Web Programming Languages

CS 6314.001

Project on:

Online Course Registration

Submitted By:

Talal Muneer – cxm180004

Chirag Shahi – cxs180005

Kanika Rawat - kxr180008

EASY ENROLL

- **Name for the Website:** Easy Enroll

- **Project Description:**

  This online course registration website is created for students in any University to enroll in courses. Generally, when we want to enroll in a course, we would sometimes search for the course, would like to know course syllabus and prerequisite to see if we are allowed to take it, check available seats in the course and instructor who is teaching that course. All these functionalities are given with online course registration website. It makes for a seamless, easy to use experience for the students as well as the admins of the website.

  - Sign up:

    The students who are visiting the website for the first time need to create an account so that their credentials get saved in the database. While signing up, if any of the field is left empty, user is notified to fill that particular field. Password is saved as a hash value using bcrypt.

  - Admin Login:
    The admin of the portal can modify the list of courses being offered, seats in a course, location, time and other values of the 'Course' data object.

  - User Login:

    Once the user registers into the system and logs into it, he/she comes on their homepage. There the user can see the current courses taken this semester. Past courses that the student has taken. Student can also see his/her personal information like name, address, email, phone number etc. on the homepage.

    - User can also go in view courses tab and view the available courses being offered this semester and next semester. User can also view available seats, total seats and other course related information
    - In 'Add Courses' navbar item, a user can add courses that he/she would like to their shopping cart.
    - In 'Drop courses' navbar item, a user can go and select the course to drop from the current list of courses that the user has taken

- ## Addition Features
  - Checking for prerequisites for a course and denying the request to register if it isn't taken before
  - Changing the status of a course to 'Closed' automatically whenever all the seats are filled
  - Not allowing a student to register for a course which is not being offered in the next semester
  - Limiting number of credits a student can take in a particular semester

- ## Database Design

We used MongoDB for our database operations. Below are examples of each collection documents used followed by their description:

Course Collection


```
{
   "_id" : ObjectId("5cb3a72e01f892abd8c2a200"),
   "name" : "Database Design",
   "code" : "CS6360",
   "section" : "1",
   "instructor" : "Murat Kantarcioglu",
   "prereq" : "CS5343",
   "dept" : "CS",
   "coursedescription" : " Methods, principles, and concepts that are relevant to the practice of database software design. Database system architecture; conceptual database models; relational and object-oriented databases; database system implementation; query processing and optimization; transaction processing concepts, concurrency, and recovery; security.",
   "totalseats" : 44,
   "enrolled" : 3,
   "availableseats" : 41,
   "location" : "ECSN 2.120",
   "starttime" : "10:30:00",
   "endtime" : "11:45:00",
   "classstatus" : "open",
   "credits" : 3,
   "courseoffering" : "Fall",
   "days" : [
      "Monday",
      "Wednesday"
   ]
}
```

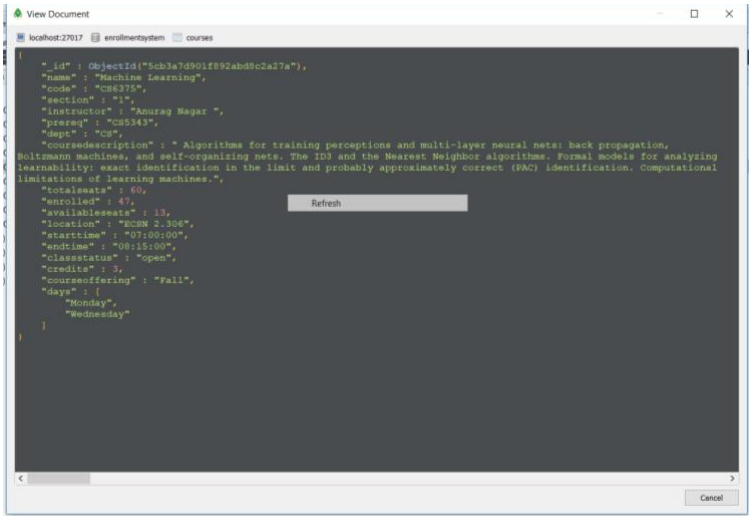
The above data model contains information regarding the courses being offered/ past courses. This model contains the number of seats in a course, vacant seats, location, instructor, time and other necessary information that aids a student to choose a course. It also has prerequisites which are used in our website to make sure that they are completed before the current subject.

User collection

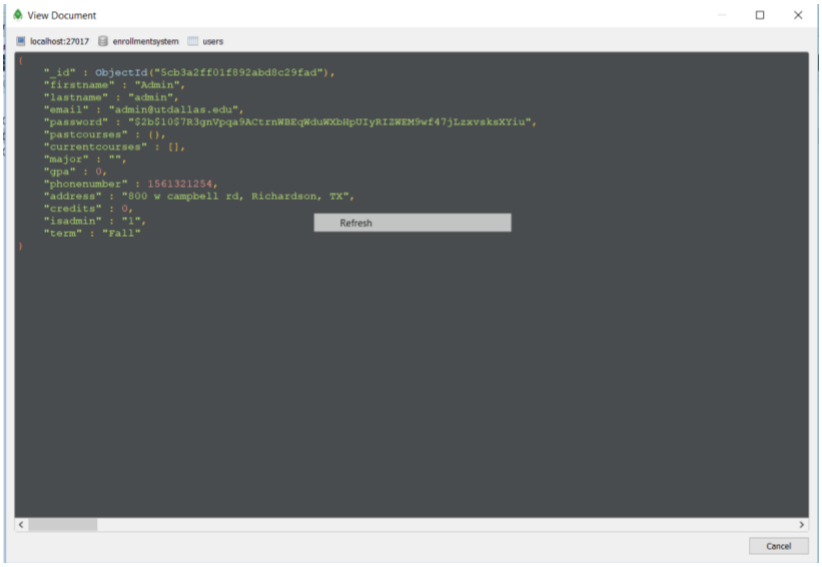

```
{
   "_id" : ObjectId("5cb3a2ff01f892abd8c29fad"),
   "firstname" : "Admin",
   "lastname" : "admin",
   "email" : "admin@utdallas.edu",
   "password" : "$2b$10$7R3gnVpqa9ACtrnWBEqWduWXbHpUIyRIZWEM9wf47jLzxvsksXYiu",
   "pastcourses" : {},
   "currentcourses" : [],
   "major" : "",
   "gpa" : 0,
   "phonenumber" : 1561321254,
   "address" : "800 w campbell rd, Richardson, TX",
   "credits" : 0,
   "isadmin" : "1",
   "term" : "Fall"
}
```


The above data model contains information regarding the users of the website i.e. the admin/instructor and the students. It contains information regarding the user which include his/her name, email, an array of past and current courses, admin flag and other credentials. It also contains a password field which is constructed using *bcrypt* module which hashes a password based on *Blowfish* cipher. This makes the password resistant to brute-force attacks even with increasing computation power.

Course Collection Object



User collection Object



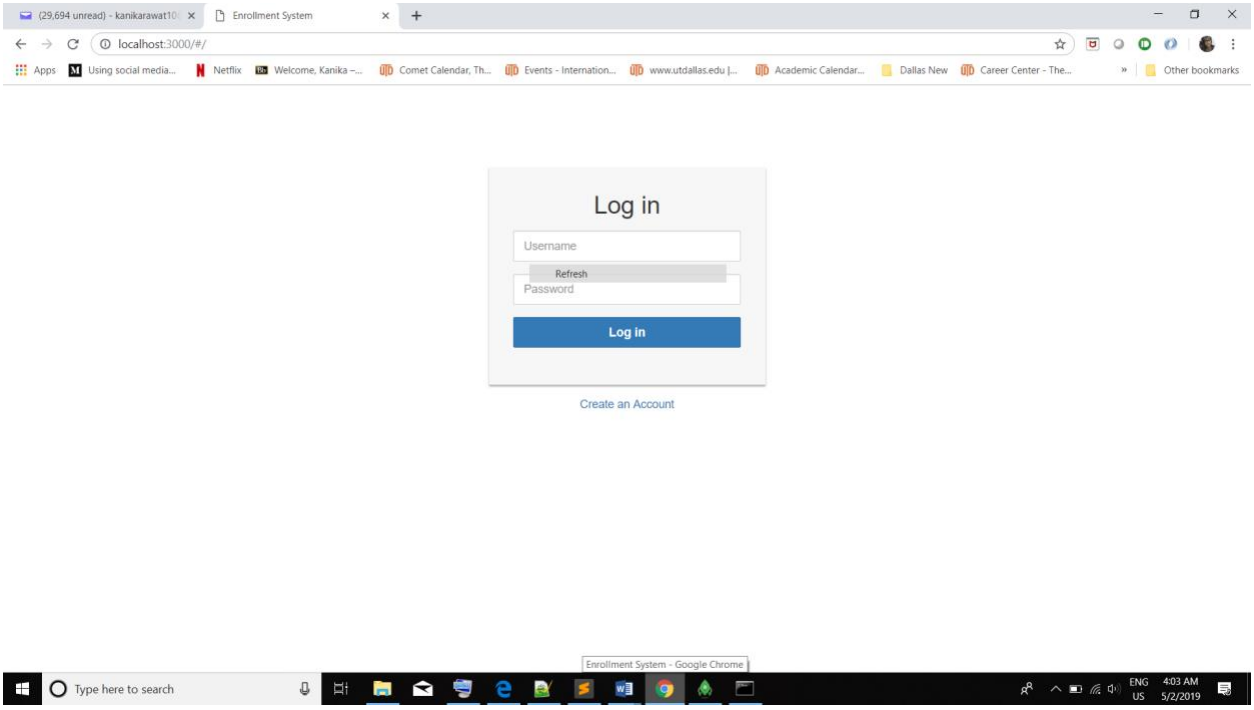
- **Languages/Frameworks used:**
  - MEAN Stack:
    - Database: MongoDB
    - Framework to support Node: ExpressJS
    - Front end framework: AngularJS
    - Back end framework: NodeJS

  - Dependencies installed for ExpressJS:
    - As used in Assignment 5:
      - cookie-parser
      - debug
      - express
      - http-errors
      - jade
      - monk
      - morgan
    - New:
      - angular-cookies
      - bcrypt
      - passport
      - passport-local

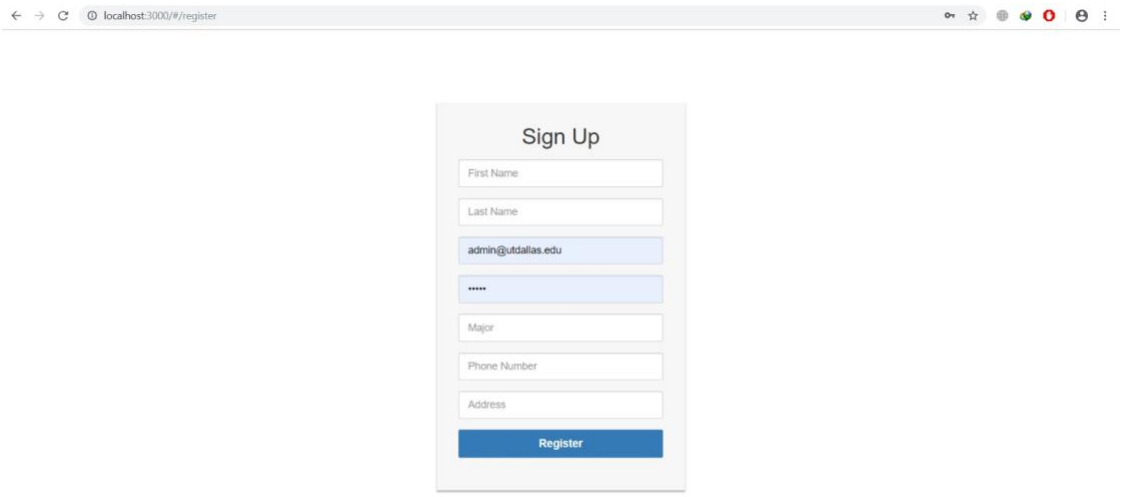
- **Screenshots of main functionalities:**

The main functionalities of the project are:

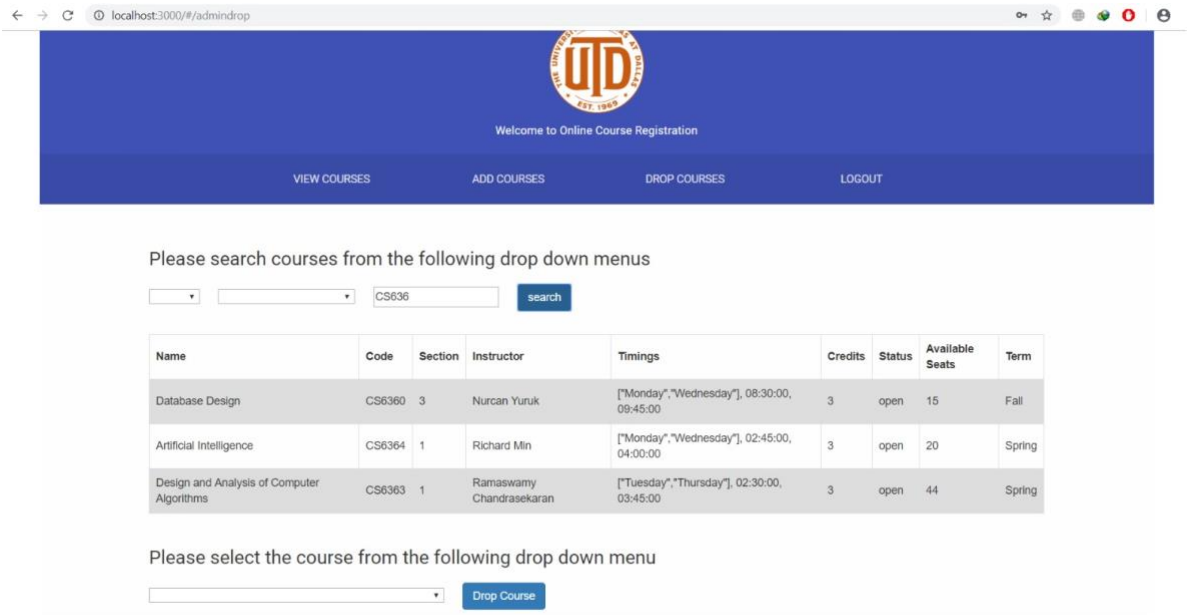
1) Login page: A student or an admin can login via their credentials.



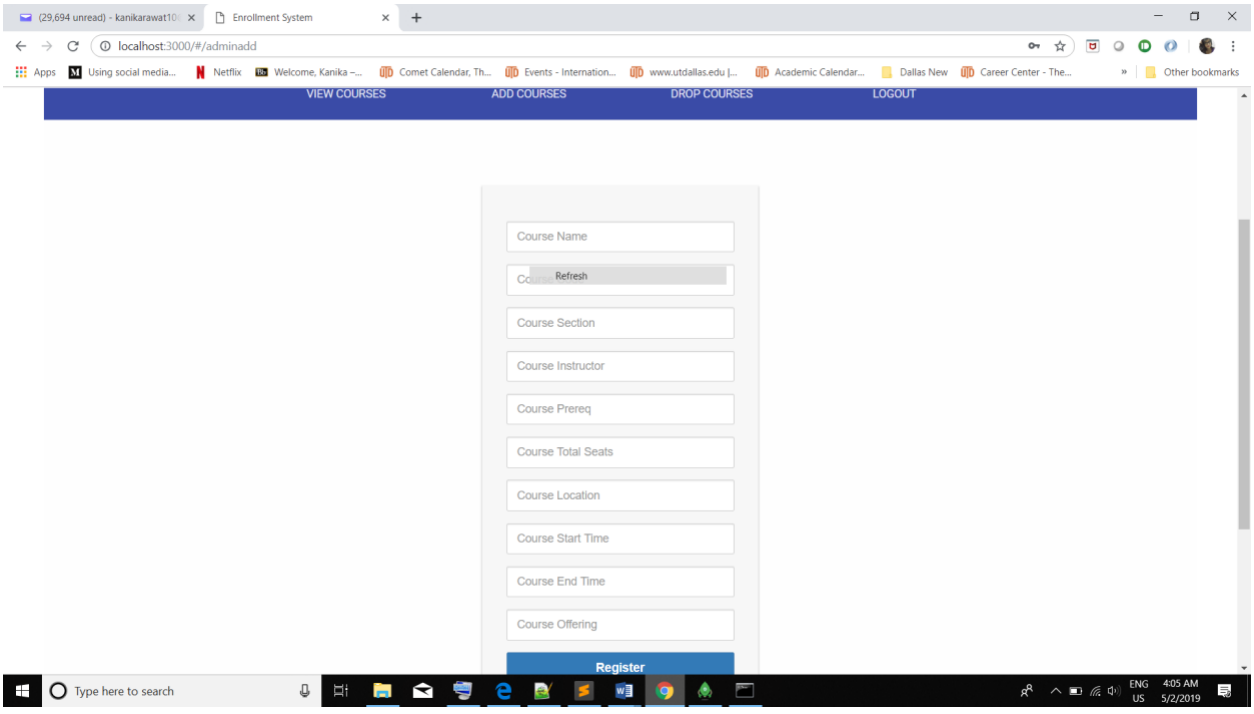
2) Sign Up Page: A student can use this to create an account before signing in.



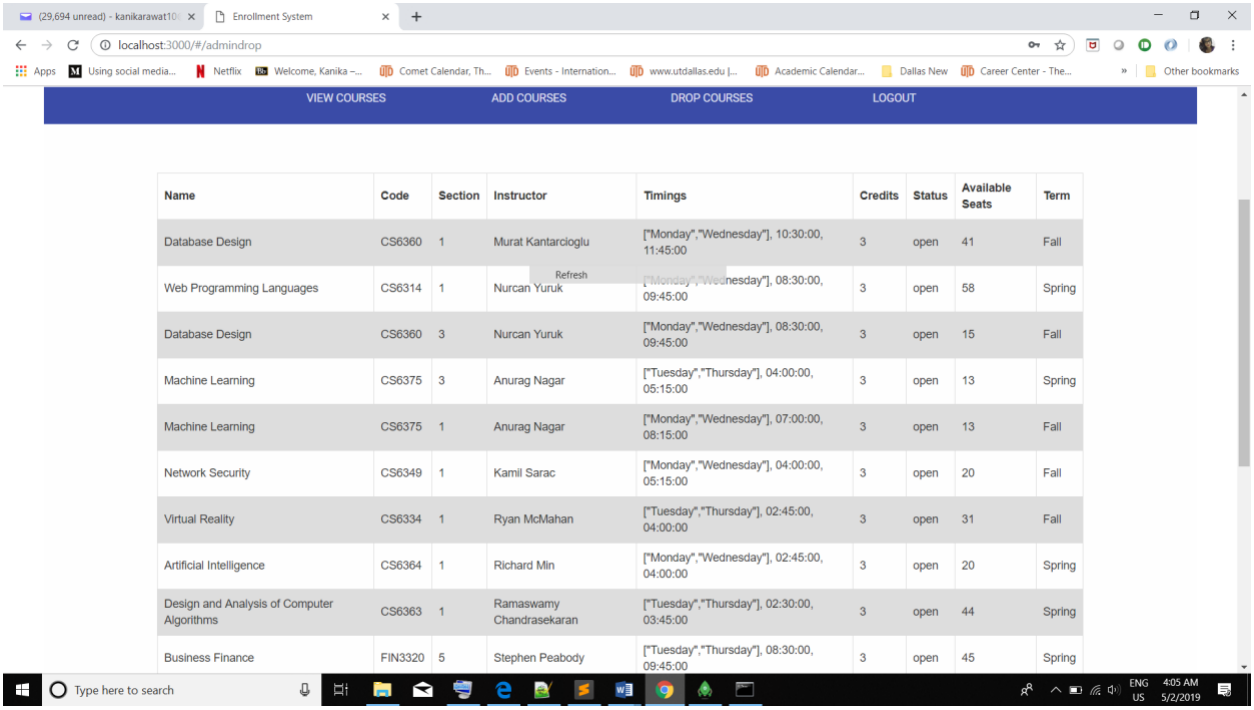
3) Search and filter courses



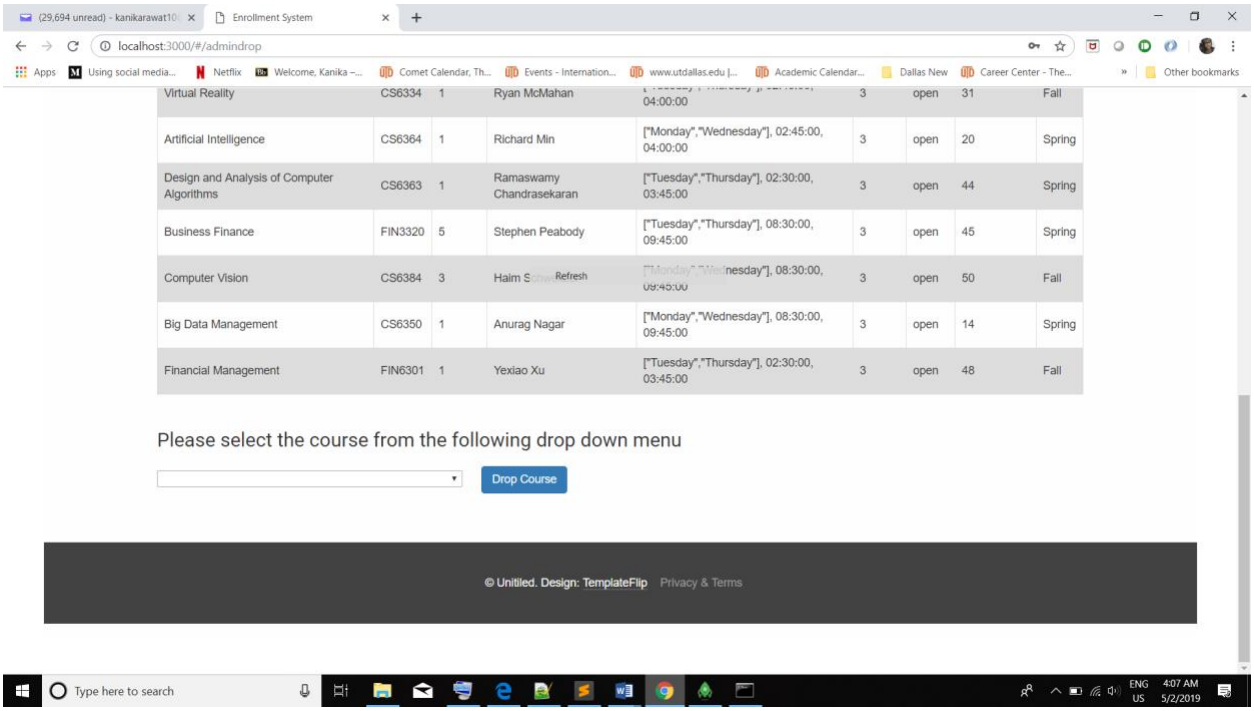
4) Admin can add a course to the portal via the ‘adminadd’ page



5) Admin can delete a course from the portal via the ‘admindrop’ page.



Name	Code	Section	Instructor	Timings	Credits	Status	Available Seats	Term
Database Design	CS6360	1	Murat Kantarcioglu	["Monday","Wednesday"], 10:30:00, 11:45:00	3	open	41	Fall
Web Programming Languages	CS6314	1	Nurcan Yuruk	["Monday","Wednesday"], 08:30:00, 09:45:00	3	open	58	Spring
Database Design	CS6360	3	Nurcan Yuruk	["Monday","Wednesday"], 08:30:00, 09:45:00	3	open	15	Fall
Machine Learning	CS6375	3	Anurag Nagar	["Tuesday","Thursday"], 04:00:00, 05:15:00	3	open	13	Spring
Machine Learning	CS6375	1	Anurag Nagar	["Monday","Wednesday"], 07:00:00, 08:15:00	3	open	13	Fall
Network Security	CS6349	1	Kamil Sarac	["Monday","Wednesday"], 04:00:00, 05:15:00	3	open	20	Fall
Virtual Reality	CS6334	1	Ryan McMahan	["Tuesday","Thursday"], 02:45:00, 04:00:00	3	open	31	Fall
Artificial Intelligence	CS6364	1	Richard Min	["Monday","Wednesday"], 02:45:00, 04:00:00	3	open	20	Spring
Design and Analysis of Computer Algorithms	CS6363	1	Ramaswamy Chandrasekaran	["Tuesday","Thursday"], 02:30:00, 03:45:00	3	open	44	Spring
Business Finance	FIN3320	5	Stephen Peabody	["Tuesday","Thursday"], 08:30:00, 09:45:00	3	open	45	Spring



Virtual Reality	CS6334	1	Ryan McMahan	["Tuesday","Thursday"], 02:45:00, 04:00:00	3	open	31	Fall
Artificial Intelligence	CS6364	1	Richard Min	["Monday","Wednesday"], 02:45:00, 04:00:00	3	open	20	Spring
Design and Analysis of Computer Algorithms	CS6363	1	Ramaswamy Chandrasekaran	["Tuesday","Thursday"], 02:30:00, 03:45:00	3	open	44	Spring
Business Finance	FIN3320	5	Stephen Peabody	["Tuesday","Thursday"], 08:30:00, 09:45:00	3	open	45	Spring
Computer Vision	CS6384	3	Haim S	["Monday","Wednesday"], 08:30:00, 09:45:00	3	open	50	Fall
Big Data Management	CS6350	1	Anurag Nagar	["Monday","Wednesday"], 08:30:00, 09:45:00	3	open	14	Spring
Financial Management	FIN6301	1	Yexiao Xu	["Tuesday","Thursday"], 02:30:00, 03:45:00	3	open	48	Fall

- **Work division among team members:**

1) Talal Muneer:
    a. Login pages logic and design
    b. Admin functionalities – add, drop, edit and update status of course
    c. Home page design and navigation bar implementation
    d. No vacant seats in a class logic implementation

2) Chirag Shahi:
    a. Password hashing using bcrypt and storing in the database
    b. Cart functionalities – add, remove and final registration of a course
    c. Creation of database objects
    d. Filter logic implementation

3) Kanika Rawat:
    a. Form validation logic implementation
    b. Prerequisites logic implementation
    c. Retrieving relevant courses from the database
    d. Search logic implementation
    e. Logout logic

All the team members worked together to brainstorm the structure of the database used. Everyone contributed to filling it with test data and on making this report.