Project Specification: 1

ONLINE BAKERY

There should be 2 types of user level: Administration and user.

The inventory module should consist of the following fields:

- Item Number
- Items [Snacks/Pastry/Chocolates& Dessert/Cheese & Dairy]
- Price
- Quantity
- Mfg Date & Exp date

The access is given to Administrator alone. An Administrator should be able to

- Create, Modify and Delete Foods
- Search on Food Number/Food Name

The Sales module should consist of the following fields:

- Date
- Item Number
- Food Name
- Quantity
- Price
- Total Cost

This module is accessible by Administrator and User. The Administrator should be able to

- Create, Modify and Delete items

The User should be able to

- Create, Modify items

At the end of the day,

- The sum of quantities of a particular Food in the sales module should be tallied with the Inventory module and report should be generated on the current stock of Foods.
- The total cost in the sales module should be tallied with sum of the prices of individual Foods sold.

TEAM MEMBERS:

Project Specification: 2

EDUCATION SYSTEM

Design and Develop an education help system

- It should provide the details on Graduate/Diploma courses offered across India after completing the school.

There should be 1 type of user: **Administrator**.

The Course module should consist of the following fields:

- Course Number
- Course Name
- Course Duration
- Course Mode (Fulltime/Part-time/Distance/Online)
- Course Type (Degree/Diploma/Certification)
- College Name (It could be more than one)
- Location
- Contact Details

Instructions:

- Search should be possible with **Course Number, Course Name, College Name and Location.**
- The Administrator should be able to **create, update, delete the entries**.

The Subscriber module should consist of the following fields:

- Subscriber Number
- Subscriber Name
- Subscriber's email id
- Subscriber status (Active/Closed)
- Newsletter status (Sent/Returned)

Instructions:

- Search should be possible with **Subscriber Number, Subscriber Name, Subscriber Status and Newsletter status.**
- The Administrator should be able to **create, update, delete the entries.**

The system should be able to generate the following report at the end of the month:

- List of subscribers with Active status

TEAM MEMBERS:

Project Specification: 3

GAMING SYSTEM.

It should teach Computer games in a simple way with real world examples.

There should be 1 type of user. Administrator.

- The **Gaming module** should consist of the following fields:

1. Game Number
2. Game Name
3. Game Details
4. Application (Real world examples – Examples:JetPack.IcyTower,Mario,Angrybirds and so on)

- Search should be possible with Game Number, Game Name and Application.

The Administrator should be able to create, update, and delete the games.

The **Subscriber module** should consist of the following fields:

1. Subscriber Number
2. Subscriber Name
3. Subscriber’s email id
4. Subscriber status (Active/Closed)
5. Newsletter status (Sent/Returned)
6. Search should be possible with Subscriber Number, Subscriber Name, Subscriber Status and Newsletter status.
7. The Administrator should be able to create, update, and delete the entries.

The system should be able to generate the following report at the end of the month:

1. List of New topics included in comparison with the previous month.
2. List of Newsletter status with Returned status (The subscriber status will be ‘Closed’ after the eNewsletter mail is bounced twice from the given email id.)

TEAM MEMBERS:

Project Specification: 4

SCAN CENTRE

There are 2 types of user: Doctor and Administrative officer
The system should maintain a doctor register and a patient register. User should be able to perform following operations:
For managing doctors: (Access is given for Administrative officers only)

- Add a new doctor.
  - Each doctor should have the following details:

        Name
        Title (Master Health Check up, Hepatitis B & C tests, Allergy Testing, Osteoporosis scan)
        Gender
        Date of Birth
        Qualification
        Medical License No.
        Center Id
        Address
        Contact number
        Email id

  - Search on Medical License No./ Center Id
  - Search on Name/Contact number
  - Modify the doctor detail
  - Add a new title

For managing patients: (Access is given for Doctor and Administrative officers)

- Add a new patient.
  - Each patient should have the following details:

        Registration Number
        Registration Date
        Name
        Gender
        Date of Birth
        Address
        Contact phone number
        Hospital Referred by
        Test Date
        Test Type (X-ray/ECG/CT Scan/MRI Scan/Cardiac CT)
        Doctor’s Name:
        Fees

  - Add more tests for the same patient.
  - Modify patient details.
  - Search on Registration number.
  - Search on patient name/phone number.

Add any other data/info for doctors & patients if required. Store the doctor & patient data in a file. When the system is started, it should fetch the patient data from this flat file. At the end, the system should update the modified data in the same flat file.

TEAM MEMBERS:

GOWTHAM RAGAV

Project Specification: 5

E-STATIONERY

The e-Stationery example is a simple e-commerce application where users can select Stationery, view the Stationery Items catalog, and purchase Stationery Items.

- Implement a module for user registration. Collect the name, contact number, shipping and billing address from the user while registering and any additional fields if required.
- Stationery Items should be organized into various categories. Display all these categories in the home page. Selecting a category should display all the Stationery Items under the category.
- A guest (who is not yet logged in) should be able to query for any Stationery Items. To buy Stationery items the user should be logged in. Implementing a shopping cart is optional.

There should be a Buy Stationery button under each stationery. This should take the user to the buy Stationery page. Also, display the count of number of available stationery items.

If zero, display appropriate message, otherwise, the customer information should be auto-populated in the buy page. However, the user should be able to edit any information.

The user should be able to select the number of stationery items required. The total price should be displayed to the user. By clicking on 'Submit Order', display success message.

- Two kinds of roles exist for the system: A regular user (end user) and an admin (or a super user). A regular user should only be able to buy stationery. However an admin can additionally add stationery to the inventory and query all the orders placed in the website.
- Maintain a log of all the stationery items sold including the details of the user who bought it. An user with the role of 'Admin'/'SuperUser' should be able to query all the orders placed in the website.

Display appropriate messages wherever applicable.

As a high level table structure, the following tables will be required:

Stationery items, User, Role, Inventory Tables,

The Stationery entity should define at-least the following attributes:

- Code Number
- Stationery name
- The price
- Whether the stationery items is on sale
- A description of the stationery items

TEAM MEMBERS:

Project Specification: 6

EXPENSE MANAGER

Expense Manager is an online application where the user can login to maintain his expenses and incomes.

The user should be able to -

- Tracking expenses and incomes by week, month and year as well as by categories
- Have Multiple accounts
- Schedule the payments.
- Handle recurring payments automatically
- Upload a picture of receipt
- Budget
- Search and generate reports (preferably in a .txt or .xls file format)
- Customize expense categories, payer/payer, payment methods, etc

Some of the sample categories to be included

- Automobile
  - Fuel
  - Maintenance
- Entertainment
  - Movies
  - Party
  - Concert
  - Sports
- Family
  - Child Care
  - Toys
  - Others
- Food
  - Breakfast
  - Lunch
  - Dinner
  - Snacks
  - Groceries
- Insurance
  - Auto
  - Health
  - Life
- Tax
  - Property Tax
  - Vehicle Tax
  - Other
- Travel
  - Airplane
  - Bus
  - Train
  - Food
  - Hotel
  - Taxi
  - Other Transportation
  - Misc
- Utilities
  - Water
  - Cable TV
  - Electrical
  - Gas
  - Internet
  - Telephone

TEAM MEMBERS:

Project Specification: 7

MATHS FUN

Design and Develop a Mathematics help system. It should teach Math topics in a simple way with real world examples.

There should be 1 type of user. Administrator.

- The **Math module** should consist of the following fields:

1. Topic Number
2. Topic Name
3. Problem Details (There will be more than one maths problem for any topic)
4. Answer
5. Answer Details (Simplified explanation of arriving at the answer)
6. Application (Real world examples – Examples: Grocery, Construction, Robotics and so on)

- Search should be possible with Topic Number, Topic Name and Application.

The Administrator should be able to create, update, delete the entries.

The **Subscriber module** should consist of the following fields:

1. Subscriber Number
2. Subscriber Name
3. Subscriber's email id
4. Subscriber status (Active/Closed)
5. Newsletter status (Sent/Returned)
6. Search should be possible with Subscriber Number, Subscriber Name, Subscriber Status and Newsletter status.
7. The Administrator should be able to create, update, delete the entries.

The system should be able to generate the following report at the end of the month:

1. List of New topics included in comparison with the previous month.
2. List of Newsletter status with Returned status (The subscriber status will be 'Closed' after the eNewsletter mail is bounced twice from the given email id.)

TEAM MEMBERS:

Project Specification: 8

iPad System

Design and Develop a iPad System.
There should be 2 types of user. Administrator and User.
The subscriber module will consist of

- Subscriber Number * Name
- Mobile Number * Product Type [Prepaid/Postpaid]
- PAN No./Credit card No. * Permanent address
- Corresponding address * State/Circle
- Email id * Scheme/Plan
- Facilities required [SMS/STD/International/Conferencing/Forwarding]
- Validity * Base Tariff
- Bill receiving mode [Post/SMS/eMail]
- Payment mode [Cash/DD/Online]
- Status

The Subscriber module will be accessible by Administrator.
The Administrator will be able to

- Create, Modify, Delete subscribers
- Search on Name/Mobile Number/Subscriber Number

The payment module will consist of

- Name * Subscriber Number
- Mobile Number * Product Type[Prepaid/Postpaid]
- Bill Number * Billing Date
- Recharge/Payment * Due Date
- Date of payment * Amount paid
- Amount due * Transaction Status

The payment module will be accessible by Administrator and User.
The Administrator will be able to

- Create, Modify and Delete payments
- Search on Bill Number/Mobile number/Subscriber Number

The user will be able to

- Create, Modify payments
- Search on Bill Number/Mobile Number/Subscriber Number

The system should be able to generate the following reports at the end of the month:

- Subscriber list with their plan details
- List of subscribers with outstanding bills

TEAM MEMBERS:

Project Specification: 9

CUPCAKE COMPANY

Design and Develop an Online Bakery

There should be 2 types of user level: Administration and user.

The inventory module should consist of the following fields:

- Item Number
- Items [Snacks/Pastry/Chocolates& Dessert/Cheese & Dairy]
- Price
- Quantity
- Mfg Date & Exp date

The access is given to Administrator alone. An Administrator should be able to

- Create, Modify and Delete Foods
- Search on Food Number/Food Name

The Sales module should consist of

- Date
- Item Number
- Food Name
- Quantity
- Price
- Total Cost

This module is accessible by Administrator and User.

The Administrator should be able to

- Create, Modify and Delete items

The User should be able to

- Create, Modify items

At the end of the day,

- The sum of quantities of a particular Food in the sales module should be tallied with the Inventory module and report should be generated on the current stock of Foods.
- The total cost in the sales module should be tallied with sum of the prices of individual Foods sold.

TEAM MEMBERS:

Project Specification: 10

MOBILE PHONE MANAGEMENT SYSTEM

Design and Develop a Mobile phone Management System.
There should be 2 types of user. Administrator and Salesperson.
The subscriber module will consist of

- Subscriber Number * Name
- Mobile Number * Product Type [Prepaid/Postpaid]
- PAN No./Credit card No. * Permanent address
- Corresponding address * State/Circle
- Email id * Scheme/Plan
- Facilities required [SMS/STD/International/Conferencing/Forwarding]
- Validity * Base Tariff
- Bill receiving mode [Post/SMS/eMail]
- Payment mode [Cash/DD/Online]
- Status

The Subscriber module will be accessible by Administrator.
The Administrator will be able to

- Create, Modify, Delete subscribers
- Search on Name/Mobile Number/Subscriber Number

The payment module will consist of

- Name * Subscriber Number
- Mobile Number * Product Type[Prepaid/Postpaid]
- Bill Number * Billing Date
- Recharge/Payment * Due Date
- Date of payment * Amount paid
- Amount due * Transaction Status

The payment module will be accessible by Administrator and Salesperson.
The Administrator will be able to

- Create, Modify and Delete payments
- Search on Bill Number/Mobile number/Subscriber Number

The salesperson will be able to

- Create, Modify payments
- Search on Bill Number/Mobile Number/Subscriber Number

The system should be able to generate the following reports at the end of the month:

- Subscriber list with their plan details
- List of subscribers with outstanding bills

TEAM MEMBERS:

Project Specification: 11

COLLEGE MANAGEMENT SYSTEM(CMS)

Design and develop a College Management System.
There are 2 types of user: Staff and Administrative officer
The system should maintain staff register, student register and results register.
User should be able to perform following operations:

For managing staff register: (Access is given for Administrative officers only)

1. Add a new staff

Each staff should have the following details:

Name　　　　　　　ID No
Gender　　　　　　Date of Birth
Address　　　　　　Contact number
Date of Joining　　　Qualification
Department　Job title
Basic Pay

1. Search on Id No　　　　　　2. Search on Name/Contact number
3. Modify the staff details　　4. Add a new department
5. Add a new job title

For managing student register: (Access is given for Staff and Administrative officers)

1. Add a new student.

Each student should have the following details:

Registration Number　　　　Date of Admission/Registration
Roll number　　　　　　　　Name
Gender　　　　　　　　　　Date of Birth
Address　　　　　　　　　Parent phone number
Class　　　　　　　　　　Section

1. Modify student details.
2. Search on Registration number/Roll number.
3. Search on Student name/phone number.

For managing results register: (Access is given for Staff and Administrative officers)

1. Add a new subject.

Each student should have the following details:

Registration Number　　　　Subject Name
Mark obtained　　　　　　　Pass Mark
Maximum Mark　　　Result

1. Modify result details.
2. Search on Registration number.
3. Search on subject name.

Following are the reports to be generated at the end.

1. Details of staff based on Date of Joining
2. Details of students based on Date of Admission
3. Details of students based on the Class
4. Details of students based on the Marks (less than 50, equal to 50, and more than 50)

TEAM MEMBERS:

Project Specification: 12

RECRUITMENT MANAGEMENT SYSTEM

Design and develop a Recruitment Management System.
There are 2 types of user: HR and Administrative officer
The system should maintain Associate Details, Openings, and ISU Details
User should be able to perform following operations:
For managing Associate register: (Access is given for HR only)
Add a new Associate

- Each associate should have the following details:

    Name
    Gender
    Address
    Date of Joining
    Team
    Allocated/Unallocated
    Emp ID
    Date of Birth
    Contact number
    Qualification
    Job title

Search on Id No
Search on Name/Contact number
Modify the Associate details
Add a team detail
Add an ISU details (INDUSTRY SOLUTION UNITS**)**

For managing Job openings details: (Access is given for HR)
Add an ISU.

- Each department should have the following details:

    NoofOpenings
    Yrs of Exp
    Domain
    Technology
    Willing to work in Shifts
    Designation

Modify job openings details.
Search on ISU
Search on Technology

Following are the reports to be generated at the end (Access is given for HR and Administrative officers)
Details of associate based on Unallocation
Details of the job openings based on ISU
Details of the ISU

TEAM MEMBERS:

Project Specification: 13

FOOD BOX

Design and Develop a Food Box
There should be 2 types of user level: Administration and user.

The inventory module should consist of the following fields:

- Food Number
- Food Name
- Food Type [South Indian/North Indian/Chinese/Continental]
- Price
- Quantity

The access is given to Administrator alone. An Administrator should be able to

- Create, Modify and Delete Foods
- Search on Food Number/Food Name

The Sales module should consist of

- Date
- Food Number
- Food Name
- Quantity
- Price
- Total Cost

This module is accessible by Administrator and User.
The Administrator should be able to

- Create, Modify and Delete items

The User should be able to

- Create, Modify items

At the end of the day,

- The sum of quantities of a particular Food in the sales module should be tallied with the Inventory module and report should be generated on the current stock of Foods.
- The total cost in the sales module should be tallied with sum of the prices of individual Foods sold.

TEAM MEMBERS:

Project Specification: 14

ONLINE TRAIN TICKETING

Design, Develop and Test an Online Train Ticketing System.
The application is a ticketing system for a government center for citizens of the state. Users log in and out, can book, plan and cancel tickets.
Implement a module for user registration. Collect the name, age, contact number, address from the user while registering and any additional fields if required.
After logging in,

1. The Home page should display two drop-down fields – From and To (have a predefined list of cities),
2. Date field and the number of tickets. By selecting these values, a page with the number of available services along with the timings and the rate should be listed.
3. Selecting a service should display the number of seats available. There should be a buy ticket button next to the selected service. Clicking on it should take the user payment page.
4. Get all the details necessary for payment like credit card details, net banking, etc. Clicking on the Confirm button should display a message saying. 'Ticket booked. Your seat number is xxx'.
5. The next user who queries the same service should see one seat less in the available seats.
6. Maintain a history of all the tickets booked for a user. He should be able to see the list of tickets booked by him.
7. The user should also be able to cancel the booked tickets till the journey date.
8. Display appropriate messages wherever applicable.

TEAM MEMBERS:

Project Specification: 15

ONLINE BUS TICKETING SYSTEM

Design, Develop and Test an Online Bus Ticketing System.

The application is a ticketing system for a government center for citizens of the state. Users log in and out, can book, plan and cancel tickets.

Implement a module for user registration. Collect the name, age, contact number, address from the user while registering and any additional fields if required.

After logging in,

1. The Home page should display two drop-down fields – From and To (have a predefined list of cities),
2. Date field and the number of tickets. By selecting these values, a page with the number of available services along with the timings and the rate should be listed.
3. Selecting a service should display the number of seats available. There should be a buy ticket button next to the selected service. Clicking on it should take the user payment page.
4. Get all the details necessary for payment like credit card details, net banking, etc. Clicking on the Confirm button should display a message saying. 'Ticket booked. Your seat number is xxx'.
5. The next user who queries the same service should see one seat less in the available seats.
6. Maintain a history of all the tickets booked for a user. He should be able to see the list of tickets booked by him.
7. The user should also be able to Cancel the booked tickets till the journey date.
8. Display appropriate messages wherever applicable.

TEAM MEMBERS:

Project Specification: 16

E-NURSERY

The e-Nursery example is a simple e-commerce application where users can select Plants, view the Plant store catalog, and purchase Plants.

- Implement a module for user registration. Collect the name, age, contact number, shipping and billing address from the user while registering and any additional fields if required.
- Plants should be organized into various categories. Display all these categories in the home page. Selecting a category should display all the Plants under the category.
- A guest (who is not yet logged in) should be able to query for any Plant. To buy a Plant the user should be logged in. Implementing a shopping cart is optional.

There should be a Buy Plant button under each Plant. This should take the user to the buy Plant page. Also, display the count of number of available for the selected Plant.

If zero, display appropriate message, otherwise, the customer information should be auto-populated in the buy page. However, the user should be able to edit any information.

The user should be able to select the number of copies required. The total price should be displayed to the user. By clicking on ‘Submit Order’, display success message.

- Two kinds of roles exist for the system: A regular user (end user) and an admin (or a super user). A regular user should only be able to buy Plants. However an admin can additionally add Plants to the inventory and query all the orders placed in the website.
- Maintain a log of all the Plants sold including the details of the user who bought it. An user with the role of ‘Admin’/’SuperUser’ should be able to query all the orders placed in the website.

Display appropriate messages wherever applicable.

As a high level table structure, the following tables will be required:

Plant, User, Role, Inventory Tables,

The Plant entity should define at-least the following attributes:

- A Plant ID
- The Plant name
- The Plant Botanical name
- The price
- Whether the Plant is on sale
- A description of the Plant

TEAM MEMBERS:

Project Specification: 17

E-STATIONERY

The e-Stationery example is a simple e-commerce application where users can select Stationery, view the Stationery Items catalog, and purchase Stationery Items.

- Implement a module for user registration. Collect the name, contact number, shipping and billing address from the user while registering and any additional fields if required.
- Stationery Items should be organized into various categories. Display all these categories in the home page. Selecting a category should display all the Stationery Items under the category.
- A guest (who is not yet logged in) should be able to query for any Stationery Items. To buy Stationery items the user should be logged in. Implementing a shopping cart is optional.

There should be a Buy Stationery button under each stationery. This should take the user to the buy Stationery page. Also, display the count of number of available stationery items.
If zero, display appropriate message, otherwise, the customer information should be auto-populated in the buy page. However, the user should be able to edit any information.
The user should be able to select the number of stationery items required. The total price should be displayed to the user. By clicking on 'Submit Order', display success message.

- Two kinds of roles exist for the system: A regular user (end user) and an admin (or a super user). A regular user should only be able to buy stationery. However an admin can additionally add stationery to the inventory and query all the orders placed in the website.
- Maintain a log of all the stationery items sold including the details of the user who bought it. An user with the role of 'Admin'/'SuperUser' should be able to query all the orders placed in the website.

Display appropriate messages wherever applicable.
As a highlevel table structure, the following tables will be required:
Stationery items, User, Role, Inventory Tables,
The Stationery entity should define at-least the following attributes:

- Code Number
- Stationery name
- The price
- Whether the stationery items is on sale
- A description of the stationery items

TEAM MEMBERS:

Project Specification: 18

BOOKSTORE

The Bookstore example is a simple e-commerce application where users can select books, view the bookstore catalog, and purchase books.

- Implement a module for user registration. Collect the name, age, contact number, shipping and billing address from the user while registering and any additional fields if required.
- Books should be organized into various categories. Display all these categories in the home page. Selecting a category should display all the books under the category.
- A guest (who is not yet logged in) should be able to query for any book. To buy a book the user should be logged in. Implementing a shopping cart is optional.

There should be a Buy book button under each book. This should take the user to the buy book page. Also, display the count of number of copies available for the selected book.
If zero, display appropriate message, otherwise, the customer information should be auto-populated in the buy page. However, the user should be able to edit any information.
The user should be able to select the number of copies required. The total price should be displayed to the user. By clicking on 'Submit Order', display success message.

- Two kinds of roles exist for the system: A regular user (end user) and an admin (or a super user). A regular user should only be able to buy books. However, an admin can additionally add books to the inventory and query all the orders placed in the website.
- Maintain a log of all the books sold including the details of the user who bought it. An user with the role of 'Admin'/'SuperUser' should be able to query all the orders placed in the website.

Display appropriate messages wherever applicable.

As a high level table structure, the following tables will be required:

Book, User, Role, Inventory Tables,
The Book entity should define at-least the following attributes:

A book ID
The author's first name
The author's surname
The title
The price
Whether the book is on sale
The publication year
A description of the book
The number of copies in the inventory

TEAM MEMBERS:

Project Specification: 19

Alumni Information System

Every school has network of Alumni who were once part of school and are now placed well in their lives. Alumni of a school generally stay in touch with their immediate friends but find it hard to stay connected with other school mates. Contact between alumni can be used to gain insight in a new field, career counselling or guidance.

The project Alumni Information System which allows get together old students and new students of a school to communicate with each other. The main purpose of this project is to provides all the detail about Alumni, School, School Mates, Social Network, Opportunity, School News, Student. In this system students to know information about each other and their current activities. The New students can ask about career, subject details with old students who has completed his studies. Alumni can also provide public posts on the system about possible career opportunities or other school related news. Students who had completed his studies and begin his professional carrier can be helpful for other students and schools for providing guidelines for new students, so this project Alumni Information System will help school and students to be in contact with alumni.

Project Modules:

- **Admin**:The admin will have to ensure that passing out students are shifted into the alumni module. The admin will also have to browse the site to ensure no objectionable content is posted.
- **User login**: Students or Alumni have to register themselves into the system to create an account. After registering successfully, they can then login into the system.
- **Search option**: Admin, Alumni or Students can search profiles of other students or alumni.
- **Forum**: Alumni or Students can post their queries and answers in the forum.

Software to be used:

- Python 3.X installation
- MySQL for storing information

TEAM MEMBERS:

Project Specification: 20

Banking Management System

The Banking Management System is an application for maintaining a person’s account in a bank. The system provides the access to the customer to create an account, deposit/withdraw the cash and other core banking features from his account. It also enables customer to view reports of all accounts present.

Project Modules:

- **Registration**: A customer can create an account in the bank by providing important information such as personal details
- **Login/Logout**: As any secure internet facing financial solution, login and logout will be provided by Banking management system
- **Core Operations**: This module enables Deposit or withdraw functionality to the customer. User can also check the balance
- **Reports**: This module will generate different kind of statements and can be used for checking balance.
- **Profile Management**: User can update his details like contact information etc

Software to be used:

- Python 3.x installation
- MySQL for storing information

TEAM MEMBERS:

Project Specification: 21

Library Management System

A school library management is a project that manages and stores books information electronically according to students needs. The system helps both students and library manager to keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for schools to keep a continuous check on the books issued and returned and even calculate fine. This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes.Hence this system reduces manual work to a great extent allows smooth flow of library activities by removing chances of errors in the details.

Project Modules:

- **Admin login**: Admin is the one who administers the system by adding or removing books into and from the system respectively.
- **Add and Update Books**: The admin can add books to the system by entering the details of the books and can even update the details.
- **User login**: Students have to register themselves into the system to create an account. After registering successfully, they can then login into the system.
- **Search option**: Admin and Students can search for books by entering the name of the book.
- **View Issued Books**: The admin can view issued books.
- **Issue Book**: The students can place issue request for the books and simultaneously the quantity of the book issued will be decremented.
- **Calculate Fine**: The student can view the issue and expiry date for the book issued and can even calculate fine.

Software to be used:

- Python 3.X installation
- MySQL for storing information

TEAM MEMBERS:

Project Specification: 22

Student Attendance Management System

The main aim of Student attendance system project is to maintain attendance records of student for any organization school or college. The attendance project has three user modules to run the system Admin, Staff and Student. Initially the system will be blank, The Administrator has rights to create classroom for school and same time he has to add staff/teacher's detail. Administrator generates unique username and password for all staff while adding staff detail. All staff maintain attendance of student, generate reports.

Project Modules:

- **Admin Functionalities**: The admin can add, update or delete classroom & division. Admin can also add, update or delete staff/teachers.
- **Staff/Teacher** : Staff can add, update or delete student. The main aim of the attendance system is fulfilled by staff. Staff can fill the attendance daily for his division.
- **Student**: After login into system student can view his attendance reports and manage his account.

Software to be used:

- Python 3.x installation
- IDE such as Spyder or PyCharm for project development (Optional)
- MySQL for storing information

TEAM MEMBERS:

Project Specification: 23

Student Report card Generation System

The objective of Student Report card generation system project is to generate report record of a student for any organization, school or college. The project has three user modules for running the system Admin, Staff and student. Initially the system will be blank, The Administrator has rights to create classroom for school and same time he has to add staff details. Assign unique username and password for each teacher to access the student module. The student module captures the student details, such as Rollno, Name, class, section, gender, marks in 5 subjects. The system then generates a achievement record.

Project Modules:

- **Admin Functionalities**: The admin can add, update or delete classroom & division. Admin can also add, update or delete staff details.
- **Staff/Teacher**: Staff can add, update or delete student. The main aim of the system is fulfilled by staff. Staff can fill the marks of a particular exam.
- **Student**: After login into system student can view his reports and manage his account.

Software to be used:

- Python 3.x installation
- MySQL for storing information

TEAM MEMBERS:

Project Specification: 24

Hospital Management System

Hospital management encompasses all facets of a hospital, as well as the coordination of all of its components. This could include everything from patient care to record keeping to drug inventory and cleanliness.

To ensure that their company runs successfully, all hospital healthcare managers collaborate with clinicians, make policy choices, oversee patient care, budgeting and accounting, and lead marketing activities.

Patient registration, appointment scheduling, document management, consultation management, lab management, drug safety, report creation, personnel management, outpatient management, and much more are all part of the hospital administration process.

Modules:
Patient Management: This module covers from the process of intake until discharge of an account of the patient's engagement with the health-care team. Communication, empathy, examination, evaluation, diagnosis, prognosis, and intervention are all part of the process.
Doctor/Physician Management: The management of the physicians would be included in creating this system. Through this process, the admin will have the information and transactions made by the doctors with the patients.
Medicine and Prescription Management: This module will handle the process of monitoring a patient's medications to verify that they are taken correctly and that the intended therapeutic outcome is achieved.
Online Appointment Management: This process is a tool that helps hospital admin manage their appointments. Internet booking is one of the tools available in an appointment management solution. Booking with a mobile app.
Medical and Transaction Management: Medical and transaction management modules aims to secure every transaction made by the patients and physicians in order to enhance healthcare quality and outcomes.
Payment and Expense Management: Payment and expense management module is meant to assist the admin in the payment management process. This will help the hospital with the full payment processing and accounts payable process.

TEAM MEMBERS:

Project Specification: 25

SCHOOL ELECTION SYSTEM

Develop a software system to automate the election process.
The election is conducted in such a way that a teacher can cast votes for all the kudumbas, where as a student is eligible to vote only to the kudumba he/she belongs to.
The post for which the students nominated are (gender wise):

- Cultural captain
- Sports caption
- Kudumba captain (4 kudumbas)

The system should collect the nominees list.
Validate the voter for a staff / student and accordingly redirect
The system should generate an overall report of winners (gender wise) with vote tally.

TEAM MEMBERS:

Project Specification: 26

HEALTH INSURANCE MANAGEMENT SYSTEM

The Health Insurance Management system need to keep track of details of policy holders, their premium payments and various types of insurance are available with it. Hence it is tremendous pressure maintaining their day to day activities, which is done manually. Entire records have to be updated timely even a slight could complicate things.

The Health Insurance Management system is a complete solution for organizations, which need to manage insurance of their employees. Organize and track insurance vendors and policies provided by coverage

Table description

Employee:
Description This table stores the information about employee.
Attributes E_id, emp_name, phone_no, address, dob
Primary key E_id Foreign key Null

Insurance:
Description This table stores the information about the insurance and its type.
Attributes ins_id, Ins_amt, ins_type, duaration, Start_date, end_date, E_id
Primary key ins_id Foreign key E_id

Claim:
Description This table stores the claim information.
Attributes claim_id, claim_amt, iss_date, ins_no
Primary key claim_id Foreign key ins_id

Disease:
Description This table stores the information about disease.
Attributes code, name
Primary key Code Foreign key Null

Hospital:
Description This table stores the information about the hospital.
Attributes hos_id, name, address, E_id, dis_code
Primary key hos_id Foreign key E_id,dis_code

Billing:
Description This table stores the information about the bill.
Attributes bill_id, bill_amt, bill_date, hos_id, claim_id
Primary key bill_id Foreign key hos_id, claim_id

TEAM MEMBERS:
