COM1025 Web and Database Systems

Coursework Assignment

[A website showcasing university societies and students]

URN	6845721
Username	gb01005
Name	Ganesh Sri Ram Bhavanasi
Date	Date Started: 13/11/2023

1 Business Rules and Assumptions

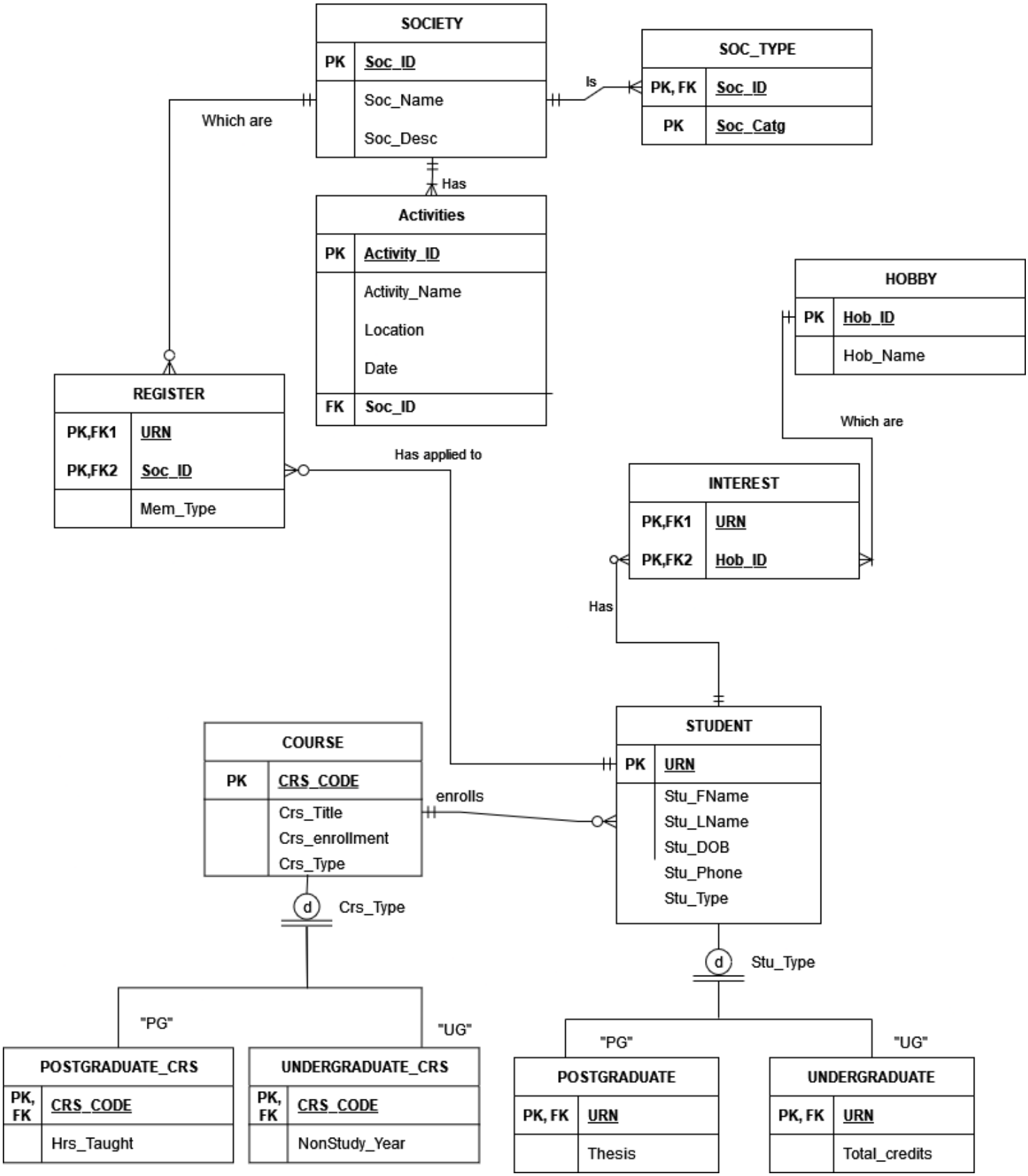
1. A university offers courses and has two categories of students: undergraduates and postgraduates.
    a. The university needs to store the following details about each student: name, date of birth, a phone number and what course they are on.
        i. We will assume that each student will only go by 1 (full) name. Also, that they will have only 1 phone number, which is 11 digits long. I will only allow those over the age of 18 to be entered as a student.
    b. A student must be a postgraduate or an undergraduate, but they cannot be both at the same time.
        i. A Student enrolled in an undergraduate course has a total number of credits.
        ii. A student enrolled in an postgraduate course has a thesis.
        iii. For simplicity, on the website when a undergraduate student is created, they are assigned 120 credits. And a postgraduate student's thesis is just called thesis.
    c. A student must be enrolled on one and only one course while a course can enrol no student or many students.
    d. A course has a course code, name and enrolment (total number of students on the course)
        i. A course must be a postgraduate or undergraduate course, but cannot be both at the same time.
        ii. Undergraduate courses include a non-study year, this is either a Year Abroad, or, a Placement Year, this is implemented as an **Enum**.
        iii. A Postgraduate course has a number of teaching hours that has to be done with the course.
        iv. An undergraduate student changing courses can only change to an undergraduate course, likewise for postgraduate students.
    e. A necessary assumption will be that the students in the database, are just some of the students in the university, i.e. there are other students in the university, that are not in the database. (This will help with keeping the database simple). For example, societies with no members from this database is still an active society (i.e. the members are students not in this database).
    f. A Student's URN should be 6 digits long and cannot contain any preceding zeros, (i.e. 100000 <= URN <= 999999

2. The university also wants to keep data on hobbies of students if they wish to provide that information so that appropriate clubs and societies can be suggested to them.
    a. A student can have no hobby or can have many hobbies.
    b. A hobby can be related to no student or to many students.
    c. The current list of hobbies include: reading, hiking, chess, Taichi, ballroom dancing, football, Tennis, Rugby, climbing, rowing. This is linked to the student via the associative entity Interest.

3. In the database that I will be creating, the database will also keep data on a list of societies available.
    a. A Student can be registered to no society or can be registered up to many societies.
    b. A Society can be related to none or can be related to many students. This comes from the assumption that there are students in societies who are not in the database.
    c. The current list of Societies include: Book Club, Chess, Martial Arts, Dancing, Water Sports, Physical Sports.
    d. The Society table will include an attribute briefly describing the society. There will also be an attribute describing the category of the society (i.e. Active, Relaxed, Difficult, In-person, and Online). This will be a **multi-valued attribute**, as a society can fit into multiple categories.
    e. As part of a society, a student can be a: Member, Chair, Treasurer, or Secretary. This will be implemented as an **Enum**, in the bridge entity Register. Multiple members are allowed to have the same membership type. E.g. There can be more than one chair (person) for a society.
    f. A Society must have at least one activity, i.e. Society has **1toM** relationship with activities. These describe events that the society hosts.

2 Extended Entity Relationship Diagram (EERD)

I have extended the EER diagram provided, using the diagrams.net website.



3 Logical Relational Database Schema

Course(Crs_Code, Crs_Title, Crs_enrollment, Crs_Type)

PRIMARY KEY: Crs_Code

Undergraduate_Crs(Crs_Code, NonStudy_Year)
PRIMARY KEY: Crs_Code
FOREIGN KEY: Crs_Code REFERENCES **Course**(Crs_Code)

Postgraduate_Crs(Crs_Code, Hrs_Taught)
PRIMARY KEY: Crs_Code
FOREIGN KEY: Crs_Code REFERENCES **Course**(Crs_Code)

Student(URN, Stu_FName, Stu_LName, Stu_DOB, Stu_Phone, Stu_Course,

Stu_Type)

PRIMARY KEY: URN
FOREIGN KEY: Stu_Course REFERENCES **Course**(Crs_Code)

Postgraduate(URN, Thesis)

PRIMARY KEY: URN
FOREIGN KEY: URN REFERENCES **Student**(URN)

Undergraduate(URN, Total_credits)

PRIMARY KEY: URN
FOREIGN KEY: URN REFERENCES **Student**(URN)

Hobby(Hob_ID, Hob_Name)
PRIMARY KEY: Hob_ID

Interest(URN, Hob_ID)
PRIMARY KEY: URN
PRIMARY KEY: Hob_ID

FOREIGN KEY: URN REFERENCES **Student**(URN)
FOREIGN KEY: Hob_ID REFERENCES **Hobby**(Hob_ID)

Society(Soc_ID, Soc_Name, Soc_Desc)
PRIMARY KEY: Soc_ID

Activities(Activity_ID, Activity_Name, Location, Date, Soc_ID)
PRIMARY KEY: Activity_ID
FOREIGN KEY: Soc_ID REFERENCES **Society**(Soc_ID)

Soc_Type(Soc_ID, Soc_Catg)
PRIMARY KEY: Soc_ID
PRIMARY KEY: Soc_Catg

FOREIGN KEY: Soc_ID REFERENCES **Society**(Soc_ID)

Register(URN, Soc_ID, Mem_Type)
PRIMARY KEY: URN
PRIMAY KEY: Soc_ID

FOREIGN KEY: URN REFERENCES **Student**(URN)

```
FOREIGN KEY: Soc_ID REFERENCES Society(Soc_ID)
```

4 Website Working with MySQL Database

- index.js – contains all the NodeJS code to make the website function. It connects to the database, loads the middleware, and has all the code for get and post requests. The code also contains validation.
- main.css – CSS file that contains all the CSS code (styling)
- footer.ejs – a common file that contains the footer, it is used in all the webpages.
- header.ejs - a common file that contains the header, it is used in all the webpages.
- index.ejs – The homepage. This contains the HTML to create/load the **homepage**.
- viewsocities.ejs – This contains HTML to create the webpage that lists all the societies.
- society_members.ejs – This contains HTML to create the webpage that lists all the members of societies.
- society_member_view.ejs – This contains HTML to create the webpage that shows which societies a specific society member has signed up to.
- students.ejs – This contains HTML to create the webpage that allows to **view all** the students.
- student_view - This contains HTML to create the webpage that allows you to view a specific student. This is the **view one** page.
- student_edit – This contains HTML to create the webpage that allows you to update one a specific student. This is the **update one** page.
- student_delete – This contains HTML to create the webpage that allows you to delete a specific student.
- student_deleted – This contains HTML to create the webpage which confirms if you have deleted a student or not.
- search.ejs – This contains HTML to create the webpage that allows you to search for data.
- search_query.ejs – This contains HTML to create the results page for the search query and type of search you have done.
- lake-background.jpg – This is an image file that is used on the homepage (index.ejs).

I have used the same middleware and modules as we have done in labs, (i.e. NodeJs, Nodemon, Express, BodyParser and MySQL).

5 Advanced Tasks

I believe I have translated the EERD to 3NF and all the tables I have created have appropriate constraints. This includes the appropriate CHECK constraint for Student URN.
The relationship between **Student** and **Hobby** is a many-to-many relationship, so I have created an appropriate bridge entity. Additionally, the relationship between **Student** and **Society** is many-to-many, as well, and so I have created an appropriate bridge entity.

I have created another specialisation hierarchy (**Course** and whether the course is undergraduate or postgraduate). This follows the requirements of a course in real life, i.e. it has a disjoint constraint and complete constraint. I believe that I have translated this correctly. The subtypes have the constraint of "**ON DELETE CASCADE**". The subtype discriminator is of type **ENUM**, such that the disjoint constraint is enforced. The subtypes themselves have different attributes, qualifying it to be valid a specialization hierarchy. Postgraduate Courses include the number of hours of teaching the students have to do for that course, whereas the Undergraduate courses contain an **Enum** stating if that course has a year abroad, or, a placement year.

I also believe that, in both the EERD diagram and the Logical Relational Database Schema, all the relations are in 3NF. So, I have created associative/bridge entities to map the M to M relationship between Student and Hobby, and Student and Society. The bridge entities also have the required constraint of "**ON DELETE CASCADE**".

In terms of constraints, all the primary keys, for the entities that I have created have the constraint of "NOT NULL"

The **Mem_Type** attribute is defined as an Enum, requiring the attribute to be only from the listed options (Member, Chair, Treasurer, or Secretary). I have also made use of **Enums** for subtype discriminator for the course specialisation hierarchy.

In Database setup file, I have populated all but the activities relation, with the required 5 minimum rows of data. I have not implemented the activities relation.

In the SQL Queries file, I have 4 instances where the JOIN statement uses 3 tables, this is more than the required minimum 1 instance of using 3 tables. Also, I have additional SQL Queries (more than 3). There are 2 subqueries (more than the required minimum of 1). There are two queries that use subqueries. There are 2 queries that use both an INNER JOIN between 3 tables and a subquery.

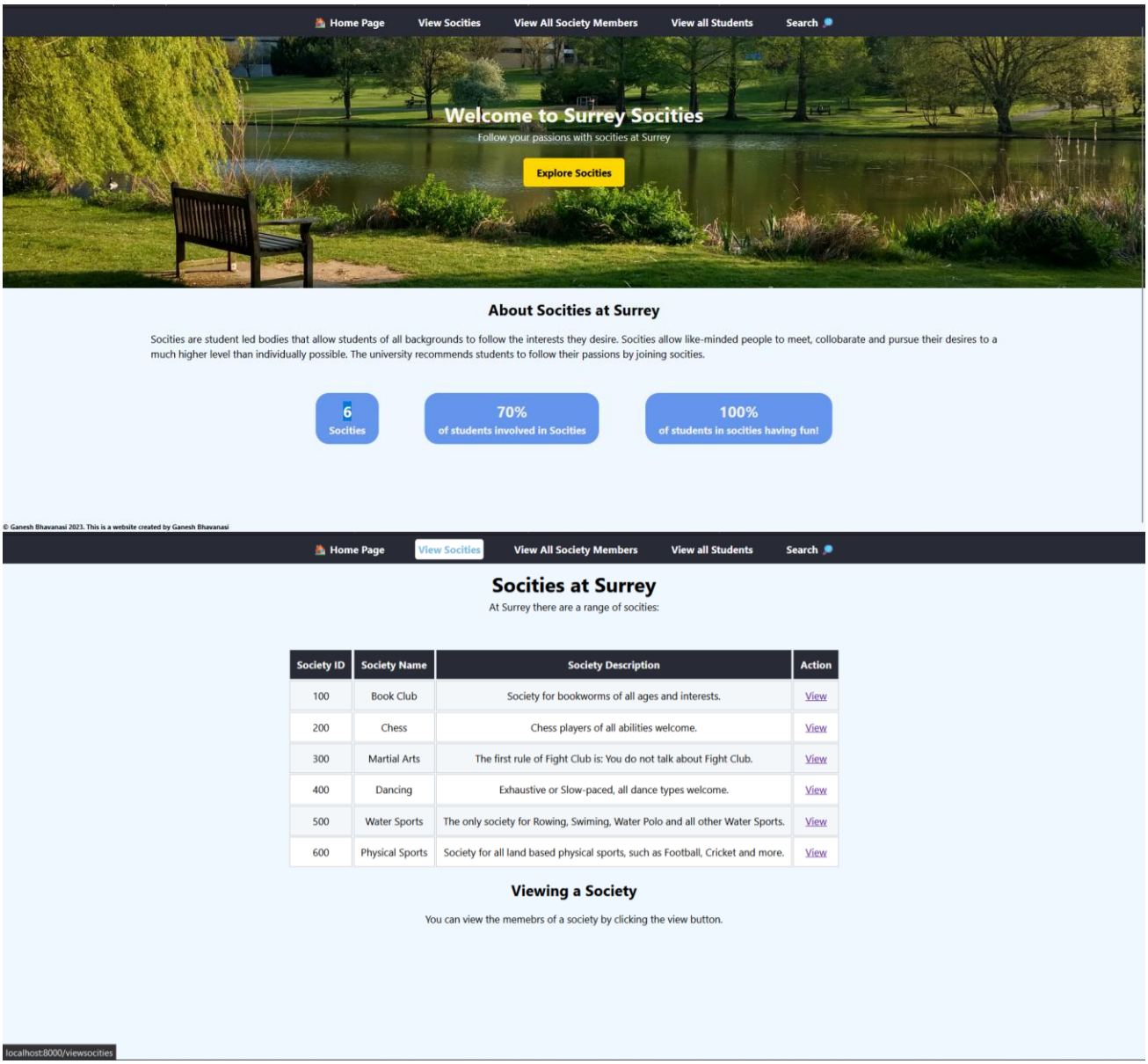
For the Advanced Taks in the Web development section, I have done numerous advanced tasks. I have added the functionality to create new rows of data (You are able to add a student to the database, from the website), there is NodeJS code to validate the inputs (e.g. checks length of phone number, age of student, the URN is 6 digits and not already used, and that the student type matches the course type (selected)), so that only a valid student is added. I have added the ability to delete rows of data (you are able to delete a student from the database), there is appropriate NodeJS code to delete rows of data. I have added the functionality to search for data, you are able to navigate to a search page, which allows you to search for data and allows you to specify which form of search you want to do. There is appropriate validation done to the search query (e.g. if they do a URN search, the query must be a URN), based on what form of search the user selects, and a results page is shown, showing the data relevant to the search.

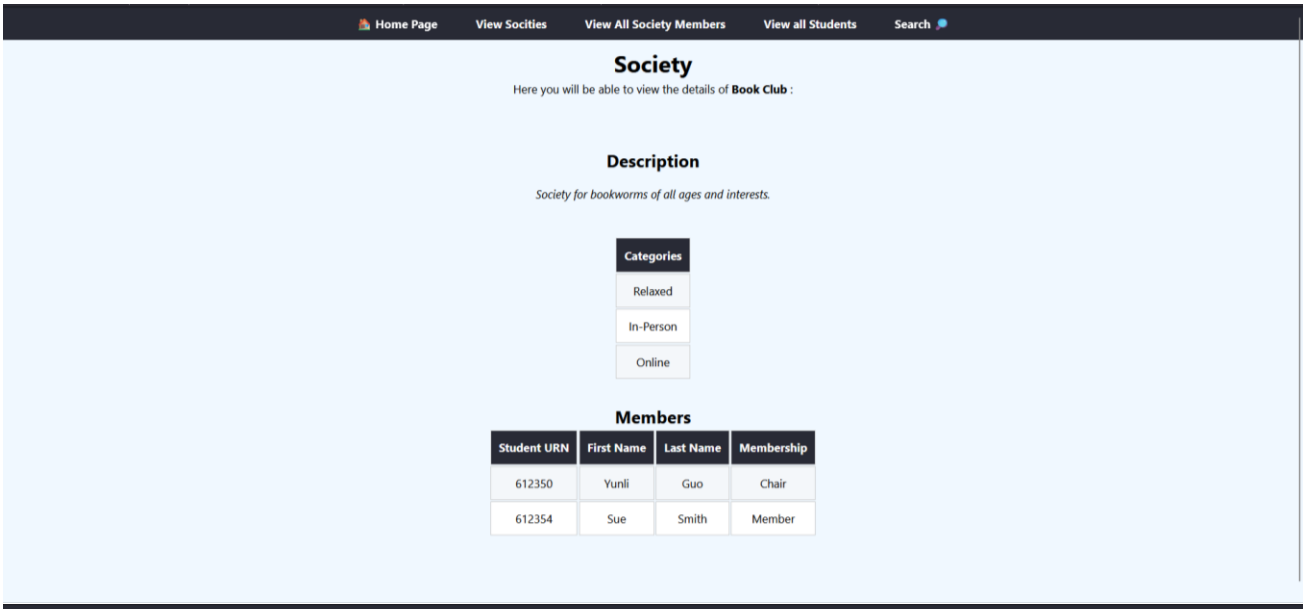
In addition to this I have made sure validation is done wherever it is required. For example, when editing a student, you are only shown course options which match their student type (i.e. for undergraduate students, you are only shown the courses that are undergraduate, likewise for postgraduate). There is appropriate validation in the JavaScript so that when creating a student, their date of birth is at least 18 years ago (i.e. only people above the age of 18 are added as students). I have made sure that when a student switches course, is deleted, or are added to the database, that the relevant courses have the enrolment numbers incremented/decremented appropriately.

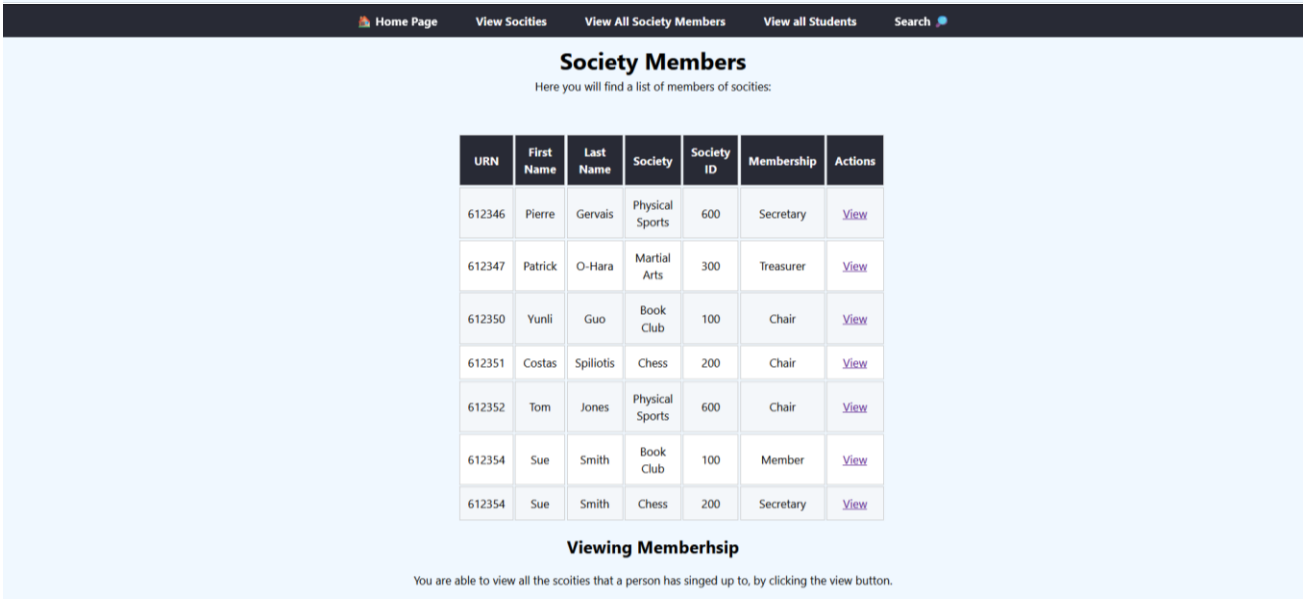
6 References

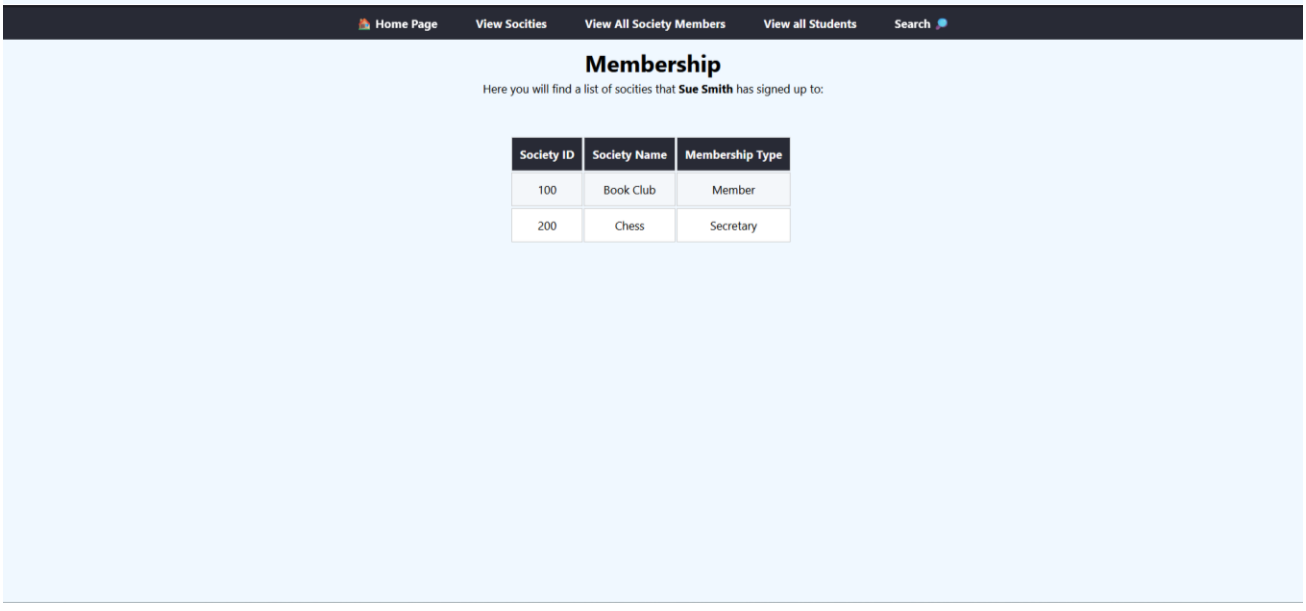
I have used the university website to add some courses in the course table (e.g. BMBS Medicine). Apart from that I have not used any websites to elicit business rules. I have not used any external libraries, or CSS templates within my website. I have used an image in my website, I got the image of a public domain website (https://unsplash.com/photos/green-grass-field-near-lake-under-blue-sky-during-daytime-ObqRu3VNDSE).

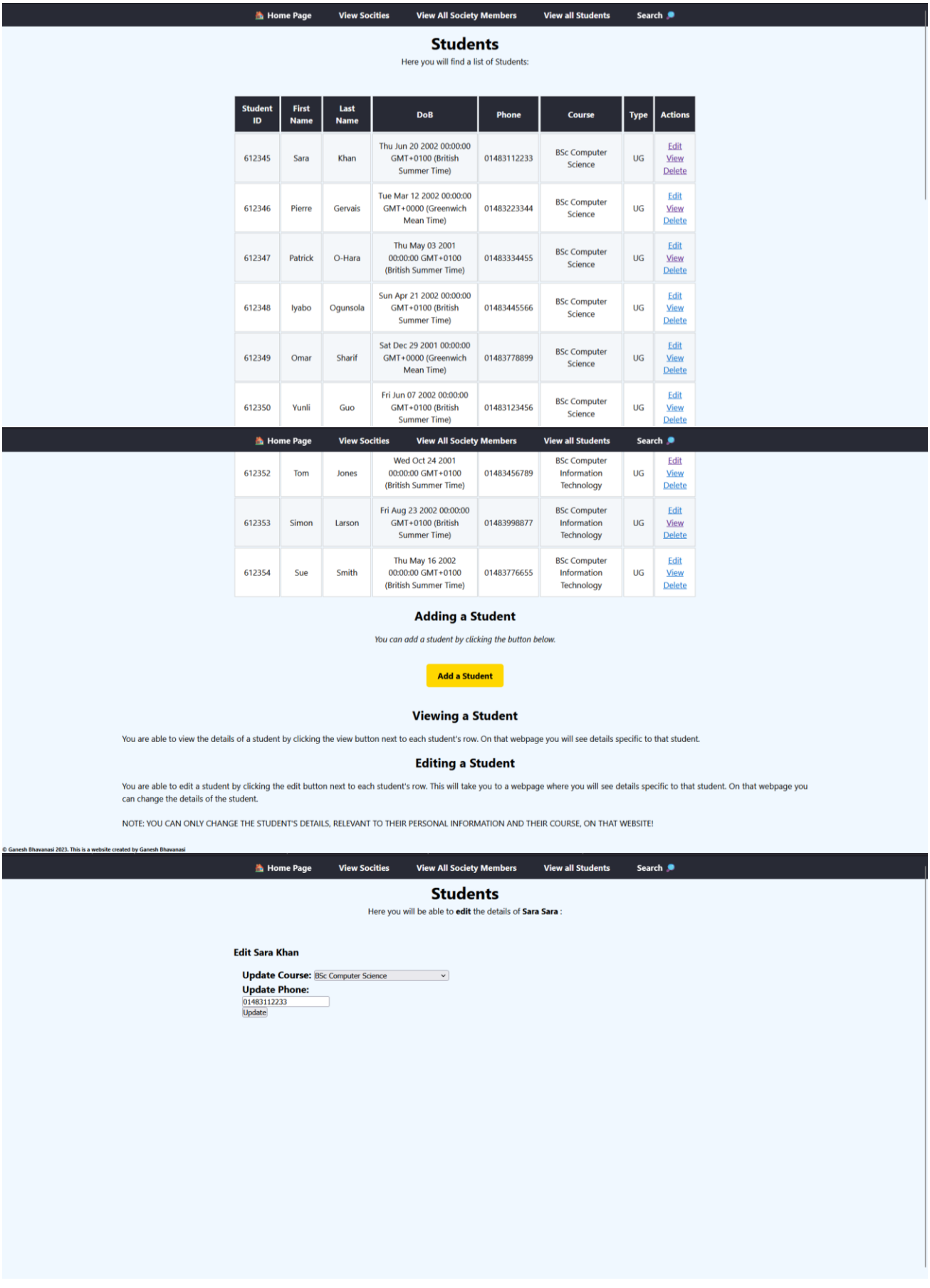
7 Appendix: Screenshots of Website

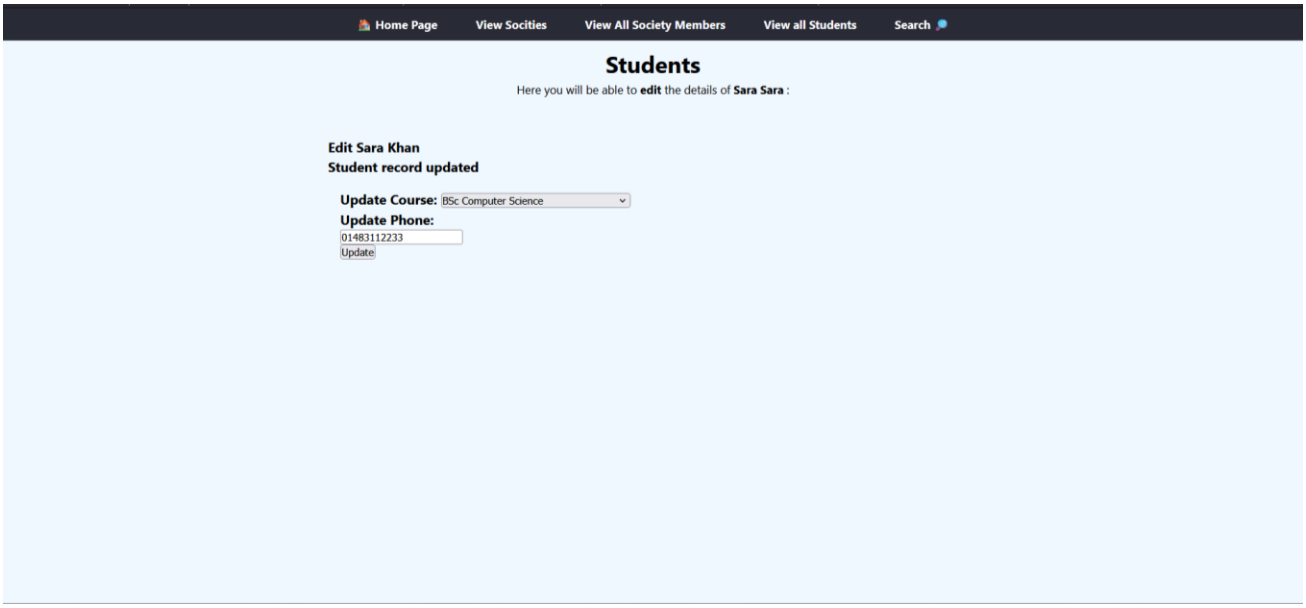


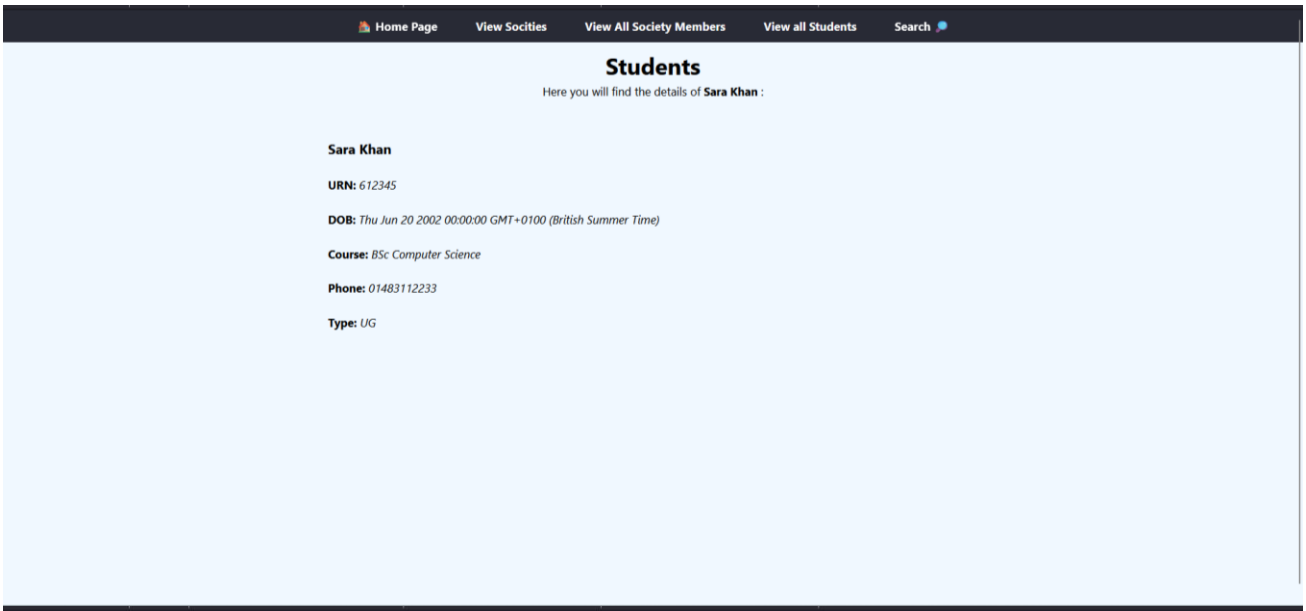

Home Page
View Socities
View All Society Members
View all Students
Search
Society
Here you will be able to view the details of Book Club :
Description
Society for bookworms of all ages and interests.
Categories
Relaxed
In-Person
Online
Members
Student URN
First Name
Last Name
Membership
612350
Yunli
Guo
Chair
612354
Sue
Smith
Member

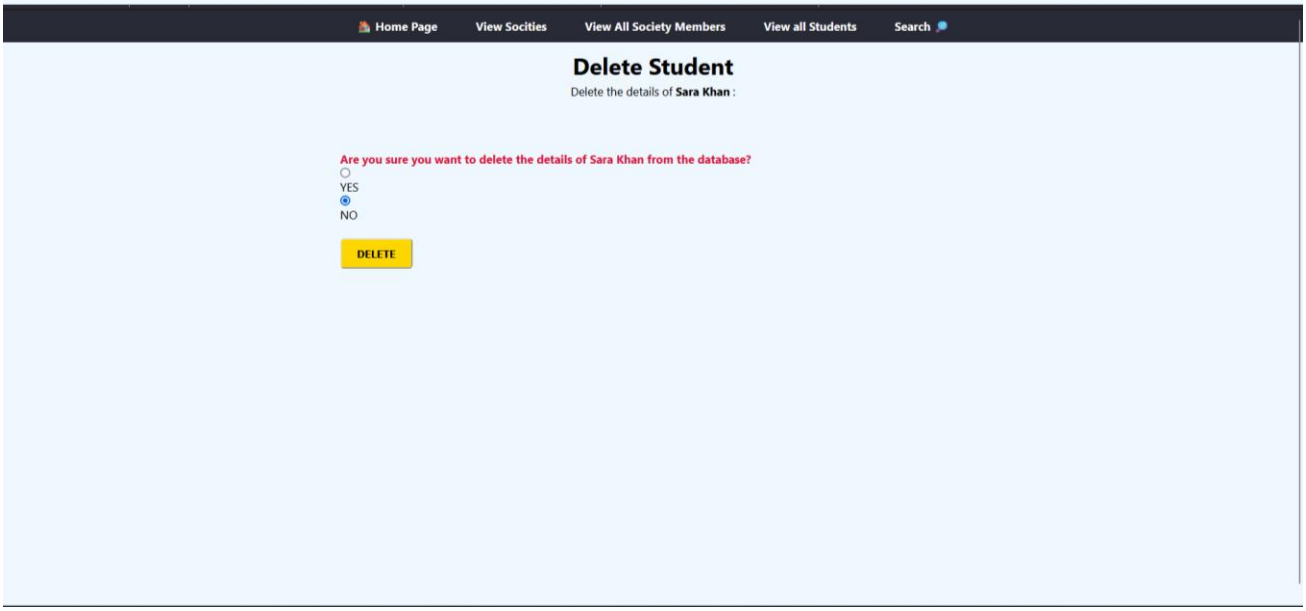

Home Page
View Socities
View All Society Members
View all Students
Search
Society Members
Here you will find a list of members of socities:
URN
First Name
Last Name
Society
Society ID
Membership
Actions
612346
Pierre
Gervais
Physical Sports
600
Secretary
View
612347
Patrick
O-Hara
Martial Arts
300
Treasurer
View
612350
Yunli
Guo
Book Club
100
Chair
View
612351
Costas
Spiliotis
Chess
200
Chair
View
612352
Tom
Jones
Physical Sports
600
Chair
View
612354
Sue
Smith
Book Club
100
Member
View
612354
Sue
Smith
Chess
200
Secretary
View
Viewing Memberhsip
You are able to view all the scoities that a person has singed up to, by clicking the view button.

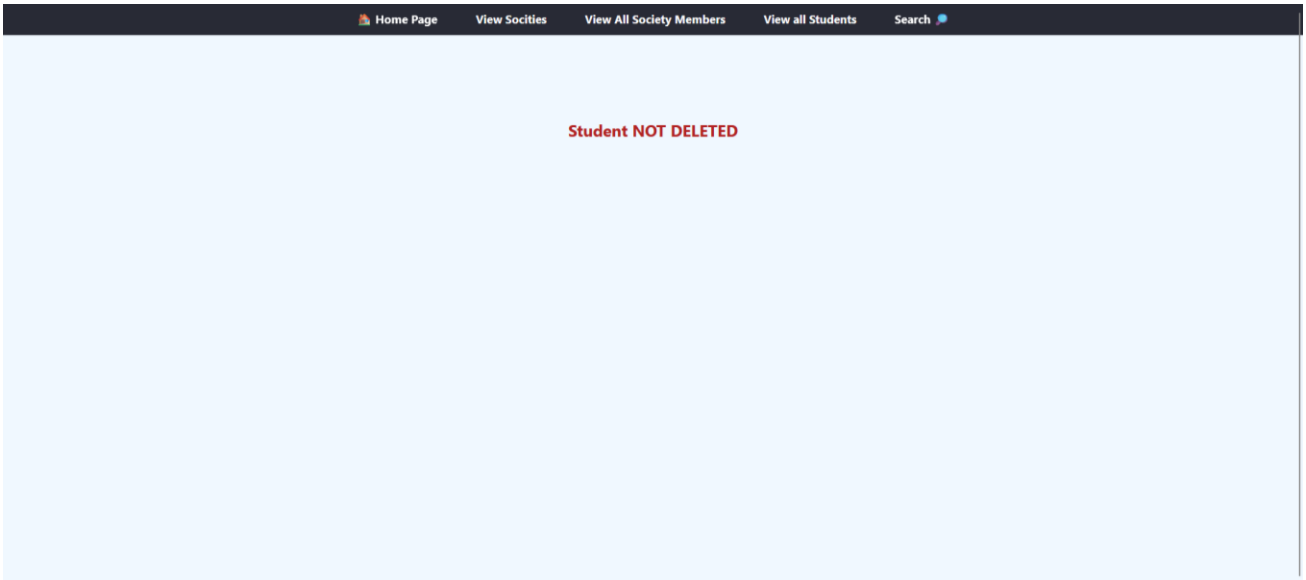

Home Page
View Socities
View All Society Members
View all Students
Search
Membership
Here you will find a list of socities that Sue Smith has signed up to:
Society ID
Society Name
Membership Type
100
Book Club
Member
200
Chess
Secretary

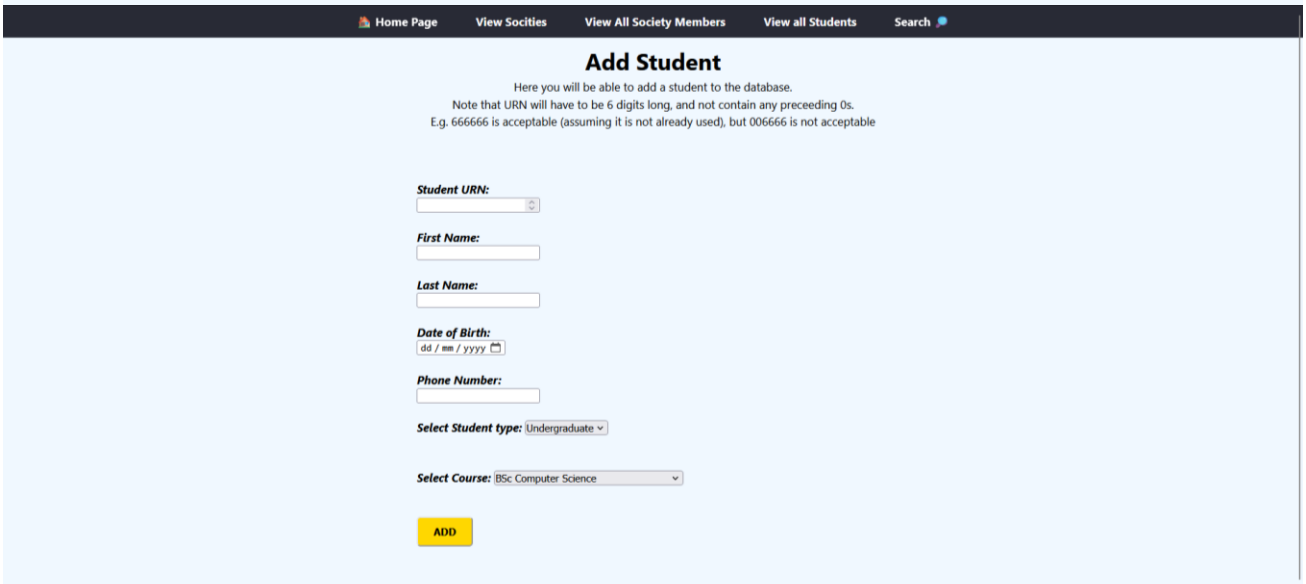

Home Page
View Socities
View All Society Members
View all Students
Search
Students
Here you will find a list of Students:
Student ID | First Name | Last Name | DoB | Phone | Course | Type | Actions
612345 | Sara | Khan | Thu Jun 20 2002 00:00:00 GMT+0100 (British Summer Time) | 01483112233 | BSc Computer Science | UG | Edit View Delete
612346 | Pierre | Gervais | Tue Mar 12 2002 00:00:00 GMT+0000 (Greenwich Mean Time) | 01483223344 | BSc Computer Science | UG | Edit View Delete
612347 | Patrick | O-Hara | Thu May 03 2001 00:00:00 GMT+0100 (British Summer Time) | 01483334455 | BSc Computer Science | UG | Edit View Delete
612348 | Iyabo | Ogunsola | Sun Apr 21 2002 00:00:00 GMT+0100 (British Summer Time) | 01483445566 | BSc Computer Science | UG | Edit View Delete
612349 | Omar | Sharif | Sat Dec 29 2001 00:00:00 GMT+0000 (Greenwich Mean Time) | 01483778899 | BSc Computer Science | UG | Edit View Delete
612350 | Yunli | Guo | Fri Jun 07 2002 00:00:00 GMT+0100 (British Summer Time) | 01483123456 | BSc Computer Science | UG | Edit View Delete
Home Page
View Socities
View All Society Members
View all Students
Search
612352 | Tom | Jones | Wed Oct 24 2001 00:00:00 GMT+0100 (British Summer Time) | 01483456789 | BSc Computer Information Technology | UG | Edit View Delete
612353 | Simon | Larson | Fri Aug 23 2002 00:00:00 GMT+0100 (British Summer Time) | 01483998877 | BSc Computer Information Technology | UG | Edit View Delete
612354 | Sue | Smith | Thu May 16 2002 00:00:00 GMT+0100 (British Summer Time) | 01483776655 | BSc Computer Information Technology | UG | Edit View Delete
Adding a Student
You can add a student by clicking the button below.
Add a Student
Viewing a Student
You are able to view the details of a student by clicking the view button next to each student's row. On that webpage you will see details specific to that student.
Editing a Student
You are able to edit a student by clicking the edit button next to each student's row. This will take you to a webpage where you will see details specific to that student. On that webpage you can change the details of the student.
NOTE: YOU CAN ONLY CHANGE THE STUDENT'S DETAILS, RELEVANT TO THEIR PERSONAL INFORMATION AND THEIR COURSE, ON THAT WEBSITE!
© Ganesh Bhavanasi 2023. This is a website created by Ganesh Bhavanasi
Home Page
View Socities
View All Society Members
View all Students
Search
Students
Here you will be able to edit the details of Sara Sara :
Edit Sara Khan
Update Course: BSc Computer Science
Update Phone:
01483112233
Update

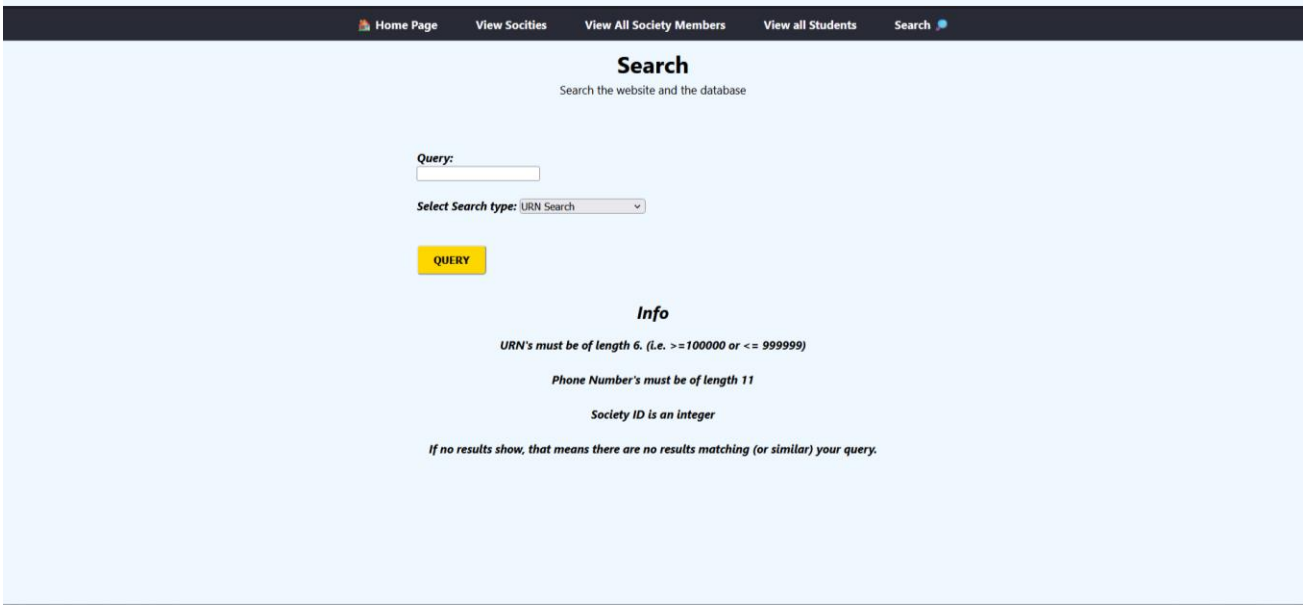

Home Page
View Socities
View All Society Members
View all Students
Search
Students
Here you will be able to edit the details of Sara Sara :
Edit Sara Khan
Student record updated
Update Course: BSc Computer Science
Update Phone:
01483112233
Update


Home Page
View Socities
View All Society Members
View all Students
Search
Students
Here you will find the details of Sara Khan :
Sara Khan
URN: 612345
DOB: Thu Jun 20 2002 00:00:00 GMT+0100 (British Summer Time)
Course: BSc Computer Science
Phone: 01483112233
Type: UG


Home Page
View Socities
View All Society Members
View all Students
Search
Delete Student
Delete the details of Sara Khan :
Are you sure you want to delete the details of Sara Khan from the database?
YES
NO
DELETE


Home Page
View Socities
View All Society Members
View all Students
Search
Student NOT DELETED


Home Page
View Socities
View All Society Members
View all Students
Search
Add Student
Here you will be able to add a student to the database.
Note that URN will have to be 6 digits long, and not contain any preceeding 0s.
E.g. 666666 is acceptable (assuming it is not already used), but 006666 is not acceptable
Student URN:
First Name:
Last Name:
Date of Birth:
dd / mm / yyyy
Phone Number:
Select Student type: Undergraduate
Select Course: BSc Computer Science
ADD


Home Page
View Socities
View All Society Members
View all Students
Search
Search
Search the website and the database
Query:
Select Search type: URN Search
QUERY
Info
URN's must be of length 6. (i.e. >=100000 or <= 999999)
Phone Number's must be of length 11
Society ID is an integer
If no results show, that means there are no results matching (or similar) your query.