Yukta Sankalp	Parima Akhil	Jatin Nehan Aman Mehak	Pradeep	Samarth Varchas
3	Kriti Shubham	Parima Akhil Yukta Mehak Parima	Jatin Nehan Aman	Pradeep Varchas Sankalp Kriti Jatin	Samarth Akshit	Isha
4	Pradeep Varchas Mehak Pradeep	Samarth Akshit Kriti	Isha Yukta Parima Samarth Isha	Shubham Aman	Akhil Sankalp Jatin	Nehan
5	Shubham Aman	Akhil Sankalp	Nehan Mehak Jatin	Varchas Kriti Pradeep Isha	Akshit	Yukta Parima Samarth
6	Shubham	Akhil	Nehan Aman	Varchas Sankalp Parima Jatin Pradeep	Akshit Mehak	Yukta Kriti
7	Samarth Akshit Kriti	Isha Yukta Parima	Shubham Aman Jatin Samarth	Akhil Sankalp	Nehan Mehak Pradeep	Varchas

Dataset 2

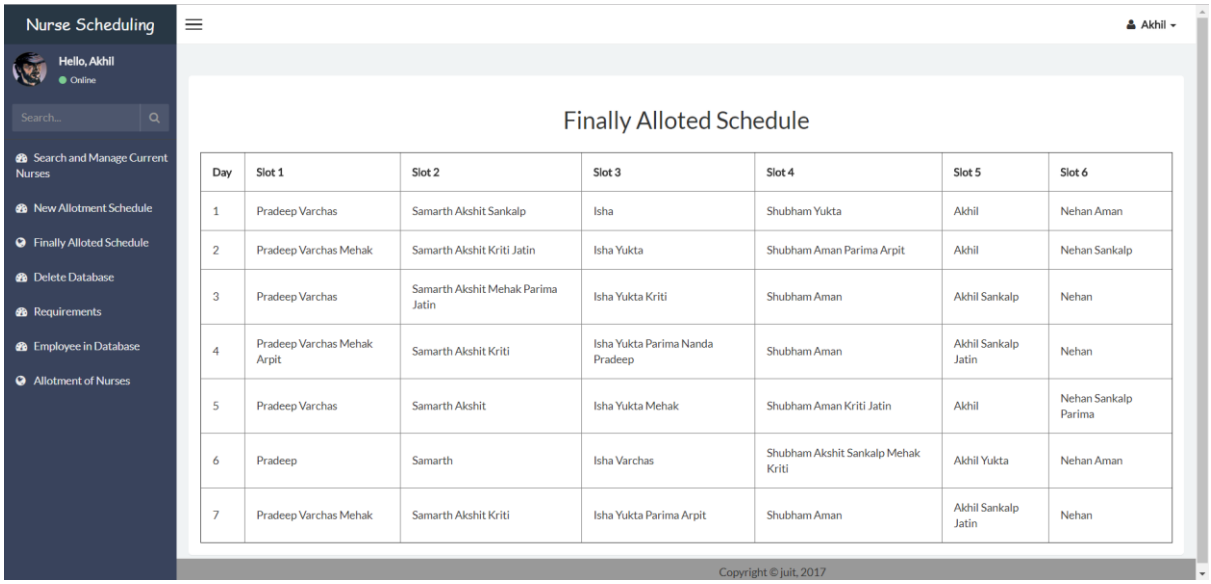
Now Optimization is done by the help of algorithm and staff requirements are safe.

Input:

	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Isha	1	1	125	1
Edit Copy Delete	4	Shubham	1	1	130	1
Edit Copy Delete	5	Akhil	1	1	150	1
Edit Copy Delete	6	Nehan	1	1	180	1
Edit Copy Delete	7	Varchas	1	2	200	1
Edit Copy Delete	8	Akshit	1	2	200	1
Edit Copy Delete	9	Yukta	1	1	250	1
Edit Copy Delete	10	Aman	1	2	300	1
Edit Copy Delete	11	Sankalp	1	1	300	1
Edit Copy Delete	12	Mehak	1	1	320	1
Edit Copy Delete	13	Kriti	1	1	390	1
Edit Copy Delete	14	Parima	1	1	430	1
Edit Copy Delete	15	Jatin	1	1	450	1
Edit Copy Delete	16	Arpit	1	1	500	1
Edit Copy Delete	17	Nanda	1	2	190	1
Edit Copy Delete	18	Aditya	1	2	120	1



Output: When there is no constraint and Optimisation is done.

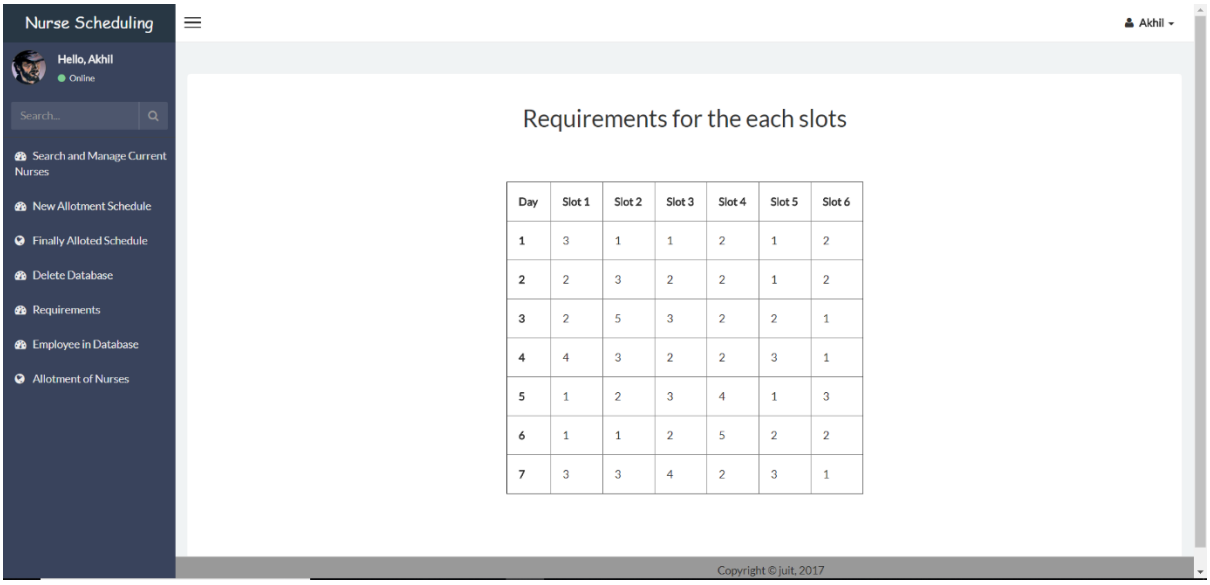


Dataset 3

Now we are keeping the first input table same as the previous dataset and just varying the other table i.e. minimum staff requirement.

Input

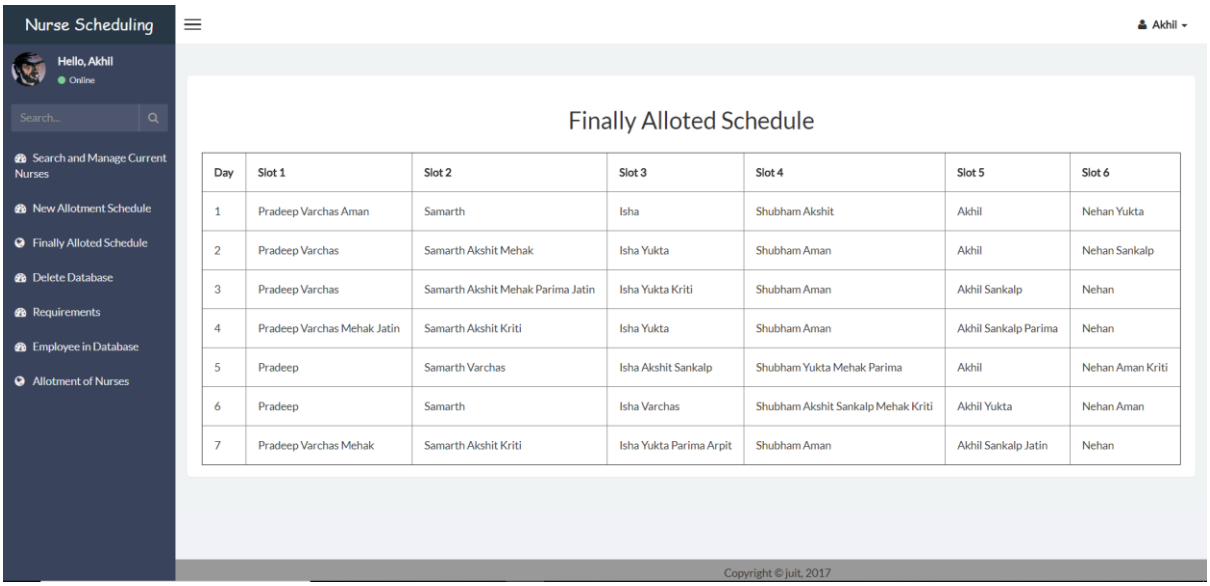
	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Isha	1	1	125	1
Edit Copy Delete	4	Shubham	1	1	130	1
Edit Copy Delete	5	Akhil	1	1	150	1
Edit Copy Delete	6	Nehan	1	1	180	1
Edit Copy Delete	7	Varchas	1	2	200	1
Edit Copy Delete	8	Akshit	1	2	200	1
Edit Copy Delete	9	Yukta	1	1	250	1
Edit Copy Delete	10	Aman	1	2	300	1
Edit Copy Delete	11	Sankalp	1	1	300	1
Edit Copy Delete	12	Mehak	1	1	320	1
Edit Copy Delete	13	Kriti	1	1	390	1
Edit Copy Delete	14	Parima	1	1	430	1
Edit Copy Delete	15	Jatin	1	1	450	1
Edit Copy Delete	16	Arpit	1	1	500	1
Edit Copy Delete	17	Nanda	1	2	190	1
Edit Copy Delete	18	Aditya	1	2	120	1



Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	3	1	1	2	1	2
2	2	3	2	2	1	2
3	2	5	3	2	2	1
4	4	3	2	2	3	1
5	1	2	3	4	1	3
6	1	1	2	5	2	2
7	3	3	4	2	3	1

Output: -

When there is no constraint on every nurse must work atleast once in a day



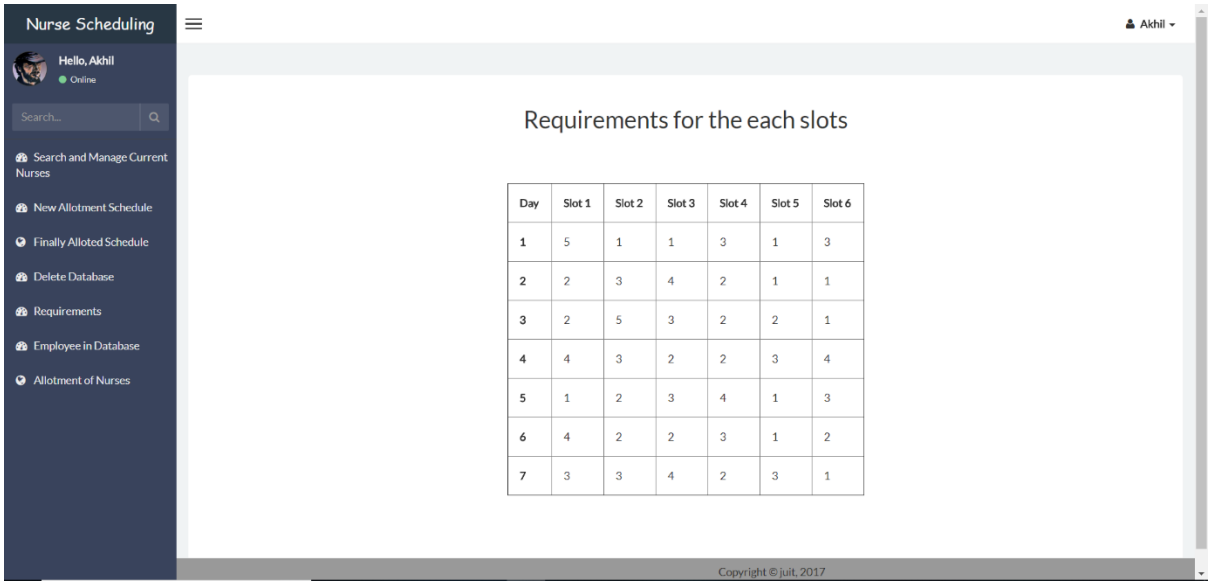
Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	Pradeep Varchas Aman	Samarth	Isha	Shubham Akshit	Akhil	Nehan Yukta
2	Pradeep Varchas	Samarth Akshit Mehak	Isha Yukta	Shubham Aman	Akhil	Nehan Sankalp
3	Pradeep Varchas	Samarth Akshit Mehak Parima Jatin	Isha Yukta Kriti	Shubham Aman	Akhil Sankalp	Nehan
4	Pradeep Varchas Mehak Jatin	Samarth Akshit Kriti	Isha Yukta	Shubham Aman	Akhil Sankalp Parima	Nehan
5	Pradeep	Samarth Varchas	Isha Akshit Sankalp	Shubham Yukta Mehak Parima	Akhil	Nehan Aman Kriti
6	Pradeep	Samarth	Isha Varchas	Shubham Akshit Sankalp Mehak Kriti	Akhil Yukta	Nehan Aman
7	Pradeep Varchas Mehak	Samarth Akshit Kriti	Isha Yukta Parima Arpit	Shubham Aman	Akhil Sankalp Jatin	Nehan

Dataset 4

Now we keeping the first input table same as the previous dataset and just varying the other table i.e minimum staff requirement.

Input

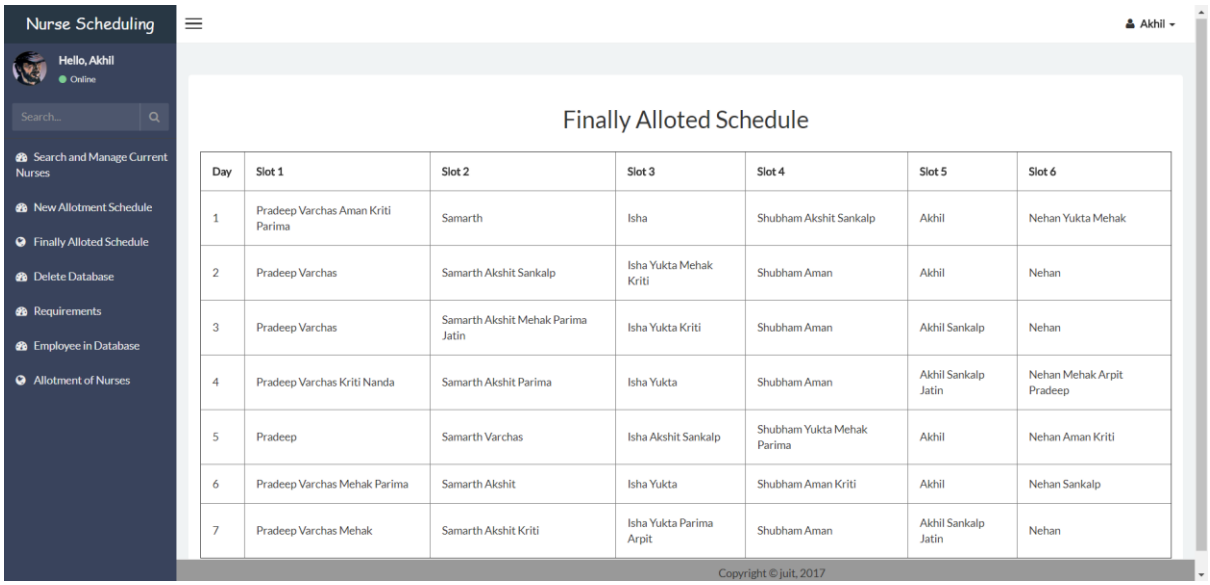
	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Isha	1	1	125	1
Edit Copy Delete	4	Shubham	1	1	130	1
Edit Copy Delete	5	Akhil	1	1	150	1
Edit Copy Delete	6	Nehan	1	1	180	1
Edit Copy Delete	7	Varchas	1	2	200	1
Edit Copy Delete	8	Akshit	1	2	200	1
Edit Copy Delete	9	Yukta	1	1	250	1
Edit Copy Delete	10	Aman	1	2	300	1
Edit Copy Delete	11	Sankalp	1	1	300	1
Edit Copy Delete	12	Mehak	1	1	320	1
Edit Copy Delete	13	Kriti	1	1	390	1
Edit Copy Delete	14	Parima	1	1	430	1
Edit Copy Delete	15	Jatin	1	1	450	1
Edit Copy Delete	16	Arpit	1	1	500	1
Edit Copy Delete	17	Nanda	1	2	190	1
Edit Copy Delete	18	Aditya	1	2	120	1



Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	5	1	1	3	1	3
2	2	3	4	2	1	1
3	2	5	3	2	2	1
4	4	3	2	2	3	4
5	1	2	3	4	1	3
6	4	2	2	3	1	2
7	3	3	4	2	3	1

Output: -

When there is no constraint on every nurse must work atleast once in a day



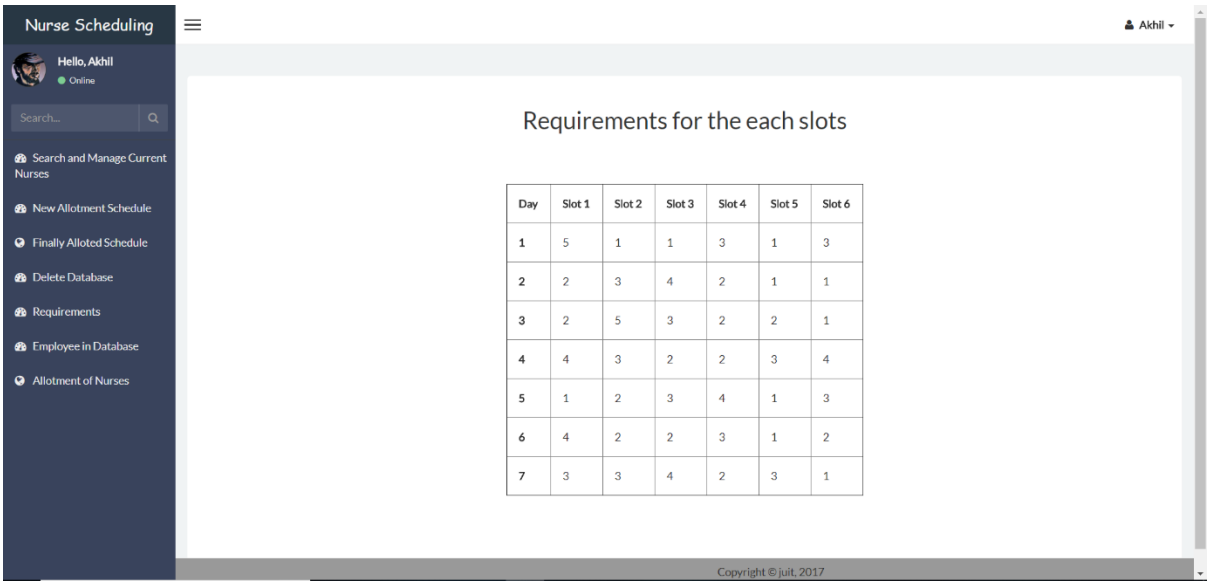
Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	Pradeep Varchas Aman Kriti Parima	Samarth	Isha	Shubham Akshit Sankalp	Akhil	Nehan Yukta Mehak
2	Pradeep Varchas	Samarth Akshit Sankalp	Isha Yukta Mehak Kriti	Shubham Aman	Akhil	Nehan
3	Pradeep Varchas	Samarth Akshit Mehak Parima Jatin	Isha Yukta Kriti	Shubham Aman	Akhil Sankalp	Nehan
4	Pradeep Varchas Kriti Nanda	Samarth Akshit Parima	Isha Yukta	Shubham Aman	Akhil Sankalp Jatin	Nehan Mehak Arpit Pradeep
5	Pradeep	Samarth Varchas	Isha Akshit Sankalp	Shubham Yukta Mehak Parima	Akhil	Nehan Aman Kriti
6	Pradeep Varchas Mehak Parima	Samarth Akshit	Isha Yukta	Shubham Aman Kriti	Akhil	Nehan Sankalp
7	Pradeep Varchas Mehak	Samarth Akshit Kriti	Isha Yukta Parima Arpit	Shubham Aman	Akhil Sankalp Jatin	Nehan

Dataset 5

Now we are keeping minimum staff requirement same & making sure that one nurse is not available.

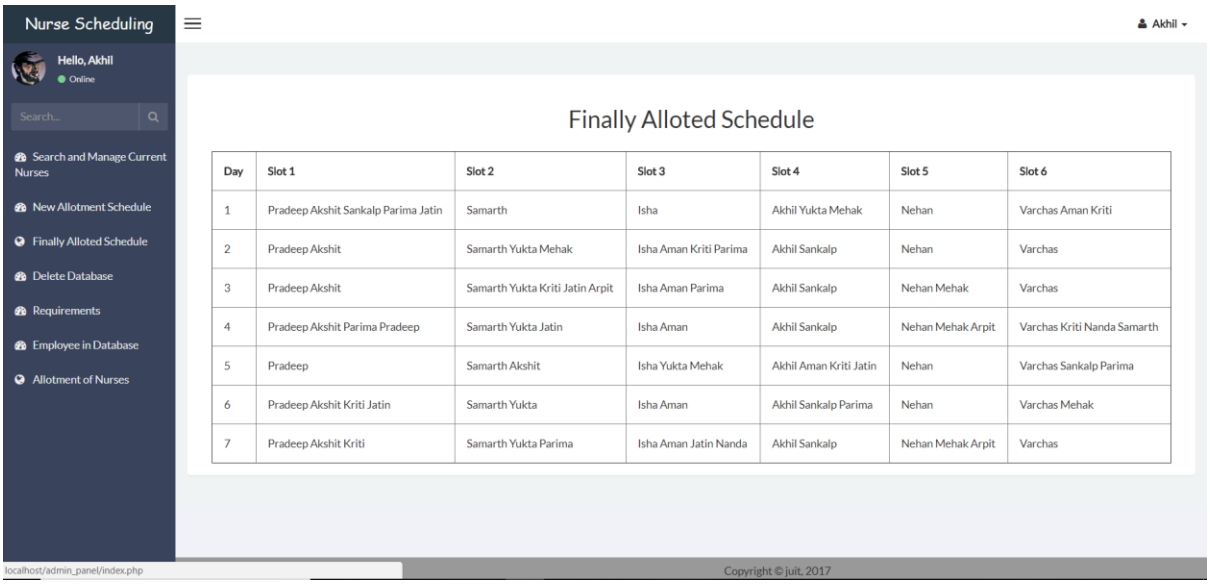
Input

	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Isha	1	1	125	1
Edit Copy Delete	4	Shubham	1	1	130	0
Edit Copy Delete	5	Akhil	1	1	150	1
Edit Copy Delete	6	Nehan	1	1	180	1
Edit Copy Delete	7	Varchas	1	2	200	1
Edit Copy Delete	8	Akshit	1	2	200	1
Edit Copy Delete	9	Yukta	1	1	250	1
Edit Copy Delete	10	Aman	1	2	300	1
Edit Copy Delete	11	Sankalp	1	1	300	1
Edit Copy Delete	12	Mehak	1	1	320	1
Edit Copy Delete	13	Kriti	1	1	390	1
Edit Copy Delete	14	Parima	1	1	430	1
Edit Copy Delete	15	Jatin	1	1	450	1
Edit Copy Delete	16	Arpit	1	1	500	1
Edit Copy Delete	17	Nanda	1	2	190	1
Edit Copy Delete	18	Aditya	1	2	120	1



Output

It shows that nurse will be eliminated from the output.

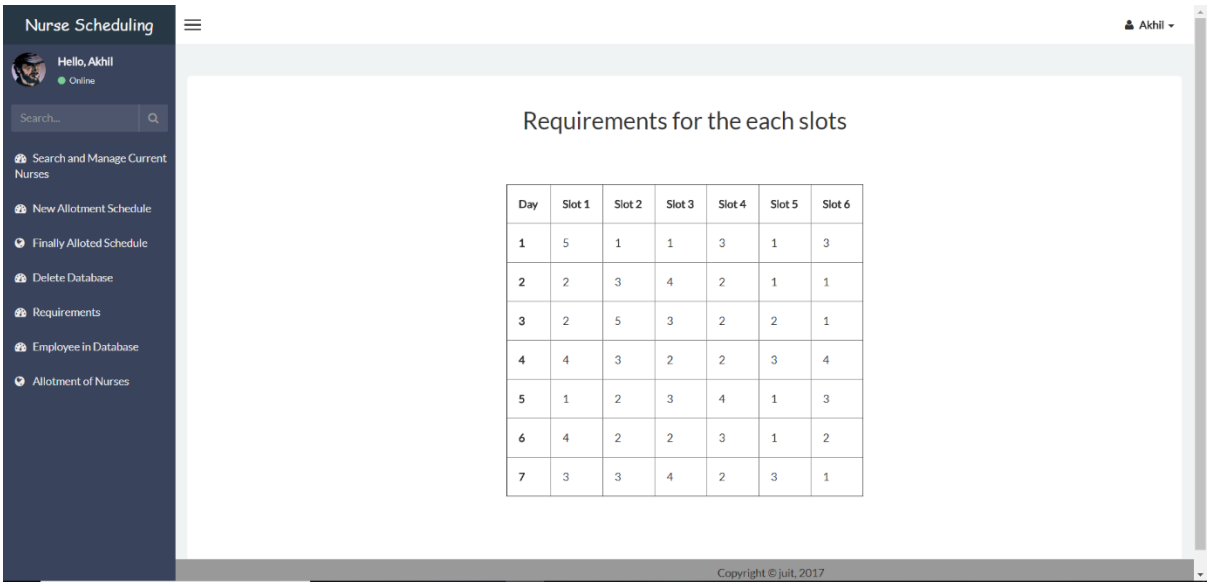


Dataset 6

Now we are keeping minimum staff requirement same & making sure that many nurses are not available.

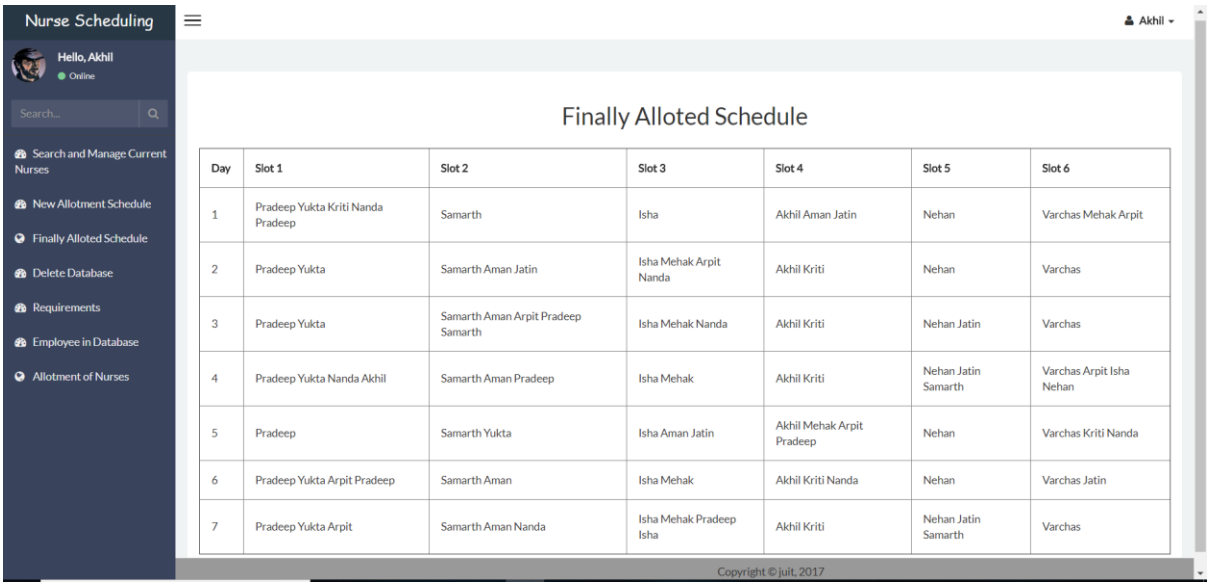
Input: -

	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Isha	1	1	125	1
Edit Copy Delete	4	Shubham	0	1	130	0
Edit Copy Delete	5	Akhil	1	1	150	1
Edit Copy Delete	6	Nehan	1	1	180	1
Edit Copy Delete	7	Varchas	1	2	200	1
Edit Copy Delete	8	Akshit	1	2	200	0
Edit Copy Delete	9	Yukta	1	1	250	1
Edit Copy Delete	10	Aman	1	2	300	1
Edit Copy Delete	11	Sankalp	1	1	300	0
Edit Copy Delete	12	Mehak	1	1	320	1
Edit Copy Delete	13	Kriti	1	1	390	1
Edit Copy Delete	14	Parima	1	1	430	0
Edit Copy Delete	15	Jatin	1	1	450	1
Edit Copy Delete	16	Arpit	1	1	500	1
Edit Copy Delete	17	Nanda	1	2	190	1
Edit Copy Delete	18	Aditya	1	2	120	1



Output

In this, if the nurses are less available then multiple duties are allotted to the nurses according to the availability.

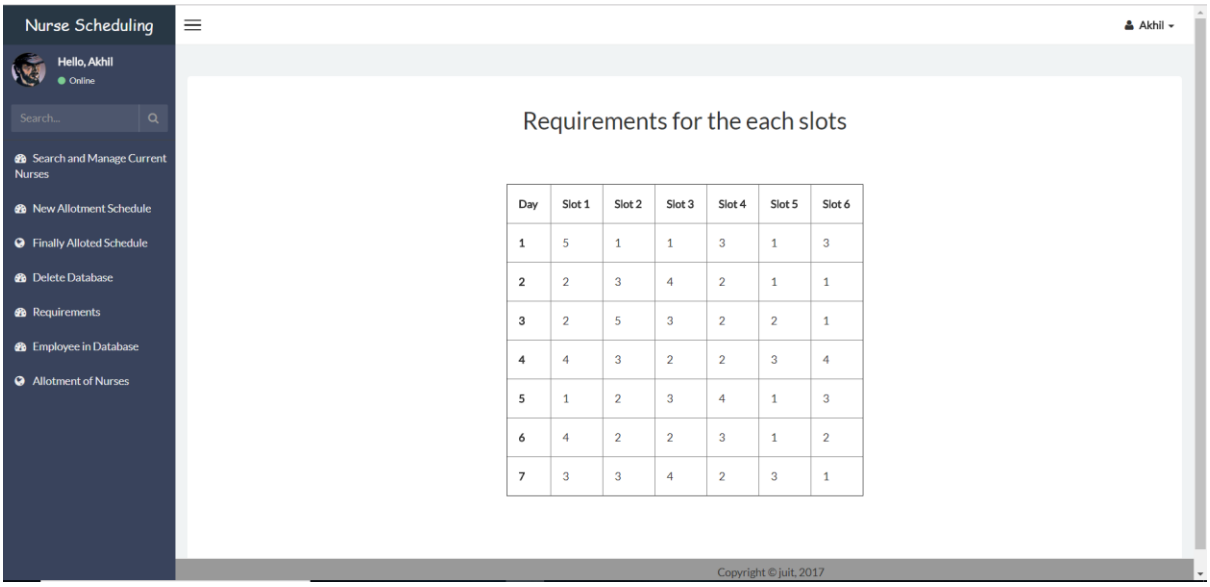


Dataset 7

Now we are increasing the number of available staff & taking the requirements as the previous dataset.

Input

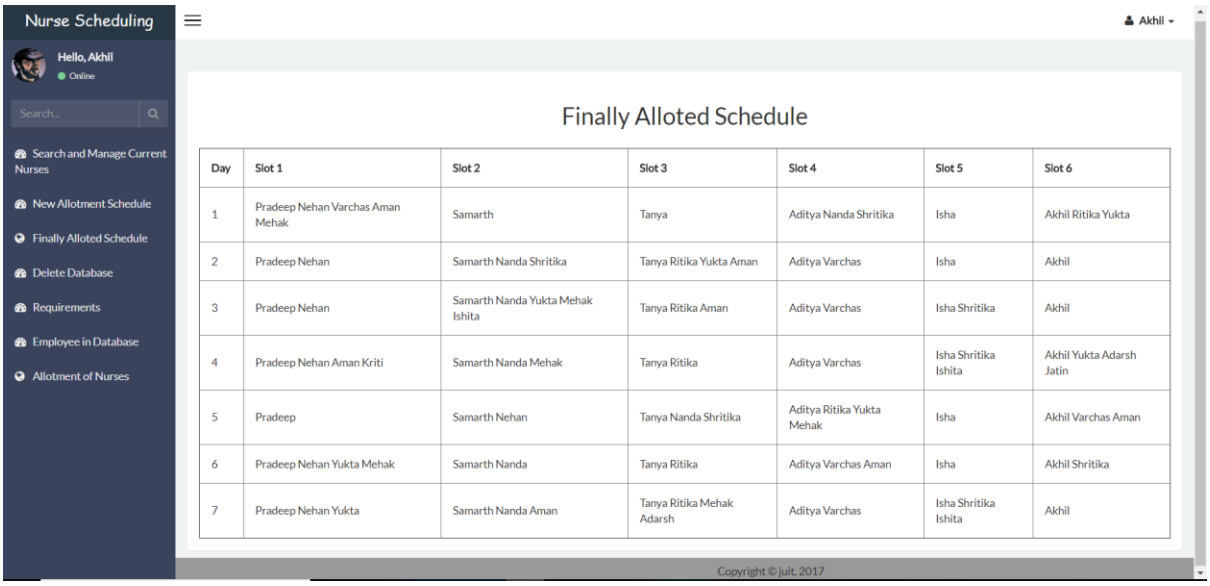
	num	name	mini	maxi	wage ▲ 1	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Tanya	1	3	110	1
Edit Copy Delete	4	Aditya	1	2	120	0
Edit Copy Delete	5	Isha	1	1	125	1
Edit Copy Delete	6	Shubham	0	1	130	0
Edit Copy Delete	7	Akhil	1	1	150	1
Edit Copy Delete	8	Nehan	1	1	180	1
Edit Copy Delete	9	Nanda	1	2	190	1
Edit Copy Delete	10	Ritika	1	3	190	1
Edit Copy Delete	12	Akshit	0	2	200	0
Edit Copy Delete	11	Varchas	1	2	200	1
Edit Copy Delete	13	Shritika	1	2	235	1
Edit Copy Delete	14	Yukta	1	1	250	1
Edit Copy Delete	15	Aman	1	2	300	1
Edit Copy Delete	16	Sankalp	0	1	300	0
Edit Copy Delete	17	Mehak	1	1	320	1
Edit Copy Delete	18	Ishita	1	2	330	1
Edit Copy Delete	19	Adarsh	1	2	370	1
Edit Copy Delete	20	Kriti	1	1	390	1
Edit Copy Delete	21	Parima	0	1	430	0
Edit Copy Delete	22	Jatin	1	1	450	1
Edit Copy Delete	23	Arpit	1	1	500	1



Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	5	1	1	3	1	3
2	2	3	4	2	1	1
3	2	5	3	2	2	1
4	4	3	2	2	3	4
5	1	2	3	4	1	3
6	4	2	2	3	1	2
7	3	3	4	2	3	1

Output

Results are calculated after increasing the number of employees & by keeping previous staff requirements.



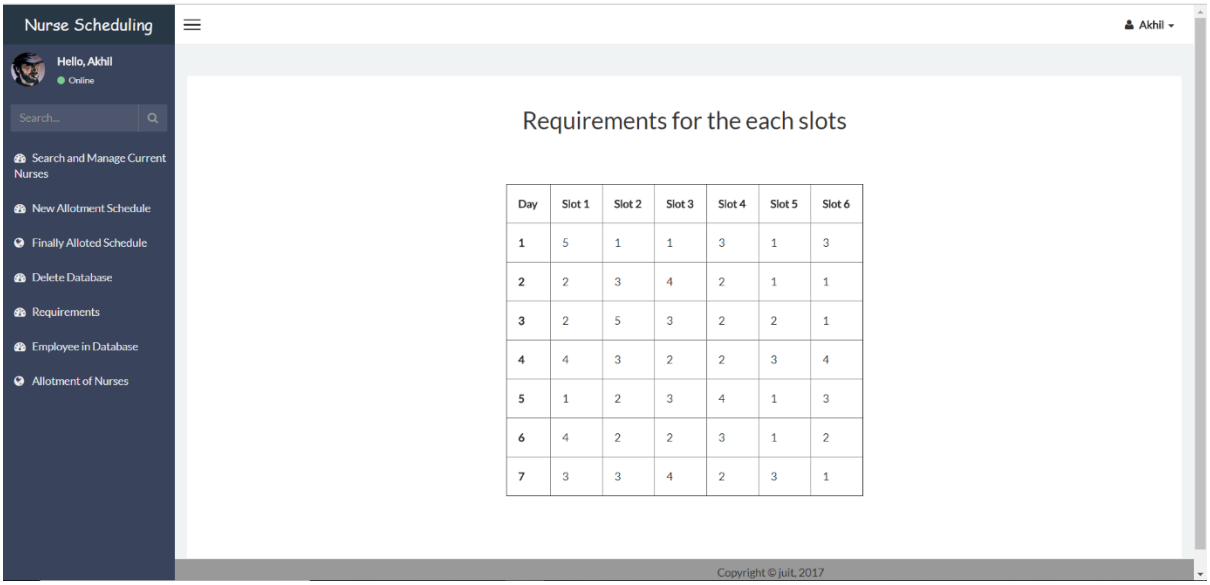
Day	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
1	Pradeep Nehan Varchas Aman Mehak	Samarth	Tanya	Aditya Nanda Shritika	Isha	Akhil Ritika Yukta
2	Pradeep Nehan	Samarth Nanda Shritika	Tanya Ritika Yukta Aman	Aditya Varchas	Isha	Akhil
3	Pradeep Nehan	Samarth Nanda Yukta Mehak Ishita	Tanya Ritika Aman	Aditya Varchas	Isha Shritika	Akhil
4	Pradeep Nehan Aman Kriti	Samarth Nanda Mehak	Tanya Ritika	Aditya Varchas	Isha Shritika Ishita	Akhil Yukta Adarsh Jatin
5	Pradeep	Samarth Nehan	Tanya Nanda Shritika	Aditya Ritika Yukta Mehak	Isha	Akhil Varchas Aman
6	Pradeep Nehan Yukta Mehak	Samarth Nanda	Tanya Ritika	Aditya Varchas Aman	Isha	Akhil Shritika
7	Pradeep Nehan Yukta	Samarth Nanda Aman	Tanya Ritika Mehak Adarsh	Aditya Varchas	Isha Shritika Ishita	Akhil

Dataset 8

In this dataset we are keeping the minimum staff requirement table same as previous one and varying the first table's attributes like hourly wage, availability etc.

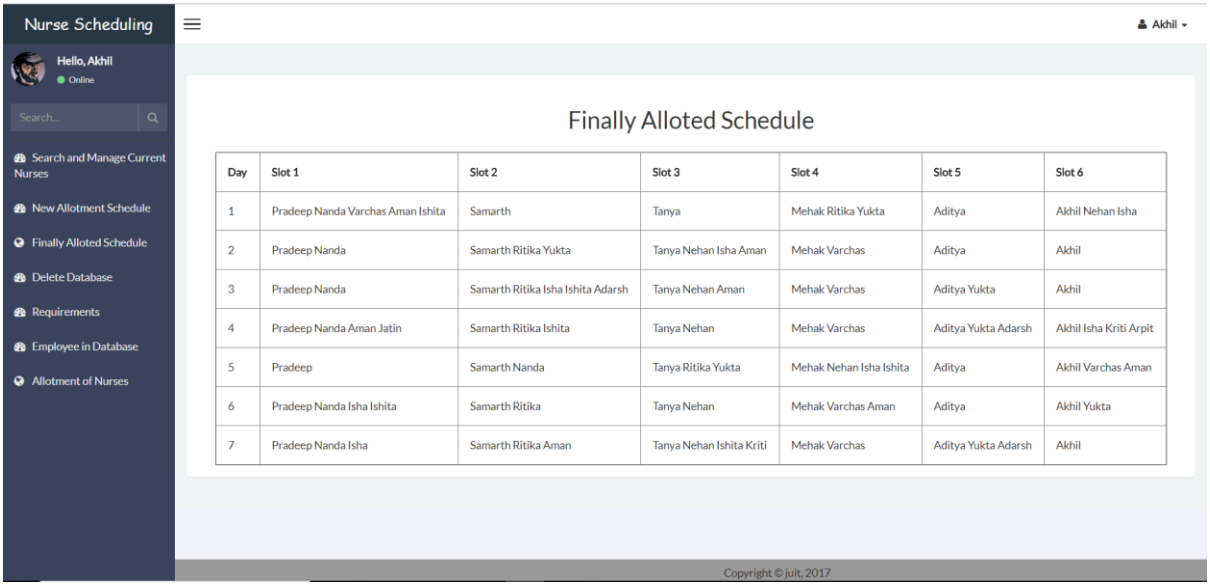
	num	name	mini	maxi	wage	allot
Edit Copy Delete	1	Pradeep	1	1	100	1
Edit Copy Delete	2	Samarth	1	1	100	1
Edit Copy Delete	3	Tanya	1	3	110	0
Edit Copy Delete	4	Mehak	1	1	115	1
Edit Copy Delete	5	Aditya	1	2	120	1
Edit Copy Delete	6	Shubham	0	1	130	0
Edit Copy Delete	7	Akhil	1	1	150	1
Edit Copy Delete	8	Nanda	1	2	190	0
Edit Copy Delete	9	Ritika	1	3	190	1
Edit Copy Delete	10	Nehan	1	1	192	1
Edit Copy Delete	11	Varchas	1	2	200	1
Edit Copy Delete	12	Akshit	0	2	200	0
Edit Copy Delete	13	Yukta	1	1	250	1
Edit Copy Delete	14	Isha	1	1	300	1
Edit Copy Delete	15	Aman	1	2	300	1
Edit Copy Delete	16	Sankalp	0	1	300	0
Edit Copy Delete	17	Ishita	1	2	330	1
Edit Copy Delete	18	Adarsh	1	2	370	0
Edit Copy Delete	19	Kriti	1	1	390	1
Edit Copy Delete	20	Parima	0	1	430	1
Edit Copy Delete	21	Jatin	1	1	450	0
Edit Copy Delete	22	Arpit	1	1	500	1
Edit Copy Delete	23	Shritika	1	2	510	1

You can also edit most values by double-clicking directly on them.

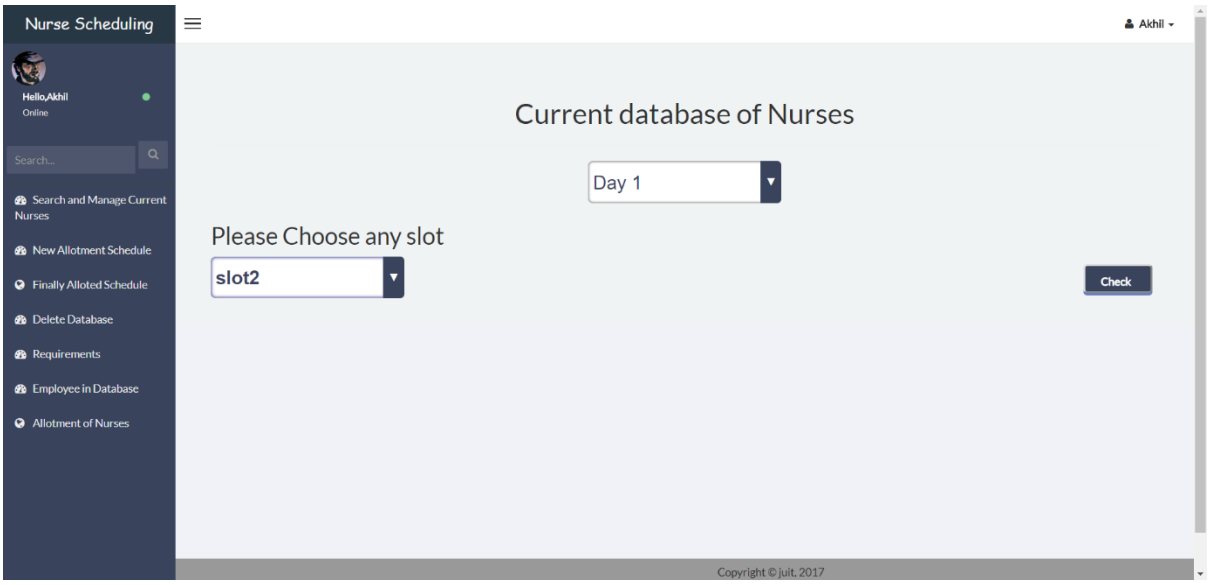


Output

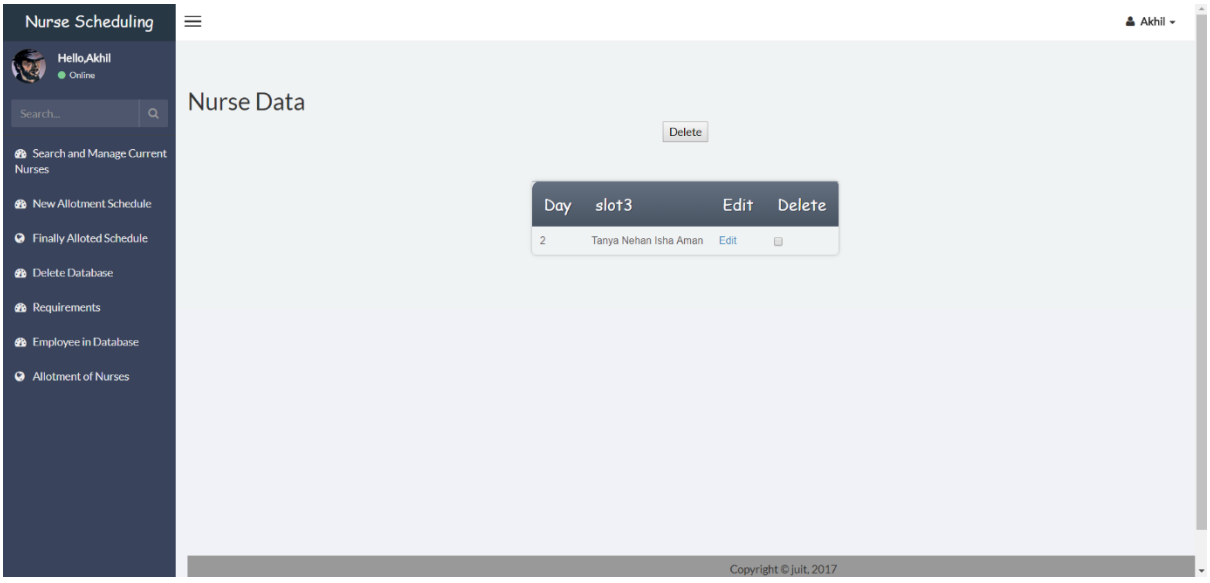
Results are shown after changing the wages, availability of the employee



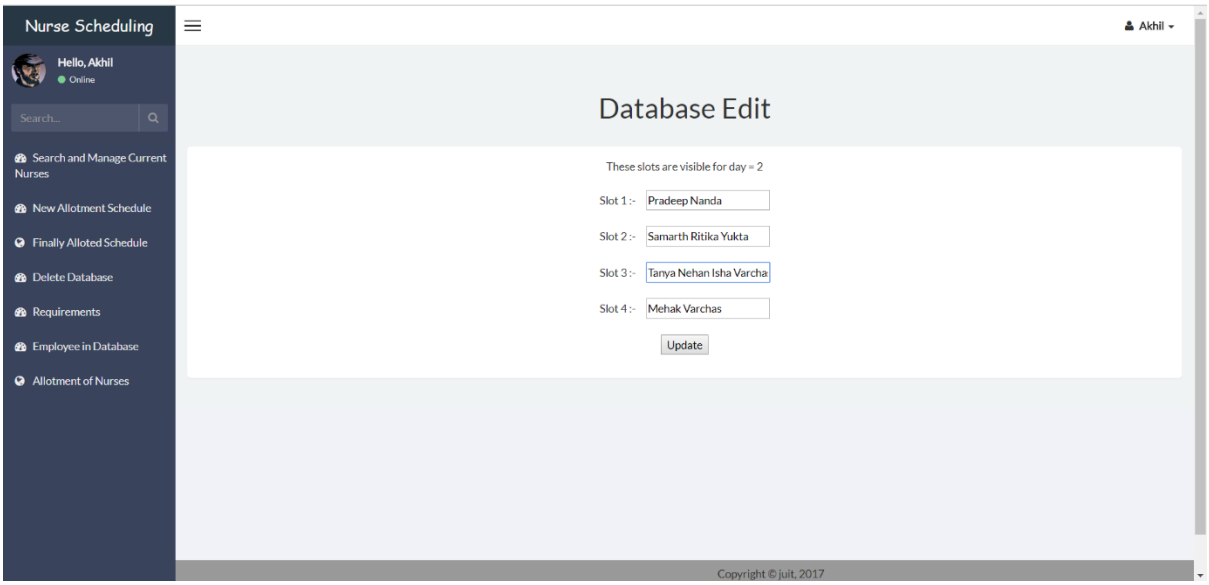
Search and Manage the scheduled employee



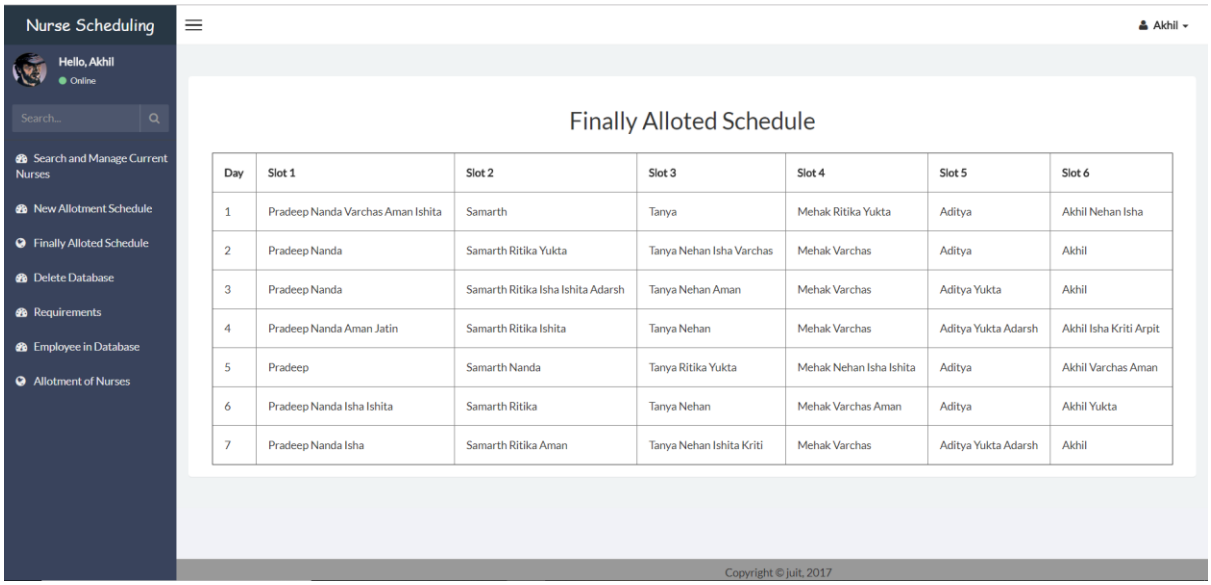
Edit Page



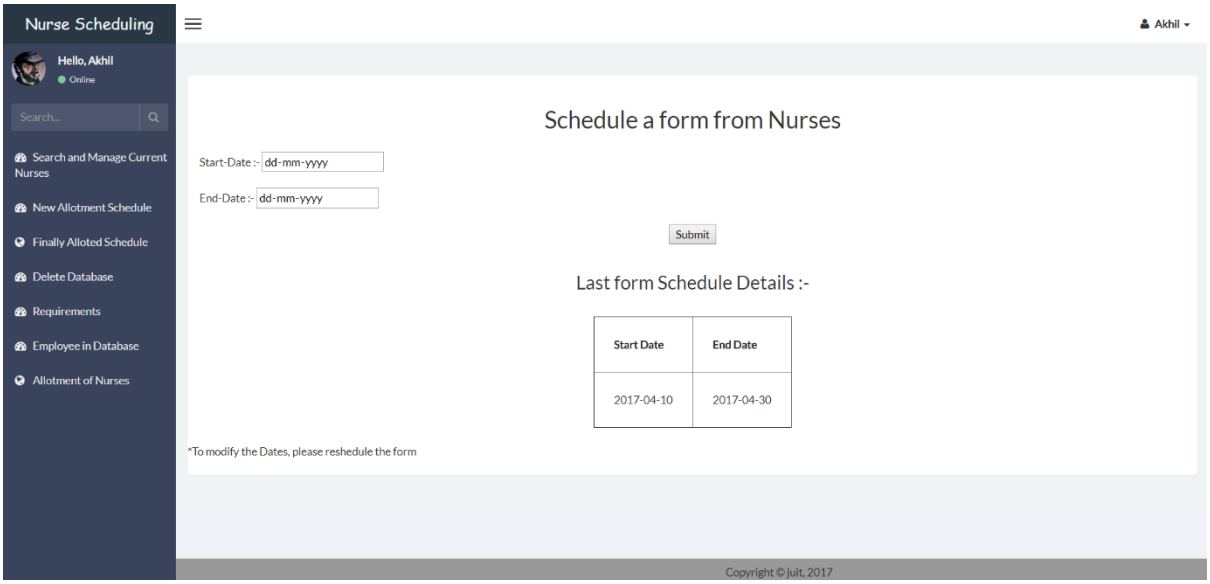
Database Edit Page



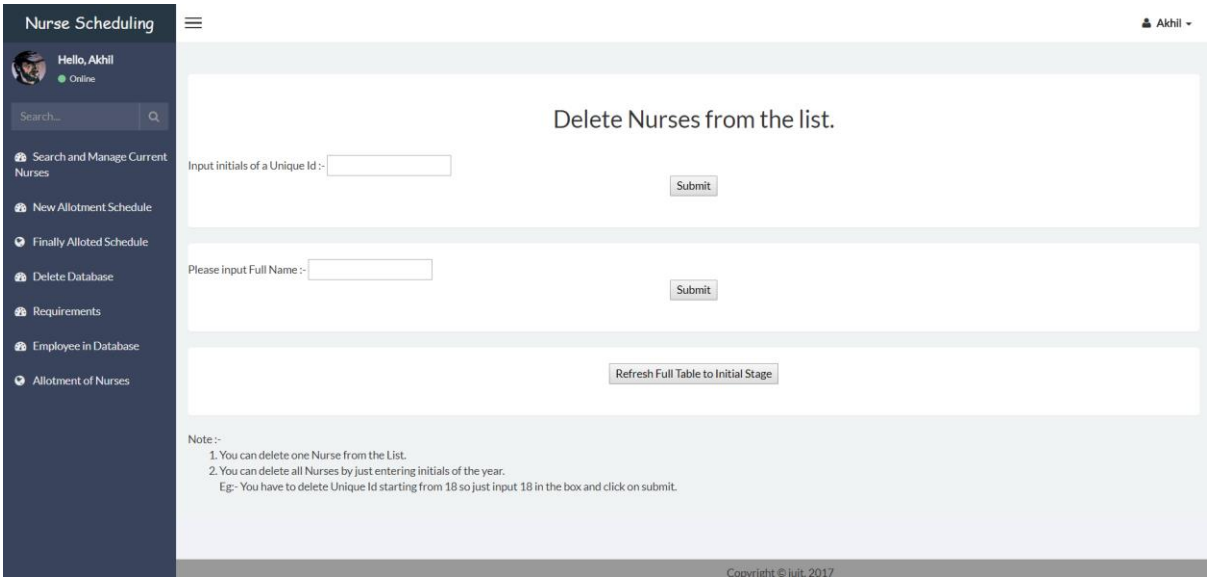
Changes reflected in the Database



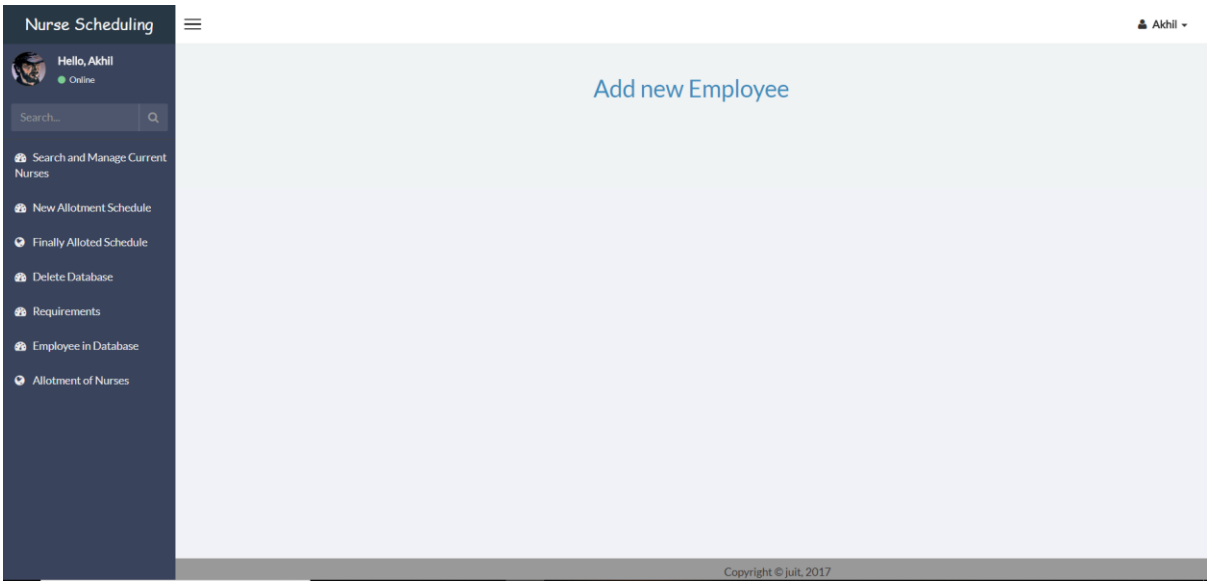
Schedule During which employee can fill their preferences.



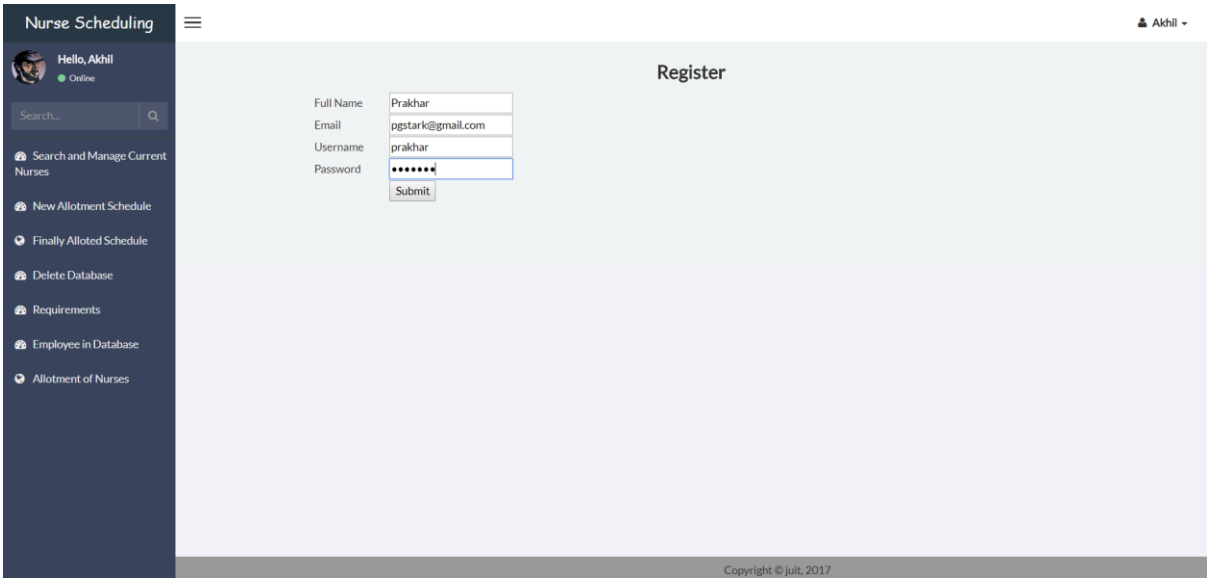
Refreshing the full Database to the Initial Stage



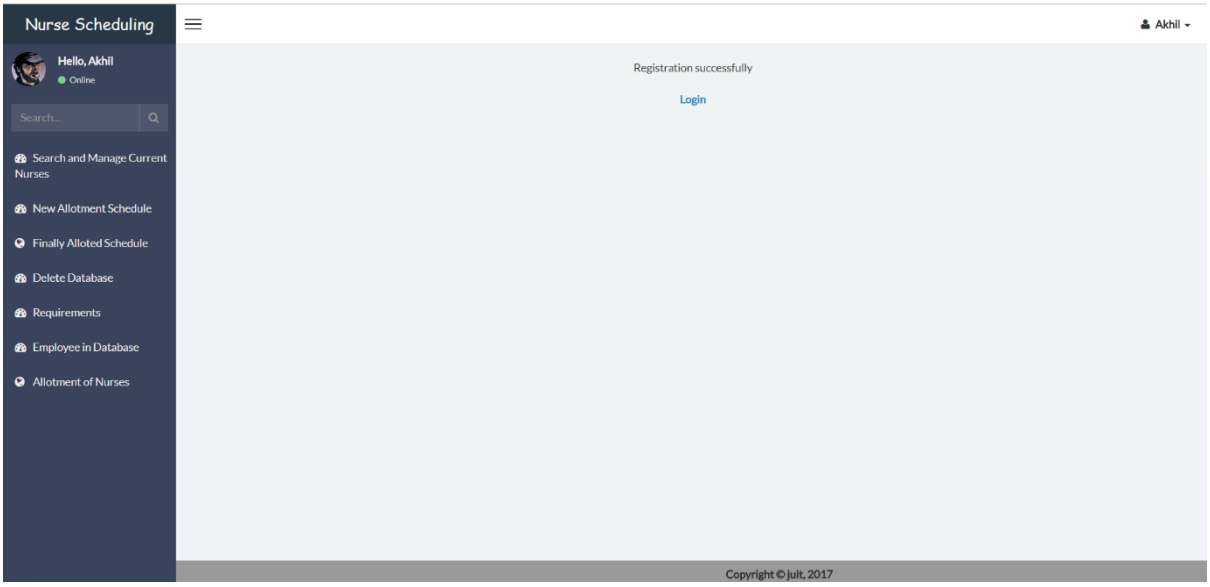
Home Page to register New Nurses



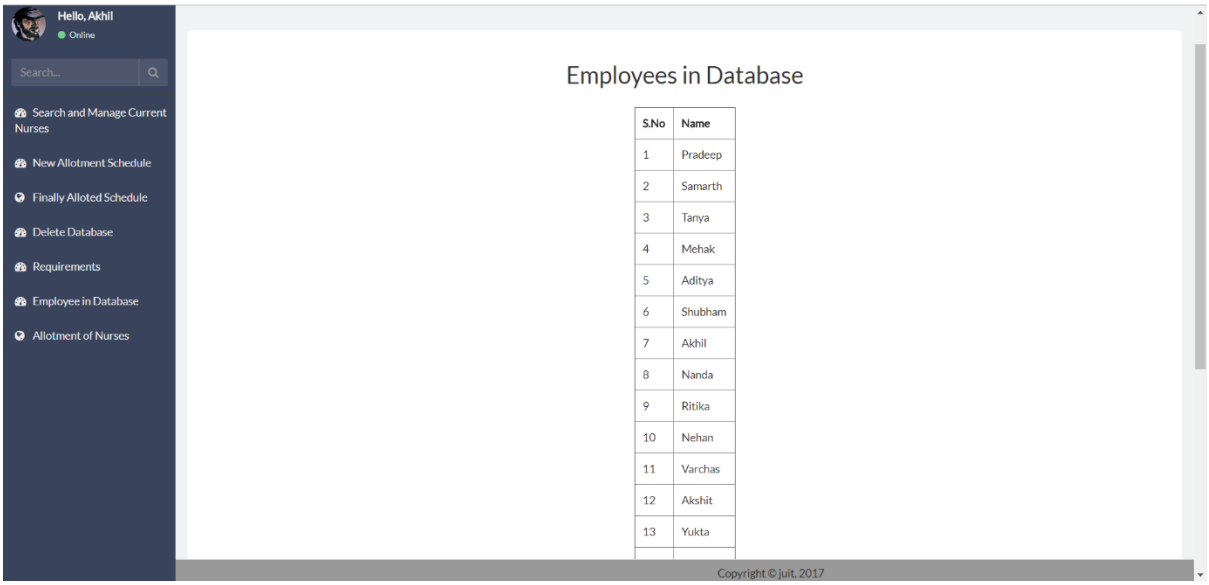
Credentials filled by Admin



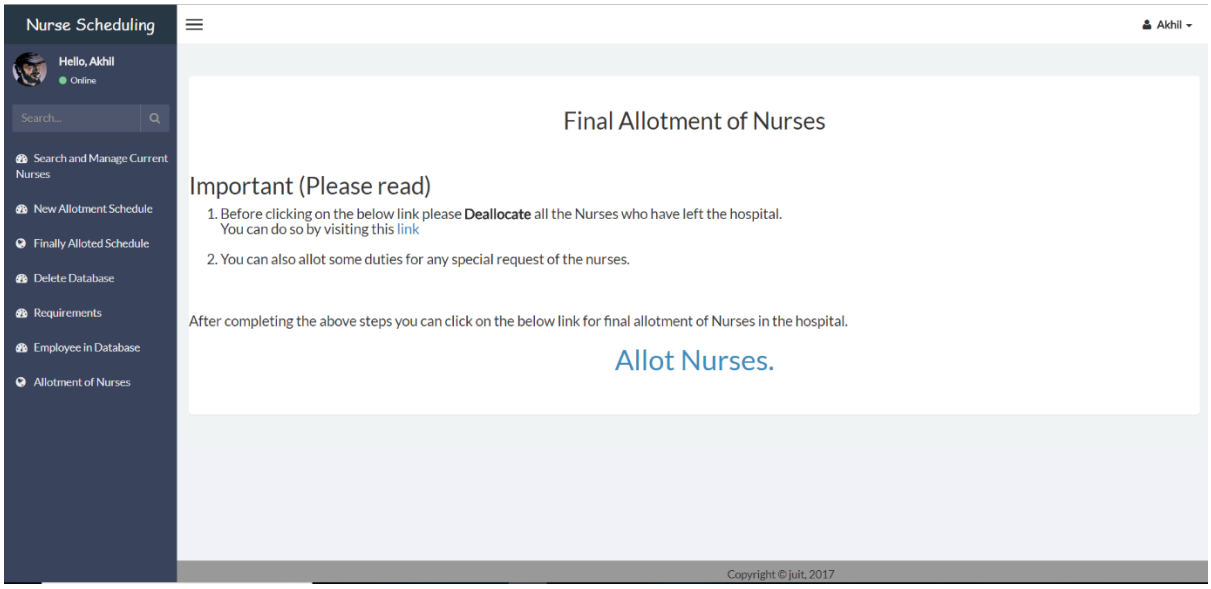
Employee Successfully registered by the Admin



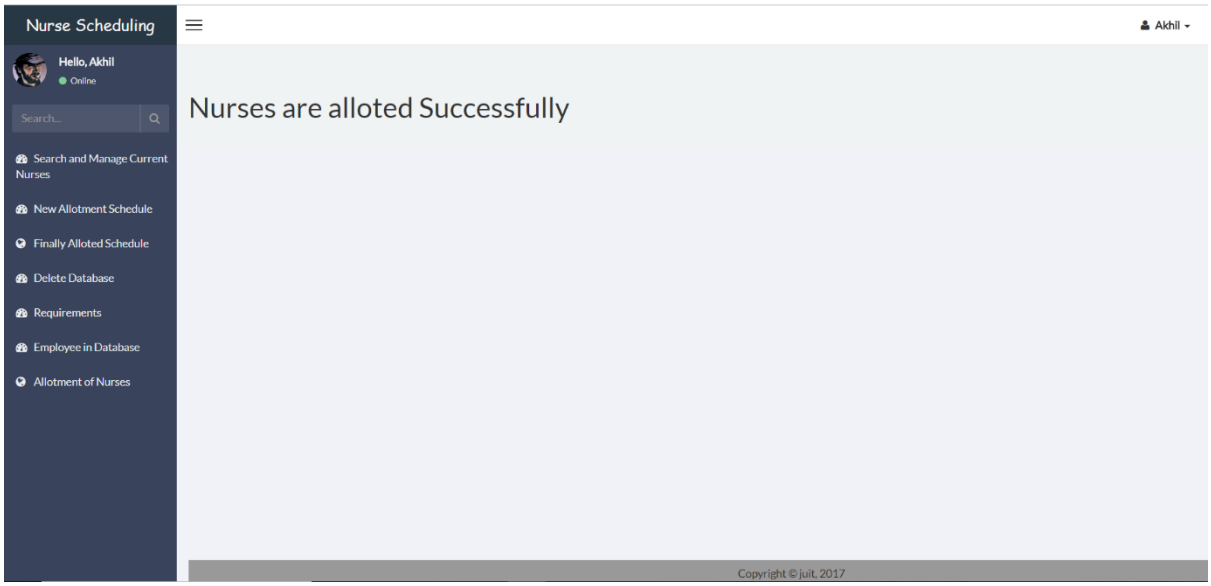
Employee who have registered themselves



Final Allotment Instructions of the Employee.



After Allotment Page



CHAPTER 5

CONCLUSIONS

5.1 Conclusion

It is an overview of the planning and nurse scheduling problem, this seeks the minimum number of nurses that can handle the hospital requirements. Although we have shown the constraints of the system in terms of shift and piece of works. The main aim of this problem is to maximize the decency of the schedule. All over nurse scheduling is a complex scheduling problem that influences hospital personnel on a daily basis. In general, it is efficiently utilizing the time and effort, to balance the workload to lead more contented and effective.

This project has produced a system that can overcome problems that have been identified in nurse schedule arrangement. In the same time it can give the detail study of genetic algorithm method in solving scheduling problem. The system can benefit the patient also where nurse can do their job professionally and give their patient the best efforts if they have a good working time.

5.2 Future Scope

As it can be observed, a large amount of work has already been finished in the area of personnel scheduling and rostering. For improvements in this area, then there is still a significant room available. We see improvements occurring not only in the area of scheduling algorithms, but also in the much more applicability of scheduling because of the flexibility that more sophisticated scheduling software tools will be able to provide in the future. Here we take out a few key areas that, in our opinion, are most likely to occupy people working in this area over the next decade. The first area of merging applies mainly to applications where scheduling is already well established such as the sectors like transportation in aviation and general industry in particular. Here the main challenge is to obtain greater and greater efficiency gains not by improving algorithms for solving any one aspect of the problem but by integrating more of the steps into a single problem. For example, it is more likely that staffing costs for any airline could be reduced if the scheduling is considered (at a fairly high level) during the scheduling and timetabling of aircraft.

Another important area that requires further work is generalization of more models and methods. Currently, algorithms and models often require significant modification when they are to be transferred to a different application area, or to accommodate changes within an organization. In a continually changing environment it is not desirable to have an organization internal structures, processes and work policies hard-wired into models, algorithms and software for personnel scheduling.

5.3 Application Area

It is widely used in Health care systems. The first rostering focus in health care systems is in nurse scheduling which is usually in acute care hospital. There are both cost and clinical imperatives linked with providing appropriate levels of staff in various medical wards in a hospital. The schedulers must provide suitably qualified nurses to cover the demand originating from the numbers of patients in the wards while observing various work regulations, distinguishing between casual and permanent staff, ensuring that weekend and night shifts are distributed fairly, allowing for days and leave off, and accommodating a range of employee preferences. The resulting scheduling problems are, in most cases, over-constrained. Some studies marked the problem of determining staff skills and levels based on the numbers of patients and their corresponding medical needs. Others adopted branch and bound techniques, mathematical programming, or goal programming, approaches in that objective which contains shift satisfaction terms, weighted coverages and the constraints enforce difficult rules like the ratios of nurse grades that must be observed on their respective shifts. Others used many other iterative algorithms to generate cyclic schedulers in which fairness can be obtained by having each nurse work the same sequence of shifts with individual shift order offset so as to provide the essential coverage and skill mix within wards.

References

1. M. Rajeswari,J. Amudhavel,Sujatha Pothula and P. Dhavachelvan,"Directed Bee Colony Optimization Algorithm to Solve the Nurse Rostering Problem",pp. 23-27,2017.
2. Al-Betar M, Bolaji A, Khader A, Awadallah M,Alkoffash M,"Hybridization of harmony search with hill climbing for highly constrained nurse rostering problem",pp.463-483,2015
3. Rafael A.M. Gomesa, Tulio A.M. Toffolo,H.G Santosa,"Variable neighborhood search accelerated column generation for the nurse rostering problem",vol 58,pp.31-38,2017
4. E.Rahimian, K.Akartunali, J.Levine,"A hybrid integer and constraint programming approach to solve nurse rostering problems",Computers and Operations Research ,Vol 82,pp.83–94,2017
5. I.X Tassopoulos, I.P. Solos and G.N. Beligiannis,"A two-phase adaptive variable neighborhood approach for nurse rostering",Computers and Operations Research,Vol 60,pp.150-169,2015
6. S.Asta,E.Ozcan,T.Curtois,"A tensor based hyper-heuristic for nurse rostering",Knowledge-Based Systems,Vol 98,pp.185–199,2016
7. J.Goua,Yu.Leia, W.Guoa,C.Wanga,Y.Caia,W.Luoa,"A novel improved particle swarm optimization algorithm based on individual difference evolution",Vol 57,pp.468-481,2017
8. Z. Lu, J.k.Hao," Adaptive neighborhood search for nurse rostering", European Journal of Operational Research, Volume 218, Issue 3, Pp. 865-876,2012
9. M.A. Awadallah, A.L Bolaji, M.A Al-Betar," A hybrid artificial bee colony for a nurse rostering problem",Applied Soft Computing, Volume 35, October 2015, pp. 726-739
10. C.Valouxis, C.Gogos, G.Goulas, P.Alefragis, E.Housos", A systematic two phase approach for the nurse rostering problem", European Journal of Operational Research, Volume 219, Issue 2, 1 June 2012, pp. 425-433

11. R.Bai, Ed.K.Burke, G.Kendall,J.Li, B.McCollum", A Hybrid Evolutionary Approach to the Nurse Rostering Problem", IEEE Transactions on Evolutionary Computation, Volume 14,Issue 4, 2010, pp. 580-590
12. G. Weil, K. Heus, P. Francois, M. Poujade", Constraint programming for nurse scheduling",IEEE Engineering in Medicine and Biology Magazine, Volume 14, Issue 4,1995,pp. 417-422

Appendices

These are the few screenshots of the code of the algorithm.

```php
<?php

mysql_query("UPDATE employee, (SELECT @id := 0) dm
             SET num = (@id := @id + 1) ");
mysql_query("UPDATE requirement, (SELECT @id := 0) dm
             SET day = (@id := @id + 1) ");

$run1=mysql_query("SELECT count(num) from employee");
$emp_counter=mysql_fetch_array($run1);
$run2=mysql_query("SELECT count(day) from requirement");
$day_counter=mysql_fetch_array($run2);

$employs=$emp_counter[0];
$days=$day_counter[0];
$row_counter=1;
$count=1;


while($row_counter<=$days)
                    {
                        $slot1_query=mysql_query("select slot1 from requirement where day=$row_counter");
                        $slot1_value=mysql_fetch_array($slot1_query);
                        $slot2_query=mysql_query("select slot2 from requirement where day=$row_counter");
                        $slot2_value=mysql_fetch_array($slot2_query);
                        $slot3_query=mysql_query("select slot3 from requirement where day=$row_counter");
                        $slot3_value=mysql_fetch_array($slot3_query);
                        $slot4_query=mysql_query("select slot4 from requirement where day=$row_counter");
                        $slot4_value=mysql_fetch_array($slot4_query);
                        $slot5_query=mysql_query("select slot5 from requirement where day=$row_counter");
                        $slot5_value=mysql_fetch_array($slot5_query);
                        $slot6_query=mysql_query("select slot6 from requirement where day=$row_counter");
                        $slot6_value=mysql_fetch_array($slot6_query);
```

```php
while($slot1_value[0]>0||$slot2_value[0]>0||$slot3_value[0]>0||$slot4_value[0]>0||$slot5_value[0]>0||$slot6_value[0]>0)
            {
                if($slot1_value[0]>0)
                    {
                      if($count>=$employs)
                        {
                            $count=1;
                        }
                      $slot1_value[0]--;
                      $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");

                      if(mysql_num_rows($nurse_query)==0)
                      {
                            $count++;
                            $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
                      }
                      $nurse_value=mysql_fetch_array($nurse_query);
                      $add_query=mysql_query("UPDATE  authority_view SET slot1=concat(ifnull(slot1,''),'$nurse_value[0] ') WHERE d
                      $count++;
                    }
                if($slot2_value[0]>0)
                    {
                     if($count>=$employs)
                      {
                        $count=1;
                      }
                      $slot2_value[0]--;
                      $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
                       if(mysql_num_rows($nurse_query)==0)
```

```
if($slot2_value[0]>0)
        {
          if($count>=$employs)
           {
               $count=1;
           }
          $slot2_value[0]--;
          $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
           if(mysql_num_rows($nurse_query)==0)
          {
               $count++;
               $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
          }
          $nurse_value=mysql_fetch_array($nurse_query);
          $add_query=mysql_query("UPDATE  authority_view SET slot2=concat(ifnull(slot2,''),'$nurse_value[0] ') WHERE day=$row_counter");
          $count++;
        }
if($slot3_value[0]>0)
        {
          if($count>=$employs)
           {
               $count=1;
           }
          $slot3_value[0]--;
          $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
           if(mysql_num_rows($nurse_query)==0)
          {
               $count++;
               $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
          }
          $nurse_value=mysql_fetch_array($nurse_query);
```

```
if($slot4_value[0]>0)
      {
         if($count>=$employs)
          {
              $count=1;
          }
         $slot4_value[0]--;
         $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
         if(mysql_num_rows($nurse_query)==0)
         {
              $count++;
              $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
         }
         $nurse_value=mysql_fetch_array($nurse_query);
         $add_query=mysql_query("UPDATE  authority_view SET slot4=concat(ifnull(slot4,''),'$nurse_value[0] ') WHERE day=$row_counter");
         $count++;
      }
     if($slot5_value[0]>0)
      {
         if($count>=$employs)
          {
              $count=1;
          }
         $slot5_value[0]--;
         $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
          if(mysql_num_rows($nurse_query)==0)
         {
              $count++;
              $nurse_query=mysql_query("select name from employee where num=$count AND mini>0");
         }
```