Project ReportCollege Timetable Generator

short line

Debanjan Dhar (304) Gaurav Sarkar (302)

Abhik Chandra Saha (313) Susanta Kumar Khan (298)

Suchismita Pal (305) Ashikur Rahaman (2214)

Bidyut Chaki (330) Ankur Nayak (337)

8th August 2020

# Index

[**Index**](#_fp24ykkikjs9) **1**

[**Abstract**](#_fy3ty6afe1u7) **3**

[**Introduction**](#_86wezt6xhqoi) **4**

[Importance](#_fomdnyblu96x) 4

[Objective](#_wtsdcwm5nxqk) 5

[**Tools and Technologies**](#_v65uy57jje2t) **6**

[Tools](#_oyyslber0mqq) 6

[IDE](#_dmgi0i3q6s8d) 6

[Online Storage](#_g1xibvvbfift) 6

[Technologies](#_hsz739x8bmas) 6

[Front End](#_ph8b6k7sdmdt) 6

[Back End](#_3toy7nmv7802) 7

[**Project Organization**](#_wk86qokcj5b) **9**

[Roles and Responsibilities](#_wvexbhv336le) 9

[Backend](#_t2pp2etmy2oc) 9

[Front-end](#_i6drkp28hnau) 9

[Timetable Generation Algorithm](#_fa0tdeog3k60) 10

[Integration](#_nwcdldd9t01w) 10

[Documentation](#_8vvpi0cysbn4) 10

[Project Plan](#_mzygnqcn9jaz) 11

[**Software Development Model**](#_kjt4mrmpwm04) **13**

[Feasibility Study](#_gs3xfa23v3d7) 13

[Requirement Analysis](#_lovnuzsul5wy) 13

[Requirement Specification](#_ott7jmtgfyxk) 13

[User Interface Design](#_7nczprms9tzf) 13

[Coding](#_gd3991lgzrcx) 14

[Validation](#_4nk9v8i0uvg5) 14

[**General Structure of Project**](#_4fmdqhwvg7we) **15**

[DFD Level 0](#_wo6wp56r8kg0) 16

[DFD Level 1](#_61xiidov3dwn) 17

[DFD Level 2](#_ogyra7l22vt4) 18

[**User Interface Design**](#_2ydn2pd4r34) **23**

[index.html](#_8gab3tpunlob) 24

[dashboard.html](#_pz8wtd5uetm0) 25

[institute\_details.html](#_ma6041lxjlsp) 26

[full\_or\_half\_days.html](#_mn9d28ok8xu4) 27

[sem.html](#_z5d36qbghs1m) 28

[sub.html](#_w32kla2y1pe2) 29

[teachers.html](#_uci0dev7poar) 31

[result.html](#_a3vznqyhwt77) 32

[**Database Design**](#_bpdxqm7pj2ei) **33**

[ER-Model](#_bg0fr2k8vdij) 35

[Tables](#_fc3p6a27qkej) 36

[User Table](#_ns3c7qfbtq4j) 36

[Sem Table](#_oeaabgv2485b) 36

[**Timetable Generation Algorithm**](#_o44rkwb7zlzx) **40**

[**Backend Design**](#_3e9cydnsbfj3) **42**

[Part I - HTTP Request Handler (htdocsRoutes.js)](#_7i8x2uispz27) 42

[Part 2 - API Request Handler (apiRoutes.js)](#_aqzydde2zgl2) 42

[**Testing**](#_5fysecvef4j0) **44**

[Sign Up Test](#_2bwju8hk2bk9) 44

[Sign In Test](#_1kd7es3gjebr) 44

[Timetable Generation Test](#_twgk3ydkicca) 45

[**Acknowledgement**](#_ae8fjrkarf8e) **49**

# 

# Abstract

A college timetable is an arrangement of a set of lectures taught by one or more professors in which given constraints are satisfied. Creating such timetables manually is a complex and time-consuming process. It is also very error-prone.

Timetable Generator is an application to elevate that problem. It enables the users to create timetables easily and print them. By automating this process with a computer assisted timetable generator, we can save a lot of precious time.

Hence we have developed the understandable, efficient and portable application which could automatically generate good quality timetables within seconds.

# Introduction

The aim of the College Timetable generator is to automatically create timetables for faculty and students. The application is only focused on generation of timetables prioritizing the needs of the teachers. Once the preference and conditions of faculty are given by the user, the software generates a timetable schedule satisfying the given conditions.

There are two types of conditions that can be provided by the faculty:-

(a) Compulsory Conditions

(b) Optional Conditions

Hard Conditions - The application makes sure that no Hard Conditions are violated during the generation of timetable.

Soft Conditions - They are less significant than Hard Conditions. The application makes sure that a minimum number of Soft Conditions are violated during the generation of timetable.

Some hard constraints are obvious. Like, the timetable needs to be generated in such a way that the number of different courses with a number of subjects in each, handled by a limited faculty provided with their slots and timings does not overlap. Some others can be entered by the user. Optimally, it is desirable for the timetable generator to satisfy both conditions.

## **Importance**

1. No more Paperwork:

The toughest thing about manual routine management is the endless list of paperwork. Timetable management website with its simple and user- friendly interface helps users with **effective routine management**. With its introduction, school related operations became paperless and confusion-free.

1. No more Confusion:

Creating timetables manually is a headache for teachers. With timetable website is possible for you to automatically generate timetables effectively, sticking by the requirements. Timetable management software generates timetables considering the limitations such as availability of teachers, slot preferences, time duration etc.

1. It is Error-Free:

When you manually prepare a timetable, you tend to make errors. But this will be error free or having list possible errors.

## **Objective**

To make an error free,hassle-free,time efficient **Automatic Routine Generation System.**

# Tools and Technologies

The tools and technologies used are mentioned as:

## Tools

### IDE

1) **Visual Studio Code** — It supports debugging, syntax highlighting, intelligent code completion, code refactoring.

2) **Mysql Workbench** — It provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup etc.

### Online Storage

1) **Github** — It offers the distributed version control and source code management functionality.

## Technologies

### Front End

1) Designing a) **HTML** - HTML is the standard markup Language that was used in the project to write web pages.

b) **CSS** - Cascading Style Sheets is a style sheet language that

was used to style the web pages.

c) **Bootstrap** - It is an CSS framework. It was used in our web application to make it responsive and work on mobile as well.

2) **Ajax** - It is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. It was used to make asynchronous requests to our API backend.

3) **JavaScript** - It is the programming language able to update and change both HTML and CSS having calculating, manipulating, validating data features.

### Back End

1)**Node Js** - It is a cross-platform, JavaScript runtime environment that executes JavaScript code outside a web browser. It is used to create our server in the project.

2)**My SQL** - It is a relational database management system. It is also open-source and free and thus was selected to use in our project. It is used to store the data about the user and the timetables created by the user in our project.

|  |  |  |
| --- | --- | --- |
| **HTML** is the standard markup Language for web pages. | **CSS** describes how HTML elements are to be displayed on screen, paper, or in other media | **Bootstrap** is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. |
| **Ajax** is short for Asynchronous JavaScript and XML, which refers to a set of web development techniques rather than an actual programming language | **JavaScript** is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server | **Node.js** (Node) is an open source development platform for executing JavaScript code server-side. |

# 

# Project Organization

## Roles and Responsibilities

The project team had been divided into five parts for easier task management.

### Backend

|  |  |
| --- | --- |
| **Team Member Name** | **Responsibility** |
| Susanta Kumar Khan | Writing API for communication with Database. |
| Debanjan Dhar | Writing API for communication with Database. |
| Gaurav Sarkar | Designing & Normalizing the Backend Table. |
| Bidyut Chaki | Writing API for communication with Database. |
| Abhik Chandra Saha | Designing & Normalizing the Backend Table. |

### 

### Front-end

|  |  |
| --- | --- |
| **Team Member Name** | **Responsibility** |
| Abhik Chandra Saha | Designing of pages with html,css,js and bootstrap. |
| Ashikur Rahaman | Designing of pages with html, css, js and bootstrap. |
| Gaurav Sarkar | Making pages responsive & Integrating the Backend API with frontend. |

### Timetable Generation Algorithm

|  |  |
| --- | --- |
| **Team Member Name** | **Responsibility** |
| Debanjan Dhar | Coding the Timetable Generator Algorithm. |

### 

### Integration

|  |  |
| --- | --- |
| **Team Member Name** | **Responsibility** |
| Gaurav Sarkar | Integrating the Backend API with frontend using AJAX, JS. |

### 

### Documentation

|  |  |
| --- | --- |
| **Team Member Name** | **Responsibility** |
| Suchismita Pal | Writing details on project report, Writing the SRS |
| Ankur Nayak | Writing details on project report, Writing the SRS |
| Debanjan Dhar | Writing details on project report, Setting the structure of documentation report. |
| Abhik Chandra Saha | Writing details on project report |

### 

## Project Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.No.** | **Task** | **Sub-Task** | **Date** | | **Person(s) Allotted** |
| **Start** | **End** |
| 1. | Research on project related technologies | Node.js, PHP, Ajax, MySQL, Bootstrap, JavaScript, Materialize.css | 01-08-2019 | 05-08-2019 | Gaurav Sarkar, Debanjan Dhar, Abhik Chandra Saha, Susanta Kumar Khan, Bidyut Chaki, Ashikur Rahaman |
| 2 | Detail design of project | ER model, DFD, Normalization on backend table | 06-08-2019 | 08-08-2019 | Gaurav Sarkar, Abhik Chandra Saha, Debanjan Dhar, Susanta Kumar Khan |
| 3 | Work on front-end | GUI design, Designing of buttons, Integrating the Backend API | 10-08-2019 | 06-01-2020 | Abhik Chandra Saha, Gaurav Sarkar, Ashikur Rahaman |
| 4 | Work on back-end | Writing API for communication with Database | 30-12-2019 | 17-02-2020 | Susanta Kumar Khan, Gaurav Sarkar, Debanjan Dhar, Bidyut Chaki |
| 5 | Work on algorithm | Coding the Timetable Generator Algorithm | 01-02-2020 | 02-03-2020 | Debanjan Dhar |
| 6 | Testing | Unit testing, testing of page transitions, login and signup, interaction with the database | 05-02-2020 | 01-08-2020 | Debanjan Dhar, Abhik Chandra Saha ,Gaurav Sarkar, Ashikur Rahaman, Susanta Kumar Khan, Bidyut Chaki, Ankur Nayak, Suchismita Pal |

### 

# Software Development Model

## Feasibility Study

During the Feasibility Study, we tried to have a rough understanding of what is required to be done from different faculty members of our college like S.H. Sir, B.R. Sir.

One of the major risks was identified to be our Timetable Generation Algorithm. Then different solutions were proposed and research was done. Finally, the Generic Algorithm was decided upon.

## Requirement Analysis

In Requirement Analysis, we tried to have a complete understanding of what is required to be done by talking to teachers and students. Here, we identified key inputs and features needed in our application.

## Requirement Specification

The user requirements were then systematically organized into a Software Requirements Specification (SRS) document. It is attached herewith the project.

## User Interface Design

User interface design generally refers to the visual layout of the elements that a user might interact with.

We decided upon making a web application so that it could be accessed on mobile as well.

## Coding

Coding is the use of computer programming languages to give computers and machines instructions on what actions to perform. It is the way humans communicate with machines.

We decided upon making a web application so that it could be accessed on mobile as well.

We have used a) HTML b) CSS c) BootStrap d) Ajax e) JavaScript (Front End) 1) Node Js , 2) My SQL (Back End)

## Validation

Validation is the documented process of demonstrating that a system or process meets a defined set of requirements.

The scope of validation process plays an important role as it mainly focuses on verifying the deliverables that are to be handed over to the customers. The primary function is to achieve deliverables that meet the standards and criteria mentioned by the customers in the project management plan.

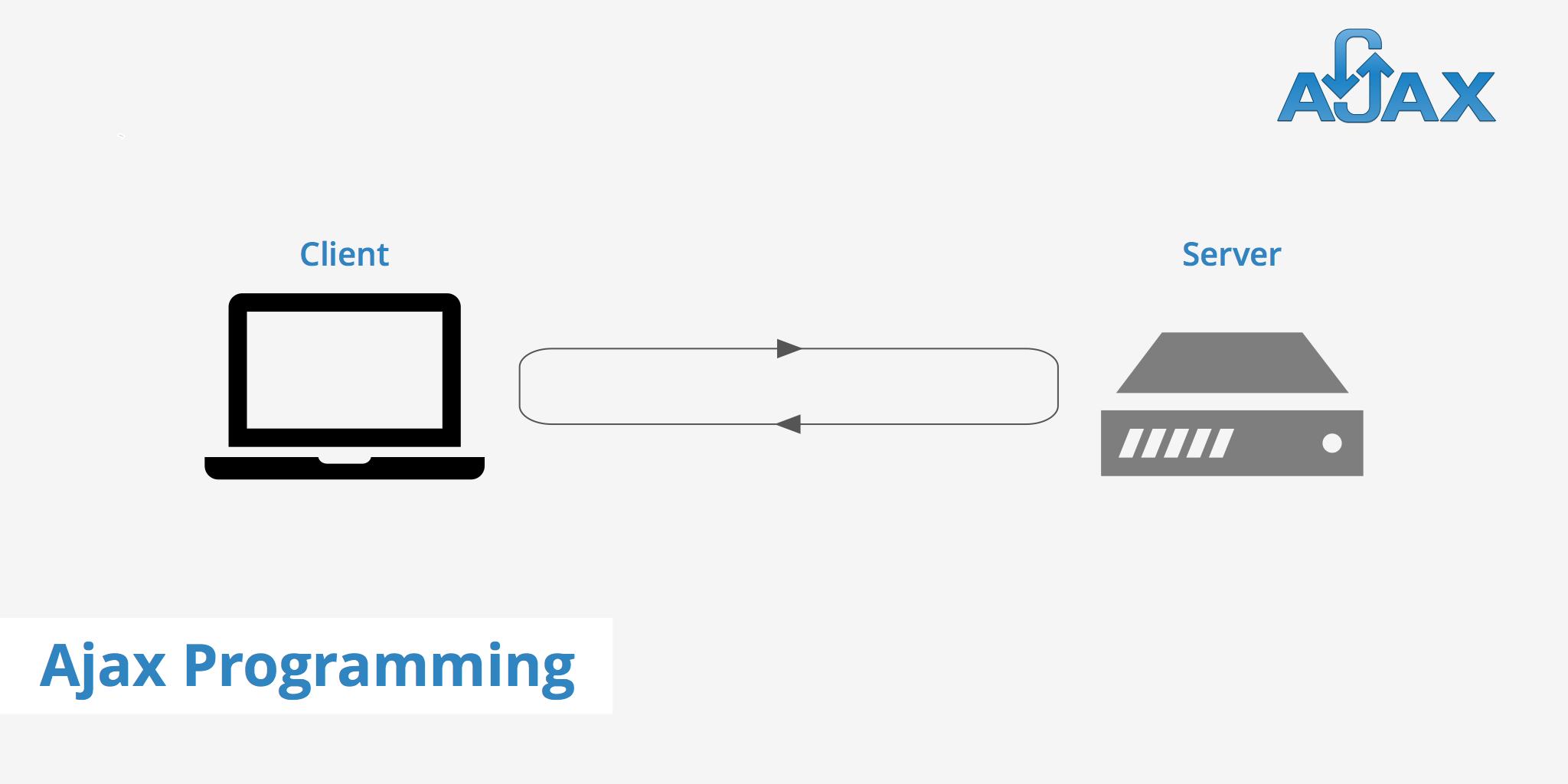
In our project, the validation was done by showing our project to our project guide since one of the main goals of the project from the start was to ease the generation of timetable in our college.

# General Structure of Project

The web application is divided into two parts:

Back End - It consists of the Node.js Server and DB.

Front End - It consists of the HTML User Interface that the user sees.



The front end and back end are interconnected via Ajax. Ajax (Asynchronous JavaScript And XML) allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

DFD

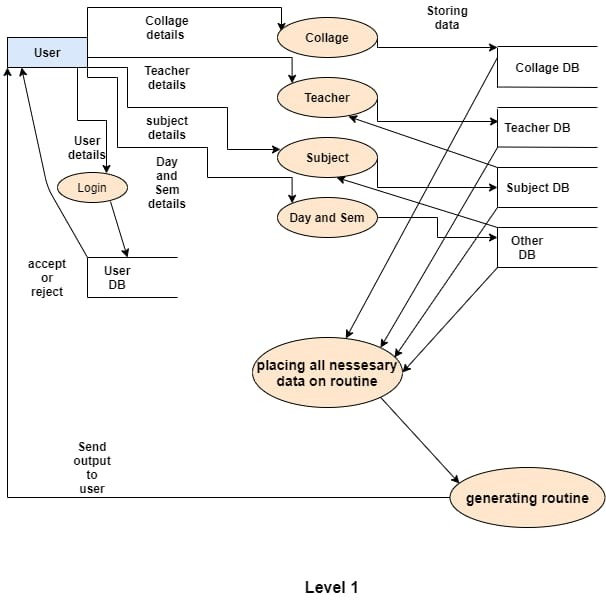
A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.

It helped us understand the system and played a key role in planning.

### DFD Level 0



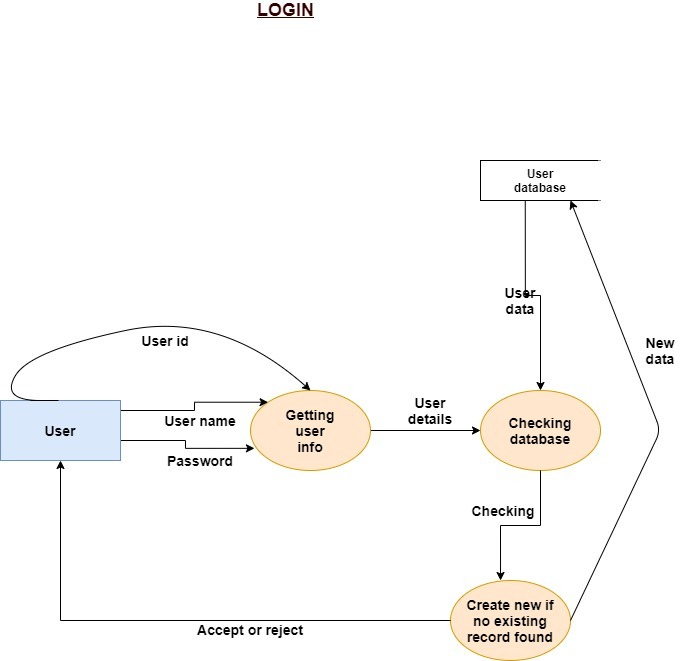
### DFD Level 1



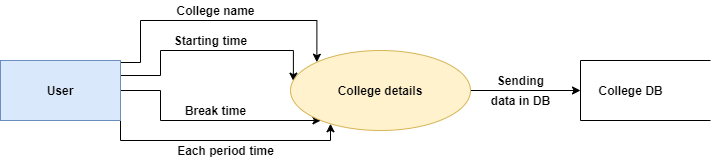
### 

### DFD Level 2

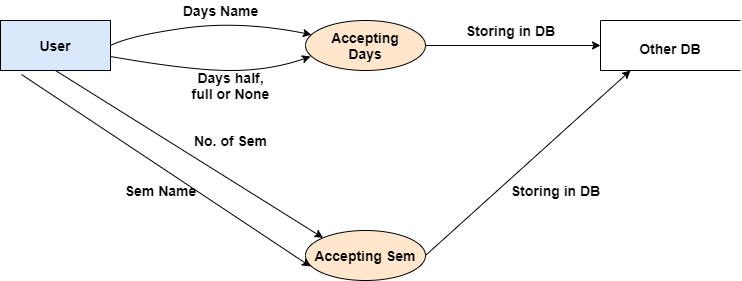
**Login**



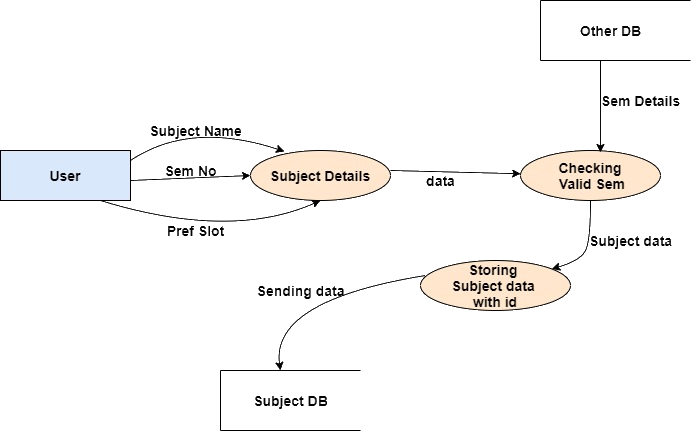
**Collage**



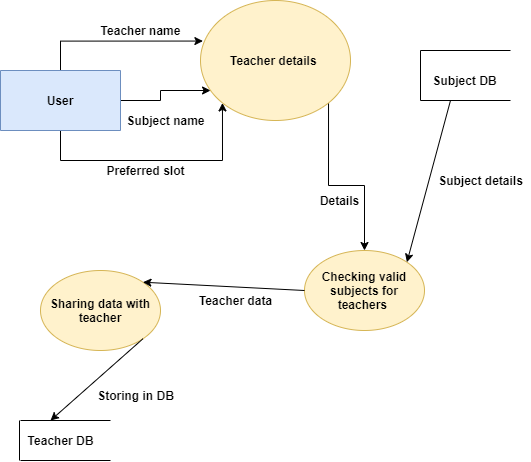
**Day & Semester**



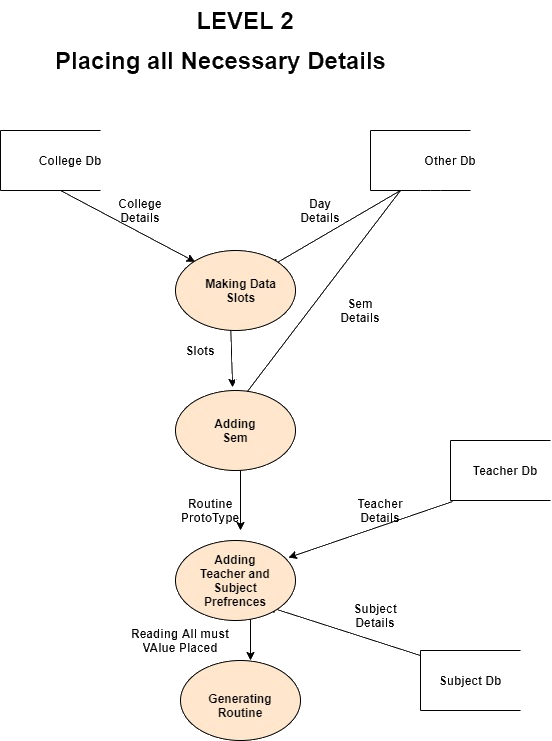
**Subject**



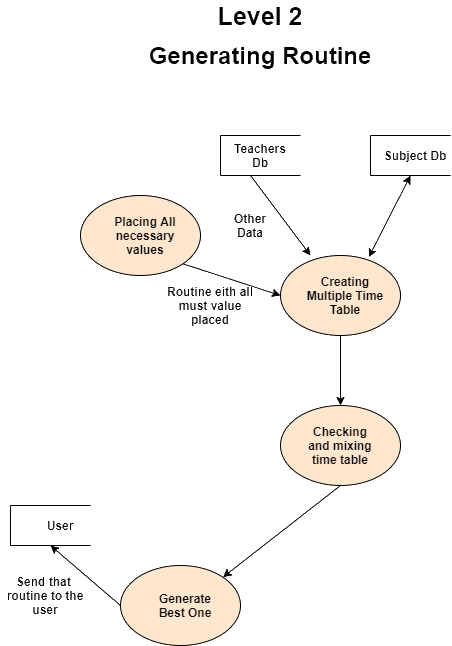
**Teachers**



**Planning all Necessary Details**

****

**Generating Routine**



# 

# User Interface Design

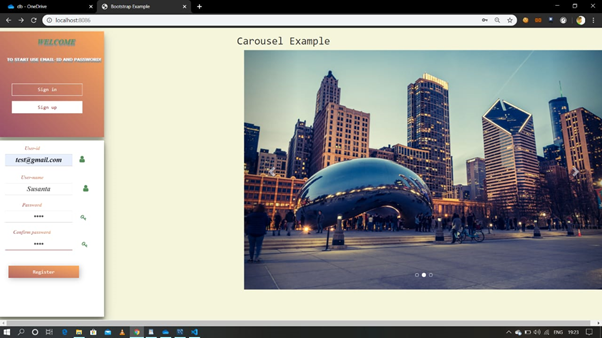
User interface design or UI design generally refers to the visual layout of the elements that a user might interact with in an application.

We understood that user interface design can dramatically affect the usability and user experience of an application. If a user interface design is too complex or not adapted to targeted users, the user may not be able to find the information or service they are looking for. In website design, this can affect conversion rates.

Thus, HTML, CSS, JS, Bootstrap have been used to design the user interface for communicating with the user flawlessly.

Bootstrap allowed our website to be usable on mobile devices naturally without zooming in.

## index.html



This is the first page of the web application. Here, the user needs to sign in / sign up before he/she can generate timetables.

1. To sign in, a user has to enter his/her email-id and password.  
   - Care has been taken to ensure the email and password match the one in the database.
2. To sign up, a user has to enter username, email-id and password.  
   - Care has been taken to ensure email format is valid and it doesn't already exist.  
   - Care has been taken to ensure the username is not taken.

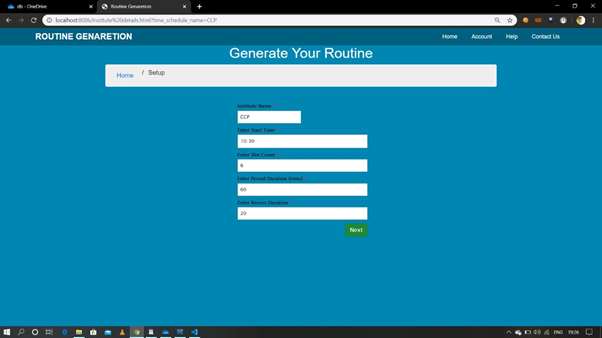
## dashboard.html



After successfully login, the user will be redirected to the dashboard page.

1. Here, navigation bars are given at the top where the title of our website is present including some links to other pages like Account, Help, Contact Us and also Home.  
   This navigation bar is also included in all other pages except index.html.
2. Here, you can see your previously generated routines neatly arranged in cards. Clicking on them allows the user to view the timetable as well as edit the input to timetables.
3. There is also an option to generate a new timetable.
4. The username of the user is also shown in this page.

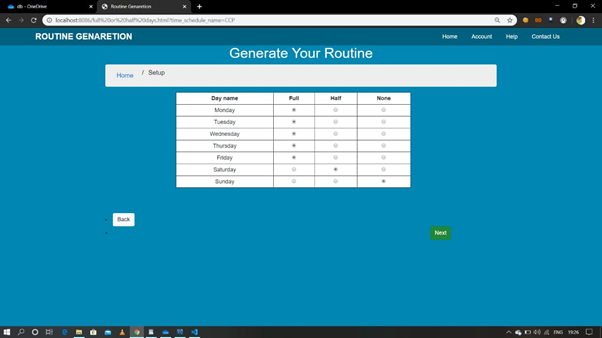
## institute\_details.html



This page allows the user to edit the institute details of a timetable. The user can reach this page by clicking on a timetable in the dashboard page.

1. Here the details like institute name, starting time of the institution, how many slots will be there in the routine, period duration, and the recess duration can be entered.
2. After entering the required details the user will have to click on the next button to go to the next page.
3. When the user clicks on the next button, the user will be redirected to the next page.

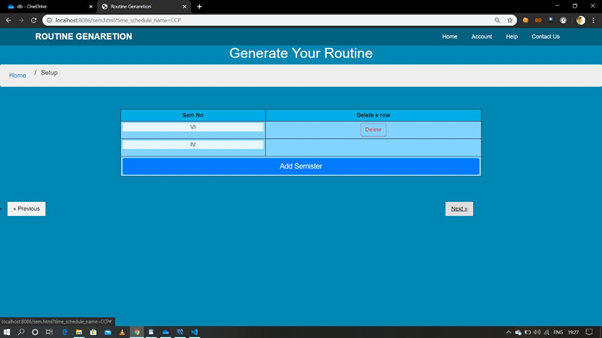
## full\_or\_half\_days.html



This page allows the user to edit the no of working days in a week of the institute. It also gives the option to set the no of days.

1. In the fourth page the user will see a table.
2. In the first column of the table, the name of the seven days is mentioned.
3. Second, third, fourth column contain radio boxes to select that the days are going to be Full or Half period or none means holiday.
4. After selecting those days the user will have to click on the next button to go to the next page.

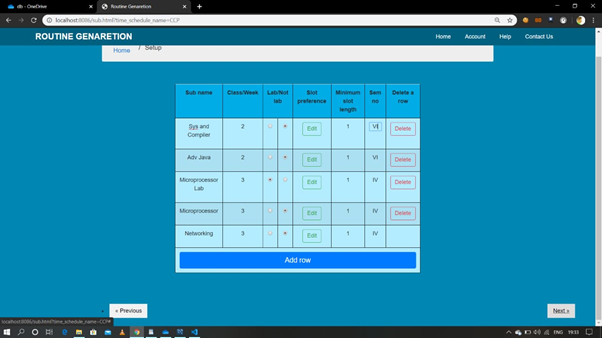
## sem.html



This page allows the user to create and edit semesters for one or multiple departments.

1. Semesters can be deleted and added by the user as per their requirement.
2. By clicking on the “delete” button, users can delete the rows and also can add by clicking on the “add semester” button.
3. The first column of the table contains the semester name - this can be edited by the user.
4. After selecting those days the user will have to click on the next button to go to the next page.

## sub.html



This page allows the user to create and edit subjects for the timetable.

1. In the first column users have to write the subjects.
2. In the second column users will write the number of classes/weeks.
3. Third column is to select whether the mentioned subject is lab or not.
4. Fourth column contains edit slot preference buttons. Clicking it will open a pop up where the user will be able to select the preferred days and periods of the subject. This is a hard constraint.
5. Fifth column is to select the no of classes that must be taught together for the subject to be taught.
6. Sixth column is to select the sem where the subject is taught.
7. To delete a subject, users need to click the delete button on the row of the subject.
8. Users can also click “Add row” to add a new subject.
9. After adding and editing the subjects, the user will have to click on the next button to go to the next page. This will redirect the user to the next page.

## 

## teachers.html

# 

This page allows the user to edit teachers as per their need and also this page gives flexibility for adding and deleting rows and also there is an option for "Add teachers" too.

1. To delete the row, the user needs to click the "Delete" option .
2. To edit teachers, the user needs to select the "Edit " option as well.
3. To edit availability the user needs to click "Edit ".
4. Users will be able to see "Teachers name " as per column basis .
5. After adding and editing the availability , teachers, and subjects taught by teachers, the user will have to click on the next button to go to the next page.

## result.html

## 

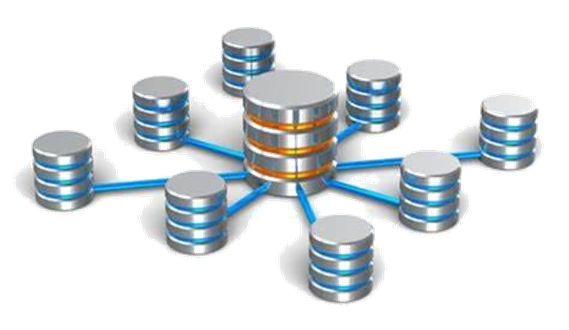
This page shows the final generated timetable. The user can click the “next routine” button to generate different timetables if he/she doesnt like the current one.

# Database Design

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed databases are easy to maintain, improves data consistency and are cost effective in terms of disk storage space.

An RDBMS is used in our project to store the data of our users. The data consists of information about our users, the timetables created by the users along with options supplied to create the timetable. It helps produce database systems

Database designing is crucial to high performance database system.Data operations using SQL is relatively simple



**Importance of Database**

The important that we are using in our project for many reason such as :

1Efficient support for complex and interrelated business processes

2.Lower cost of database ownership

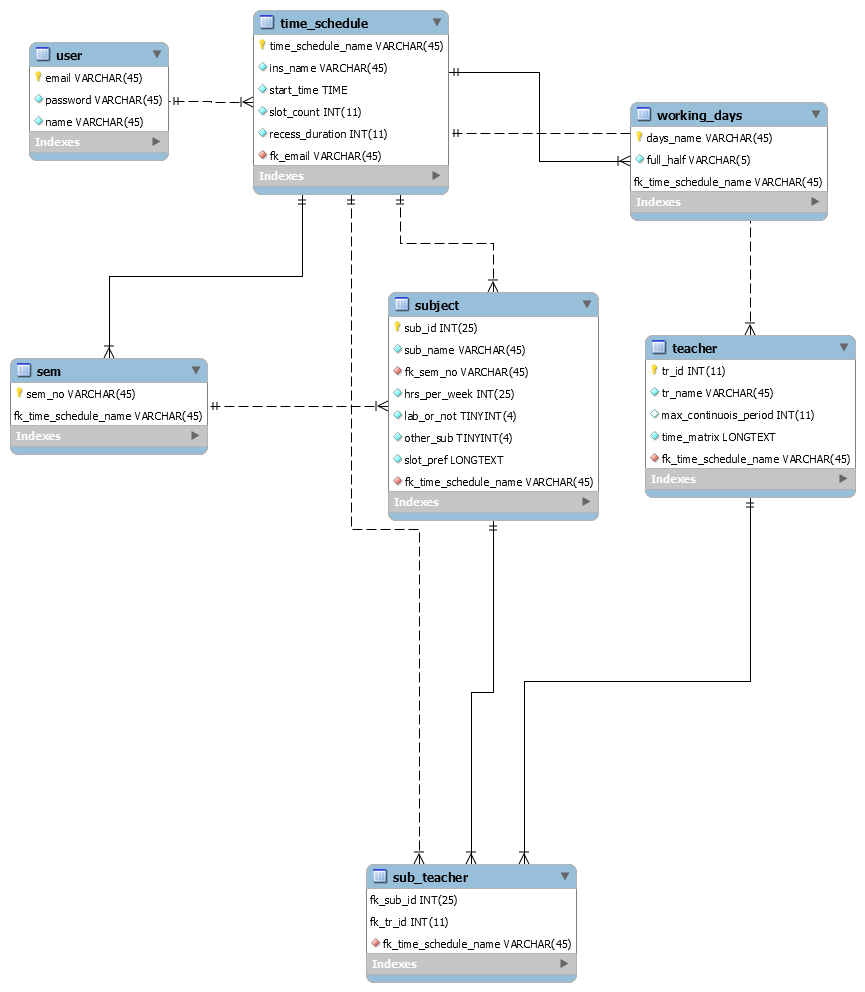
3.Consistent availability of data to support business operations and decision making

4.Reduction in redundant data storage

5.Increased productivity at work

6.Avoidance of inconsistent data

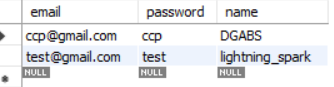
## ER-Model

An Normalization, the ER model of the database was first drawn before actual implementation of the database. It enabled us to have better visualization of db.

## Tables

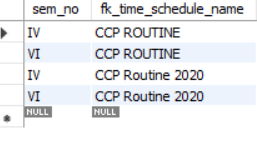
Show each table screenshot along with a small description. - Suchi. Also, maintain format below.

### User Table

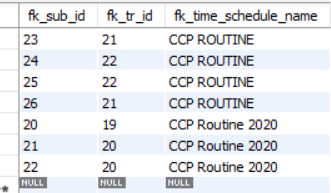


1. email - It is used to store the email of the user. The field is of type varchar2(50).
2. password - It is used to store the password of the user. The field is of type varchar2(50).
3. name - It is used to store the username of the user. The field is of type varchar2(50).

### Sem Table



1. sem\_no - It is used to store sem of students.This field is of type varchar2(20)
2. fk\_time\_schedule\_name-It is used to store the name of the routine generated.This field is of type varchar2(100)

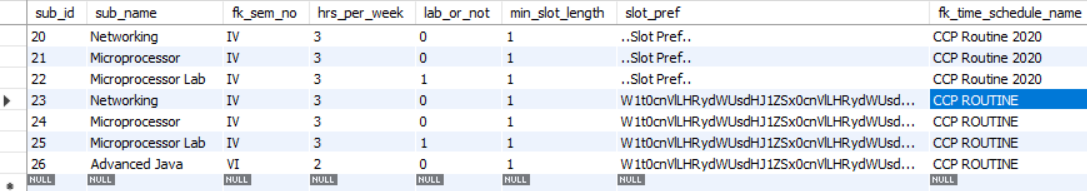
Subject Teacher Table

# 

# 

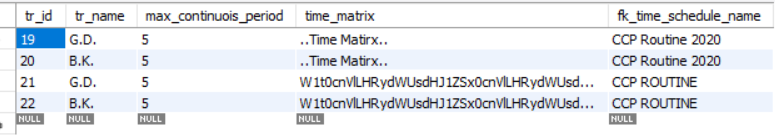
1. fk\_sub\_id - It stores subject id from subject table.It is foreign key here of Sub id in Subject Table.
2. fk\_tr\_id - It stores teachers id from teachers table.It is foregin key here of Teacher id in Teachers Table.
3. fk\_time\_schedule\_name - It is basically the names of different timetables generated by the user.

Subject Table



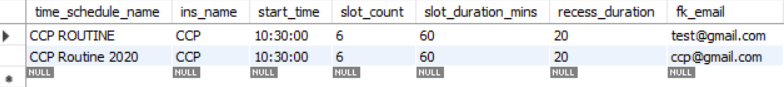
1. sub\_id -It is the primary key of the table storing unique id of each subject .
2. sub\_name -It stores the name of each subject to be added in routine.
3. fk\_sem\_no - It stores the sem number ,from the sem table.
4. hrs\_per\_week - It stores Hours of class in week.
5. lab\_or\_not - It gives the faculty to choose whether the class to be taken iss lab class or normal class.
6. min\_slot length - It stores the minimum value of how long a single session of class can go.
7. slot\_pref - It stores the slot preference of teachers.
8. fk\_time \_schedule\_name - It stores the name of different generated routines.

Teachers Table



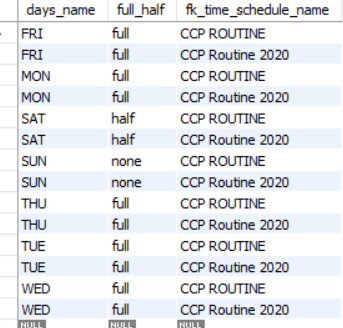
1. tr\_id- It means “Teachers id” and is used stored in this column,which is unique for each teacher.
2. tr\_name-It stores the names of teachers.
3. max\_continuois \_period-Teacher can choose and fill the max continuous classes they can take so it stores their choice.
4. time\_matrix-It contains strings of preferences of teachers.Which they prefer is stored here in a string.
5. fk\_time \_schedule\_name - It simply stores the name of routines. It is of varchar2(50) type.

Time Table



1. time\_schedule\_name - It contains the names of routines generated.It is inserted by the user.
2. ins\_name - It contains the institution name which can be taken as user input.
3. start\_time - It contains when a routine is started.
4. slot\_count - It contains how many slots will it have in that routine in one row that is classes.
5. slot\_duration\_mins - It contains the slot duration ,i.e length of classes preferred.
6. recess\_duration - It contains the duration of recess break.
7. fk\_email - It contains the email of a routine creator.

Working Days



# 

# 

# 

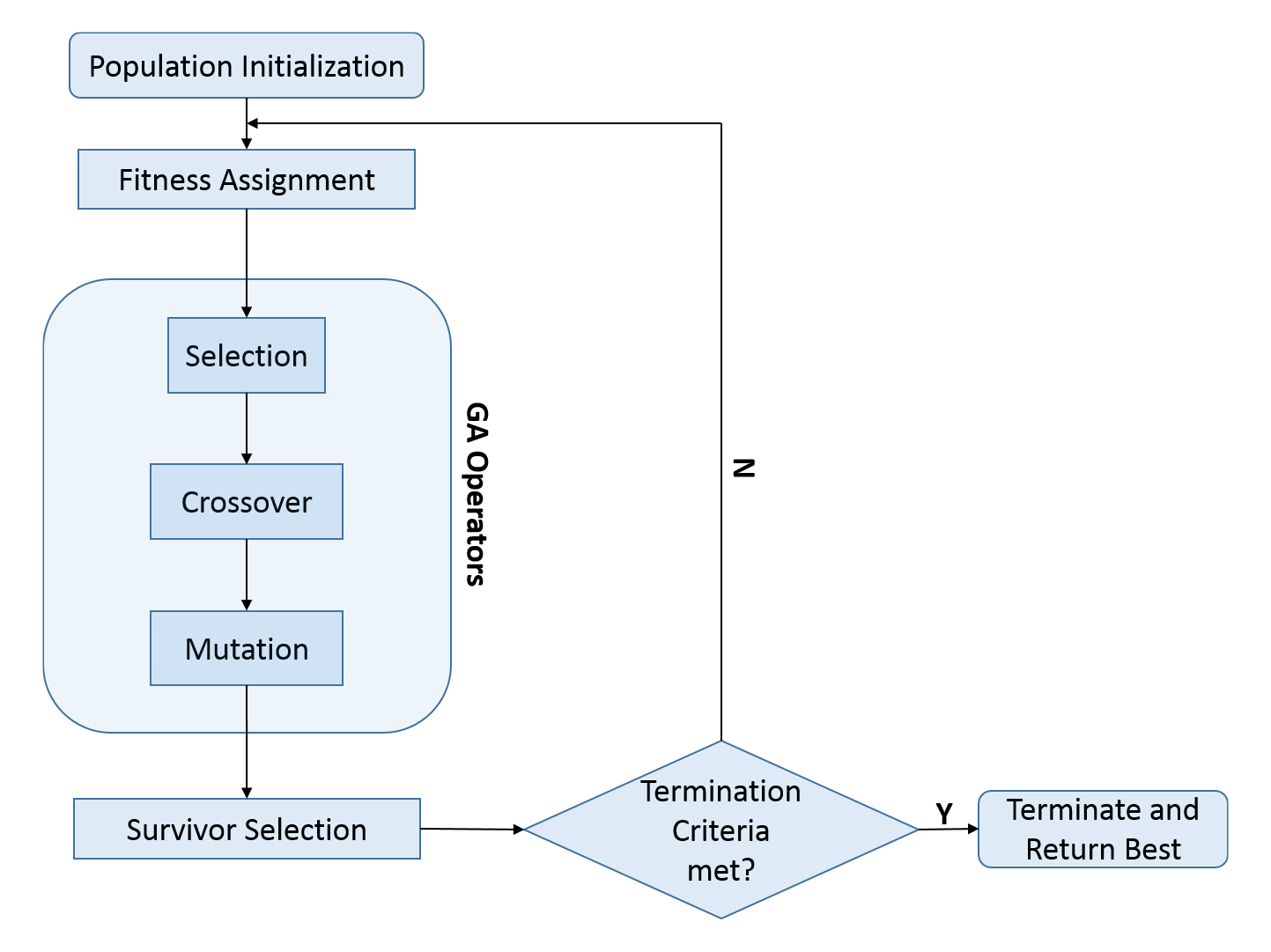
# 

1. days\_name - It contains the name of week days which is used in routine.
2. full\_half -I t contains the preference whether that day is a full day or half day.
3. fk\_time\_schedule\_name - It contains the name of a routine generated.

# Timetable Generation Algorithm

The problem of timetable generation is a famous one - it is known as the Scheduling Problem. It is a NP-Hard problem. That is, there is no existing algorithm that can solve this problem in polynomial time.

However, such problems can still be solved by algorithms that do trade-offs per se – reducing the optimality of the solution to increase the speed. This type of algorithm which trades optimality i.e. correctness, for time is called heuristic algorithm.



One such algorithm called Generic Algorithm was thus chosen.

Genetic algorithms are commonly used to find good quality solutions to search problems. It was introduced in 1960 and was based on Darwin’s theory of evolution. The general idea is to generate solutions randomly at the beginning and then, do crossovers between solutions while selecting good parts to create a new solution. The new solutions are called offsprings and this process is repeated over several generations until we are satisfied with the solution. This idea is similar to how evolution works.

In this case, different timetables were first generated.

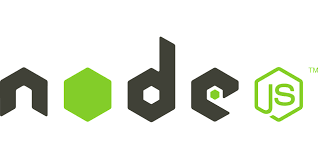
Then, the timetables were made to cross-over over thousands of generations. The cross-overs were done between timetables randomly selected with weight on the goodness. Here, by "goodness", it is meant that a fitness function that calculates agreement of each subject’s placement by different input constraints by the user. Also, by “randomly selected with weight on the goodness”, I mean solutions with greater goodness have a greater chance of being selected.

Mutations were done in-between generations. Mutation refers to randomly changing a part of a solution for diversity. That is, the algorithm was made to randomly change a subject’s position in-between generations. This is an important step as without mutation the solutions become too similar to each other and fail to diversify and look for better traits to generate an even better solution.

It was made to stop after it completed thousands of generations. Then the final timetable i.e. the timetable with most goodness was made to output.

# Backend Design

The backend server is written in node.js. Node JS is an open source and cross platform runtime environment for executing JS in backends.



For simplicity and convenience, in our application, the server is split into two parts - HTTP Request Handler and API Request Handler.

## Part I - HTTP Request Handler (htdocsRoutes.js)

It handles all static HTTP Document requests as well as requests for other types of media like images.

Since, this is based on node.js, it is more performant than php servers in cases like these - mainly due to its non-blocking, event-driven I/O paradigm. Node.js is also lightweight and efficient. It shines the best in cases like developing a web application.

## Part 2 - API Request Handler (apiRoutes.js)

An application programming interface (API) is an interface which defines interactions between multiple software intermediaries. Here, api is used for communication between static HTML documents and database.

Using API allows our application to maintain a higher level of security as well as makes eases the task of various teams easier. It allowed us to isolate tasks in a big team of 8 people.

It reduces the risk of database corruption. Also, it keeps open the possibility of easy releasing of an public api which will help other people and organizations interested to control our application programmatically.

# Testing

## Sign Up Test

**User action -** Clicks on ‘Register’ Button.(after filling up)

|  |  |  |
| --- | --- | --- |
| Test cases | Inputs | Successful or not |
| 1 | User\_name=”Gourav” ,  User\_email="[sakar66gorav@gmail.com](mailto:sarkargourav12@gmail.com)"  & Password=”noneed" | Successful |
| 2 | Username=”RGgroup” ,  User\_email="[RGgroup@gmail.com](mailto:RGgroup@gmail.com)"  & Password=”a123" | Successful |
| 3 | Username="” ,  User\_email=""  & Password=”yuoh" | Unsccessful  Due to  Empty field(s) |

## Sign In Test

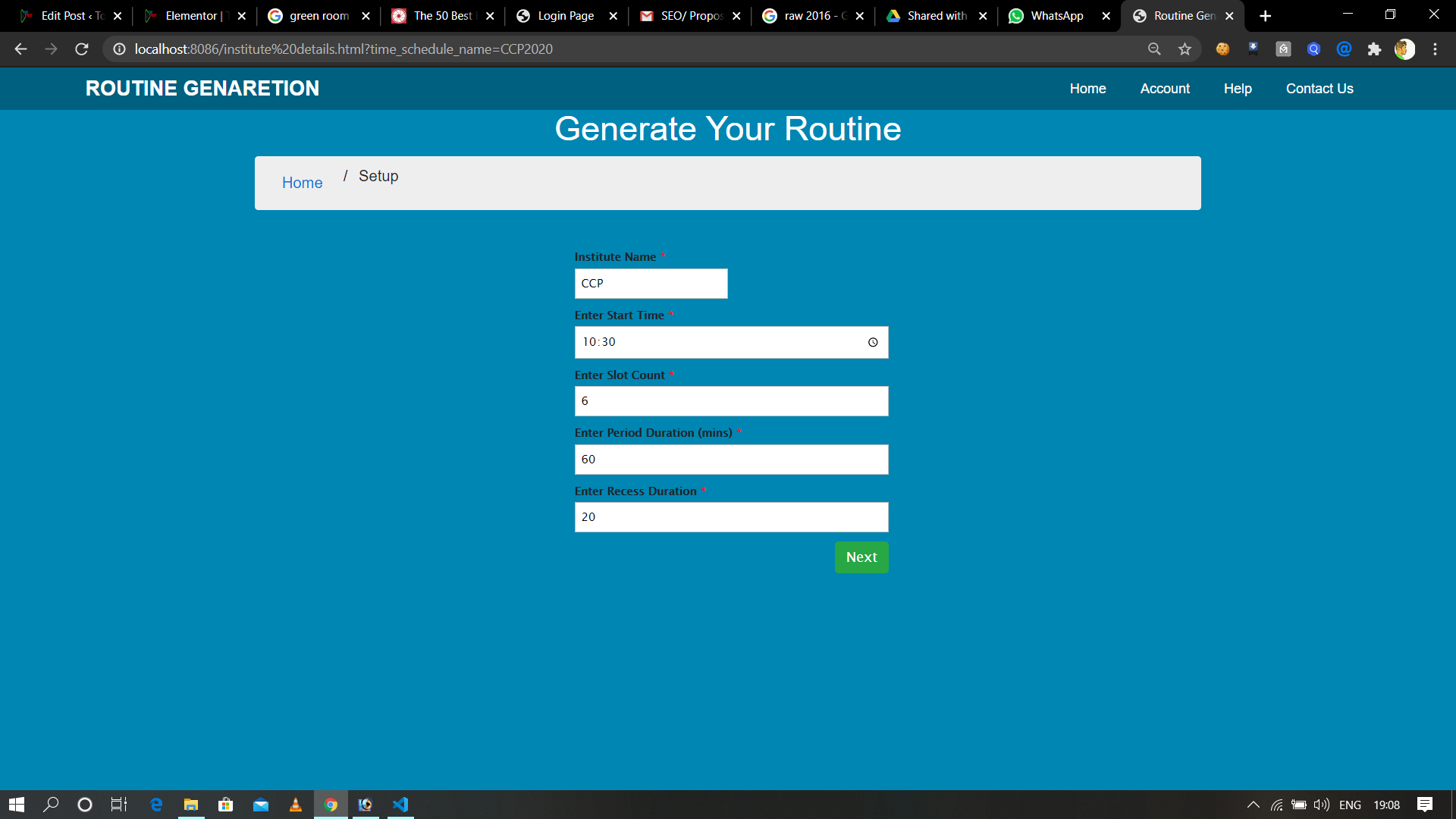
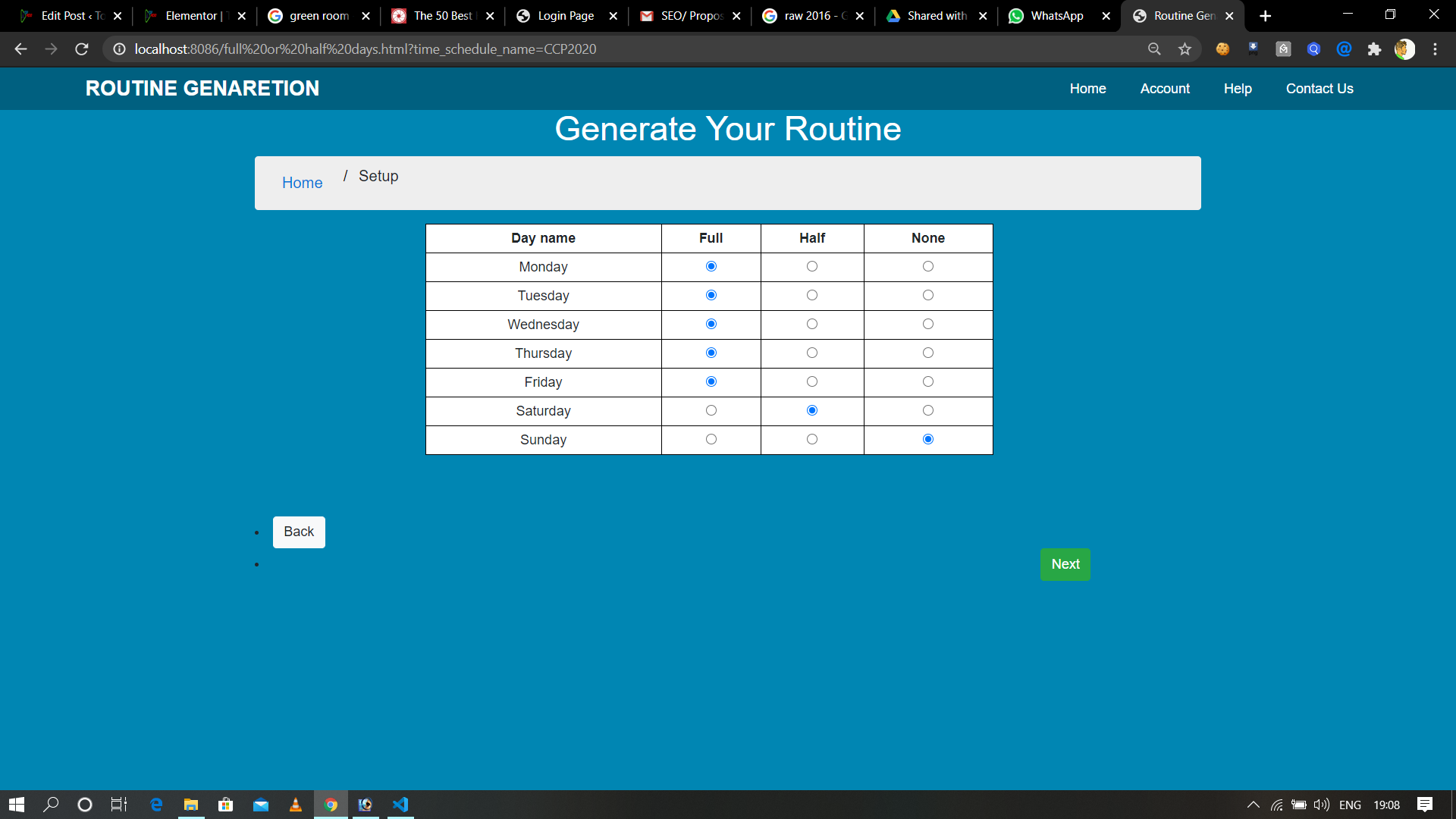
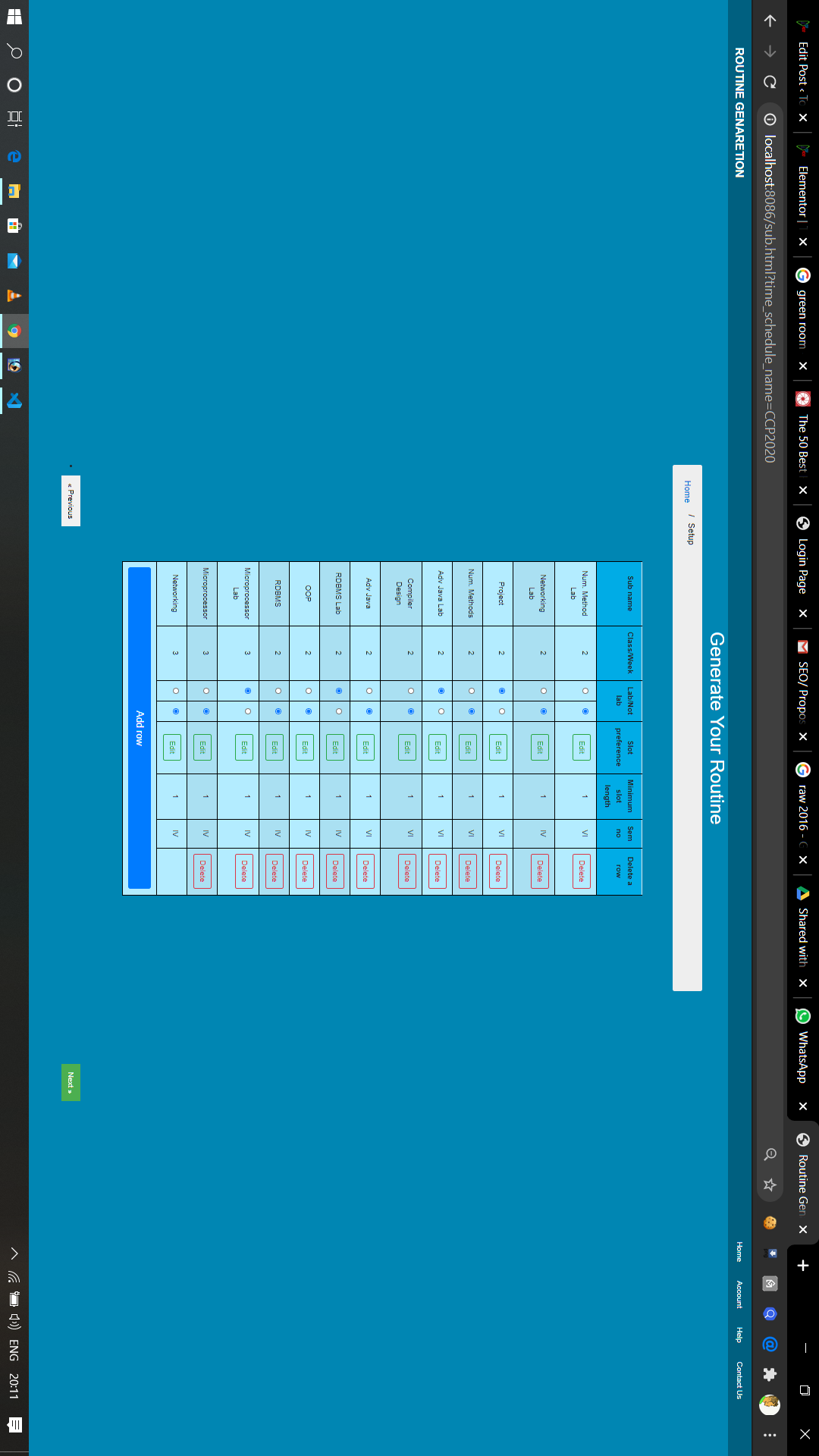
**User action**- Clicks on ‘login’ Button.(after filling up)

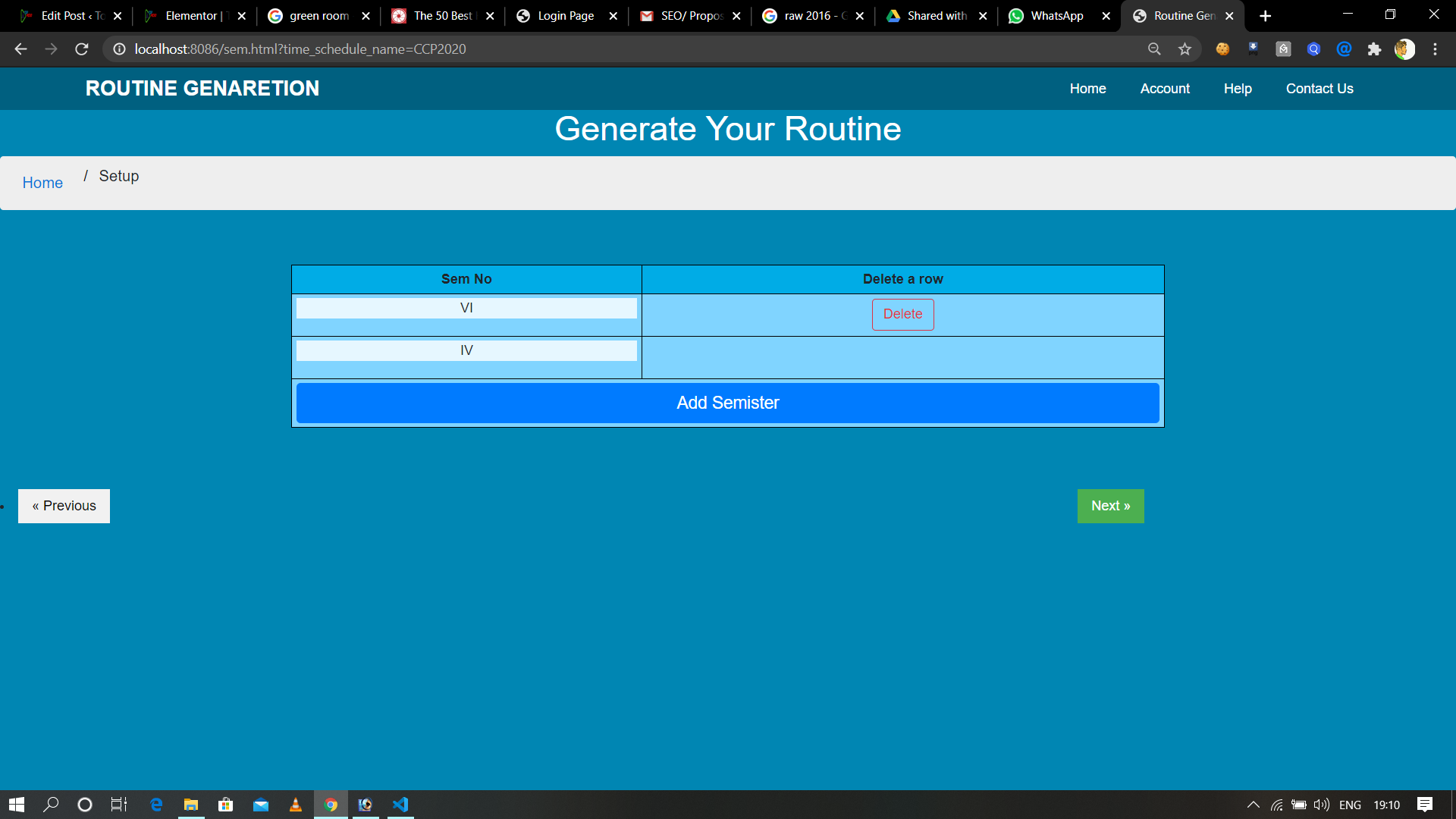
|  |  |  |
| --- | --- | --- |
| Test cases | Inputs | Valid or invalid |
| 1 | User\_email="[sakar66gorav@gmail.com](mailto:sarkargourav12@gmail.com)"  & Password=”noneed" | “valid” |
| 2 | User\_email="[aaabdgss@gmail.com](mailto:sarkargourav12@gmail.com)"  & Password=”ccp" | “Invalid username/password” |
| 3 | User\_email=""  & Password=”" | “Invaild empty field(s)” |
| 4 | User\_email="[RGgroup@gmail.com](mailto:RGgroup@gmail.com)"  & Password=”a123" | “valid” |

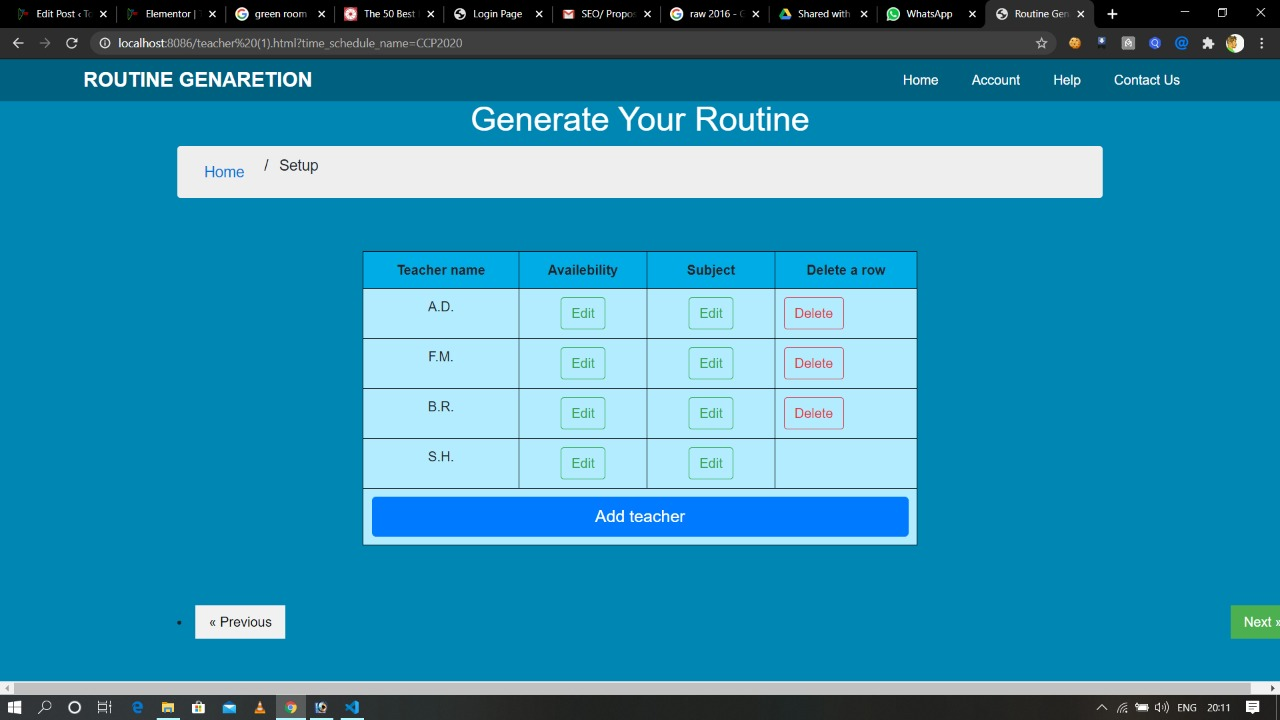
## Timetable Generation Test

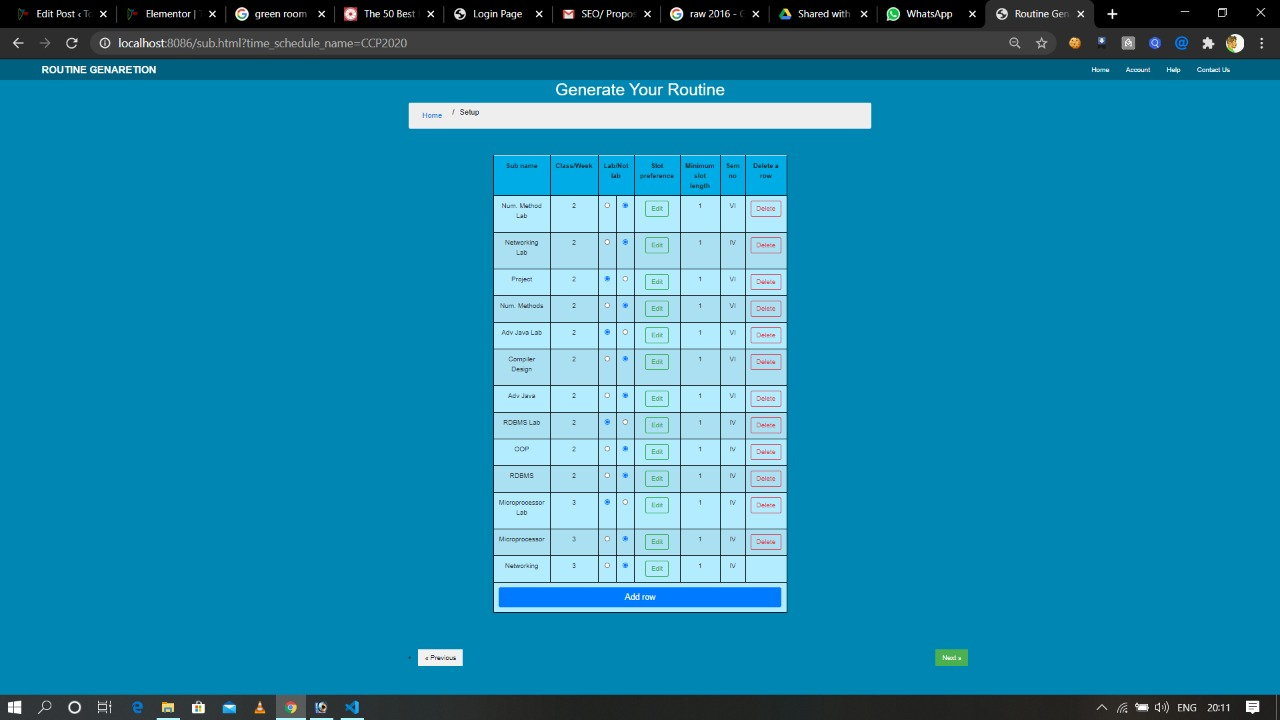
As a testing process, we have tried to create results with many types of inputs.We also generated routine with some college sem inputs .

Here is the screenshot of each inputs page with respected inputs-

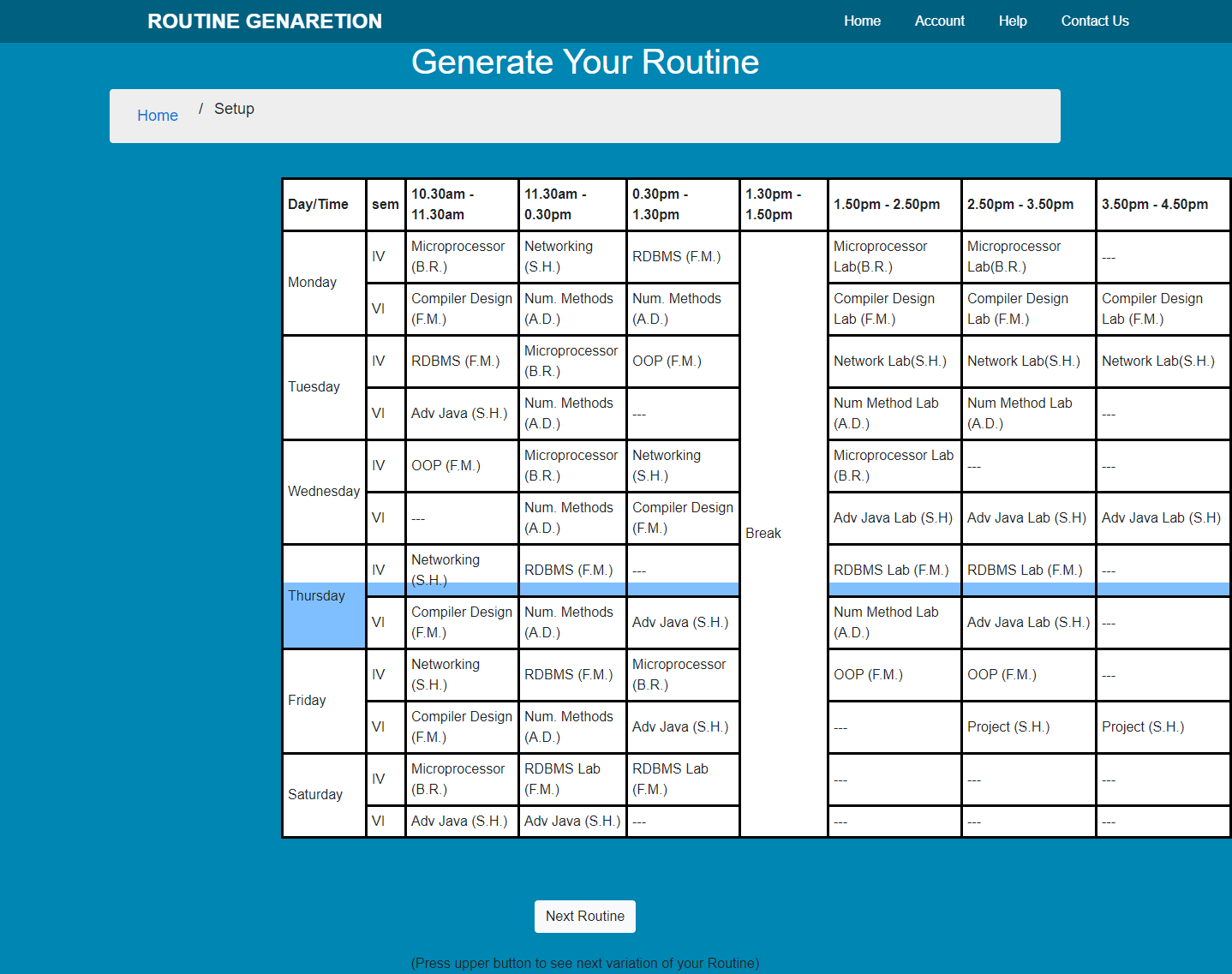








Output-



# Acknowledgement

Any project is like a bridge between theoretical and practical working. The satisfaction that accompanies the successful completion of such a project would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. This project consumed a huge amount of work, research and dedication.

We would like to express our special thanks of gratitude to our project guide - **Mr. Samaresh Hazra** who gave us the golden opportunity to do this wonderful project. It helped us know about so many new things which we are truly really thankful for. We are truly grateful to him for all the guidance, inspiration and constructive suggestions that he provided in helping us see the completion of this project.

We also feel obliged in taking the opportunity to sincerely thank Dr. Subrata Mondal (Principal of Central Calcutta Polytechnic). Moreover, we are highly obliged in taking the opportunity to sincerely thank all the faculty members of the computer science department for their generous attitude and friendly behaviour. We have no valuable words to express my thanks, but our hearts are still full of the favours received from every person.