**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

"Jnana Sangama", Belagavi: 590 018



A Database Management Systems Mini Project report on

**Arena Rental Stores**

 Submitted in partial fulfillment of the requirement for the award of Degree of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

By

**NAME: Amoolya M Kulkarni USN: 1AY19CS006**

**NAME: Sarfaraz Contractor USN: 1AY16CS131**

Under the guidance of

**Prof. T Sujatha**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ACHARYA INSTITUTE OF TECHNOLOGY**

(Affiliated to Visvesvaraya Technological University, Belagavi)

**2021-2022**

**ACHARYA INSTITUTE OF TECHNOLOGY**

(Affiliated to Visvesvaraya Technological University)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**Certificate**

Certified that the Database Management Systems mini project entitled **“Arena Rental Stores”** is a bonafide work carried out by **Amoolya M Kulkarni (1AY19CS006) and Sarfaraz Contractor (1AY16CS131)** of 5th semester in partial fulfillment for the award of degree of **Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University**, **Belagavi**, during the year **2020-2021.** It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

**Signature of Guides                                                                                   Signature of H.O.D**

**Name of the examiners**                                       **Signature with date**

1.

2.

**ACKNOWLEDGEMENT**

I express my gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff whose gratitude was of immense help in completion of this seminar successfully.

I express my sincere gratitude to our principal, **Dr. Prakash M R** for providing required environment and valuable support for developing this mini project.

My sincere thanks to **Dr. Ajith Padyana,** Head of the Department, Computer Science and Engineering, Acharya Institute of Technology for his valuable support and for rendering us resources for this mini project work.

I express my gratitude to **Prof. T Sujatha,** Assistant Professor, Dept. Computer Science and Engineering, Acharya Institute of Technology who guided me with valuable suggestions in completing this mini-project at every stage.

My gratitude thanks should be rendered to many people who helped me in all possible ways.

**Amoolya M Kulkarni**

**(**1AY19CS006**)**

**Sarfaraz Contractor**

**(**1AY16CS131**)**

**Abstract**

Arena Rentals is a web based application for a sports equipment rental system. It is a one-stop destination to rent out all the gear needed for a sport. Its end user can be teams who will require it in bulk, individuals who are looking for cheaper options than buying it, or just anyone looking to rent sports gears. With the help of this web application, they can rent out the equipment for a period of one month and use it. A user, after having signed up and logged in can choose from the various sports kits available to rent out the one he needs. A user can rent out more than one gear at a time, but the returning of the equipment is done in such a way that the kit rented first will be returned first, and then the second and so on. The web app also keeps a track of a user’s rental history and a user can see his previous transactions, under the ”My Profile” section of the web application, meaning he can look at all the gears he has rented out previously and has since returned. In a world heading towards digitalization, this web application is a great way to book your sports gears online using a virtual interface without having to go down to the store.

**TABLE OF CONTENTS**

**Acknowledgement i**

**Abstract ii  
1. Introduction 1 - 4**1.1 Introduction to DBMS 1  
 1.1.1 Why DBMS? 1  
 1.1.2 Database applications 1  
 1.1.3 Advantages of DBMS 2  
 1.1.4 Components of DBMS 3  
**2. System Requirements 5**2.1 Functional Requirements 5  
2.2 Non-Functional Requirements 5

2.2.1 Hardware Requirements 5

2.2.2 Software Requirements 5

**3. Design 6 - 7**3.1 ER Diagram 6  
3.2 Schema Diagram 7

**4. Implementation 8-16**4.1 Table Creation 8

4.1.1 Signup Page 8

4.1.2 Equipment Page 9

4.1.3 Issue Page 10

4.1.4 Return Page 11

4.1.5 Pay Page 12

4.2 Table Insertion 13

4.2.1 Insertion for Equipment Page 13

4.3 Queries and Frontend 15

4.3.1 Triggers 15

**5. Results and Implementation 17-23**

5.1 Signup Page 17

5.2 Login Page 18  
 5.3 Equipment Page 19

5.4 Issue Page 20

5.5 Billing Page 21

5.6 Return Page 22

5.7 My Profile Page 22

5.8 About us Page 23 **Conclusion 24**  
**Bibliography 25**

**LIST OF FIGURES**

1.1 Components of Database Management System 3

3.1 Entity Relationship Diagram 6

3.2 Schema Diagram 7

5.1 Signup Page 17

5.2 Login Page 18

5.3.1 Equipment Page 1 19

5.3.2 Equipment Page 2 19

5.3.3 Equipment Page 3 20

5.4 Issue Page 21

5.5 Billing Page 21

5.6 Return Page 22

5.7 My Profile Page 22

5.8 About us Page 23

**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction to DBMS**

DBMS stands for **D**ata**b**ase **M**anagement **S**ystem. We can break it like this DBMS = Database + Management System. The database is a collection of data and a Management System is a set of programs to store and retrieve those data. Basically, DBMS is a software tool to organize (create, retrieve, update and manage) data in a database.

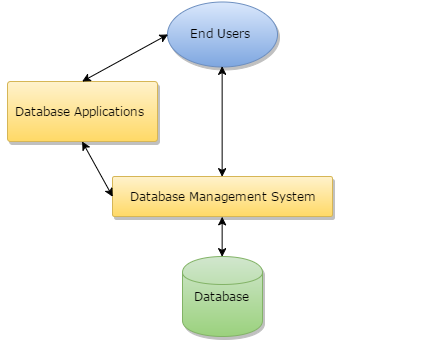
The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. Normally people use software such as DBASE IV or V, Microsoft ACCESS, or EXCEL to store data in the form of a database. A datum is a unit of data. Meaningful data combined to form information. Hence, information is interpreted data – data provided with semantics. MS. ACCESS is one of the most common examples of database management software.

Database systems are meant to handle large collections of information. Management of data involves both defining structures for the storage of information and providing mechanisms that can do the manipulation of those stored information. Moreover, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

* + 1. **Why DBMS?**
* To develop software applications in less time.
* Data Independence and efficient use of data.
* For uniform data administration.
* For data integrity and security.
* For concurrent access of data, and data recovery from crashes.
* To use user-friendly declarative query language.
  + 1. **Database applications**
* **Telecom:** There is a database to keep track of the information regarding calls made, network usage, customer details, etc. Without the database systems, it is hard to maintain that huge amount of data that keeps updating every millisecond.
* **Industry:** Where it is a manufacturing unit, warehouse, or distribution center, each one needs a database to keep the records of ins and outs. For example, distribution centers should keep a track of the product units that are supplied into the centre as well as the products that got delivered out from the distribution centre on each day; this is where DBMS comes into the picture.
* **Education sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot of interrelated data that needs to be stored and retrieved efficiently.
* **Online shopping:** You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.
* **Banking system:** For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.
  + 1. **Advantages of DBMS**

A DBMS manages data and has many advantages.

* **Data Independence:** Application programs should be as free or independent as possible from details of data representation and storage. DBMS can supply an abstract view of the data for insulating application code from such facts.
* **Efficient data access:** DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data competently and this feature becomes important in cases where the data is stored on external storage devices.
* **Data integrity and security:**  If data is accessed through the DBMS, the DBMS can enforce integrity constraints on the data.
* **Data administration:** When several users share the data, integrating the administration of data can offer major improvements. Experienced professionals understand the nature of the data being managed and can be responsible for organizing the data representation to reduce redundancy and make the data to retrieve efficiently.
* **Providing backup and recovery:** A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery.
* **Permitting inferencing and actions using rules:** Some database systems provide capabilities for defining deduction rules for inferencing new information from the stored database facts.
  + 1. **Components of DBMS**

****

**Fig-1.1: Components of a Database Management System**

* **Users:** Users may be of any kind such as DB administrator, System developer or database users.
* **Database application:** Database application may be Departmental, Personal, organization’s and / or Internal.
* **DBMS:** Software that allows users to create and manipulate database access.
* **Database:** Collection of logical data as a single unit.
* **Database access language:** This is used to access the data to and from the database, to enter new data, update existing data, or retrieve required data from databases. The user writes a set of appropriate commands in a database access language, submits these to the DBMS, which then processes the data and generates and displays a set of results into a user readable form.

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

**2.1 Functional Requirements:**

* User can log into the system.
* User can select any one of the given kits from the kit list.
* User can choose the date of issue.
* User can return the kits in a chronological manner.
* User can view his previous transactions.

**2.2 Non Functional Requirements**

**Hardware Requirements**

* **Processor:** Intel Core2 Quad @ 2.4Ghz on Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit.
* **RAM:** 2GB of RAM
* **Memory:** 256GB Hard drive
* **Keyboard:** MS-compatible keyboard
* **Mouse:** MS-compatible mouse

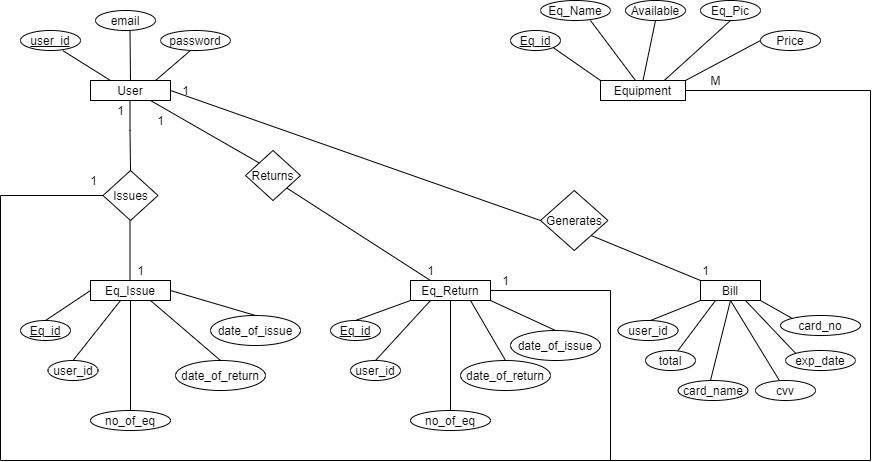
**Software Requirements**

* **Operating system:** Windows® Vista 64-Bit / Windows® 7 64-Bit / Windows® 8 64-Bit / Windows® 8.1 64-Bit.
* **Front end:** HTML, BOOTSTRAP, JAVASCRIPT
* **Back end:** NODEJS and EXPRESS
* **Orm:** SEQUELIZE
* **IDE:**  VS Code

**DESIGN**

**CHAPTER 3**

**3.1 ER Diagram**

****

**Fig-3.1: Entity Relationship Diagram**

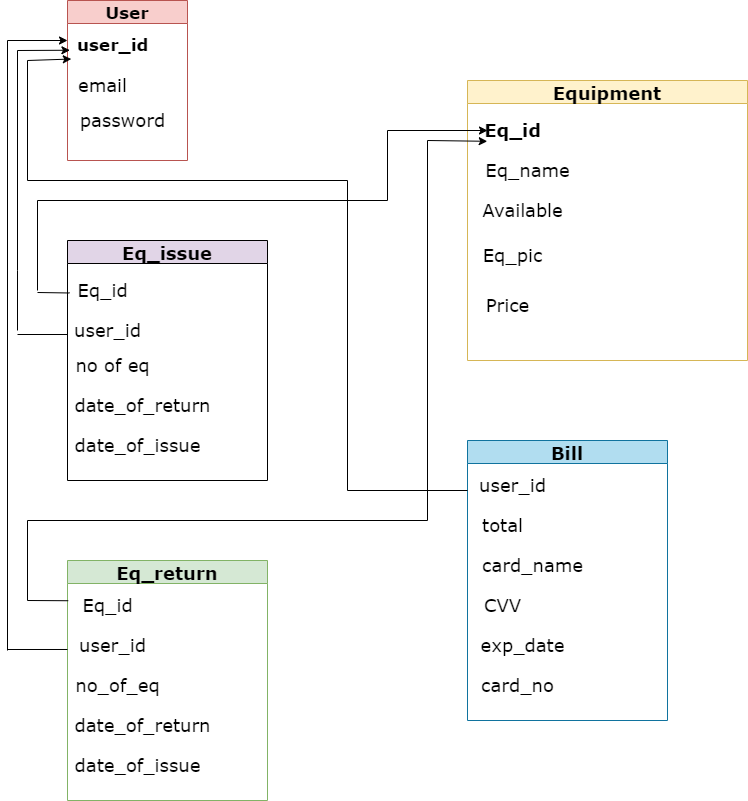
An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

The relations showed in this diagram are the following:

1:1 – One-to-One - each user can issue and return one kind of equipment at a time.

M:1 – Many-to-One – more than one equipment of one kind can be issued by a user at a time.

**3.2 Schema Diagram**

****

**Fig-3.2: Schema Diagram**

The description of a database is called the database schema, which is specified during database design and is not expected to change frequently. A schema diagram displays only some aspects of a schema, such as the names of record types and data items, and some types of constraints. The schema is sometimes called the intension, and a database state is called an extension of the schema. Application requirements change occasionally.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 Table Creation**

**4.1.1 Signup page**

const connection = require("../DBConnection/connection") //connection

const sequelize = require("sequelize");

const { DataTypes } = require('sequelize');

const User = connection.define('SignUpPage',

{

userid: {

type: DataTypes.INTEGER,

primaryKey: true,

autoIncrement: true

},

emailid: {

type: DataTypes.STRING,

primaryKey: true

},

password: {

type: DataTypes.STRING,

allowNull: false

}

},

{

initialAutoIncrement: 101,

}

);

console.log(User);

module.exports = User;

**4.1.2 Equipment Page**

const connection = require("../DBConnection/connection") //connection

const sequelize = require("sequelize");

const { DataTypes } = require('sequelize');

const User = connection.define('EquipmentPage', {

equipment\_no : {

type: DataTypes.INTEGER,

primaryKey: true,

autoIncrement: true

},

equipment\_name : {

type: DataTypes.STRING,

allowNull: false

},

equipment\_available : {

type: DataTypes.INTEGER,

allowNull: false

},

equipment\_img : {

type: DataTypes.STRING,

allowNull: false

},

price : {

type: DataTypes.INTEGER,

allowNull: false

}

}, {

hasTrigger: true

}

);

console.log(User);

module.exports = User;

**4.1.3 Issue Page**

const connection = require("../DBConnection/connection") //connection

const sequelize = require("sequelize");

const { DataTypes } = require('sequelize');

const User = connection.define('IssuePages', {

userid: {

type: DataTypes.INTEGER,

},

equipmentId: {

type: DataTypes.INTEGER,

},

equipmentName: {

type: DataTypes.STRING,

allowNull: false

},

date\_of\_issue: {

type: DataTypes.STRING,

allowNull: false

},

date\_of\_return: {

type: DataTypes.STRING,

allowNull: false

},

number\_of\_eq: {

type: DataTypes.STRING,

allowNull: false

},

price: {

type: DataTypes.INTEGER,

allowNull: false

},

returned: {

type: DataTypes.BOOLEAN,

allowNull: false,

defaultValue: false

}

},{

hasTrigger: true

});

console.log(User);

module.exports = User;

**4.1.4 Return Page**

const connection = require("../DBConnection/connection") //connection

const sequelize = require("sequelize");

const { DataTypes } = require('sequelize');

const User = connection.define('ReturnPage', {

userid: {

type: DataTypes.INTEGER,

primaryKey: true,

autoIncrement: true

},

equipmentId: {

type: DataTypes.INTEGER,

primaryKey: true,

},

equipmentName: {

type: DataTypes.STRING,

allowNull: false

},

// date\_of\_issue: {

// type: DataTypes.STRING,

// allowNull: false

// },

date\_of\_return: {

type: DataTypes.STRING,

allowNull: false

},

number\_of\_eq: {

type: DataTypes.STRING,

allowNull: false

}

},{

hasTrigger: true

});

console.log(User);

module.exports = User;

**4.1.5 Pay Page**

const connection = require("../DBConnection/connection") //connection

const sequelize = require("sequelize");

const { DataTypes } = require('sequelize');

const User = connection.define('PayPage',

{

userid: {

type: DataTypes.INTEGER,

primaryKey: true,

autoIncrement: true

},

equipmentId: {

type: DataTypes.INTEGER,

primaryKey: true,

},

equipementName: {

type: DataTypes.STRING,

allowNull: false

},

number\_of\_eq: {

type: DataTypes.STRING,

allowNull: false

},

total: {

type: DataTypes.INTEGER,

allowNull: false

}

},

{

hasTrigger: true

});

console.log(User);

module.exports = User;

**4.2 Table insertion**

**4.2.1 Insertion for equipment page**

const data = await EquipmentPage.bulkCreate([

{

equipment\_name: "Tennis",

equipment\_available: 15,

equipment\_img: "https://www.sportsmomsurvivalguide.com/wp-content/uploads/2018/06/Wilson-Tennis-Racket-1.jpg",

price: 200

},

{

equipment\_name: "Baseball",

equipment\_available: 30,

equipment\_img: "https://bloximages.newyork1.vip.townnews.com/dothaneagle.com/content/tncms/assets/v3/editorial/0/b2/0b28042e-6dae-11eb-97c2-8f164877b14e/60274ba4c866b.image.jpg?resize=1200%2C800",

price: 350

},

{

equipment\_name: "Archery",

equipment\_available: 25,

equipment\_img: "https://m.media-amazon.com/images/I/71Q-IbUOQlL.\_AC\_SL1500\_.jpg",

price: 500

},

{

equipment\_name: "Cricket",

equipment\_available: 40,

equipment\_img: "https://5.imimg.com/data5/SELLER/Default/2021/2/QT/JD/AW/119792758/rk-nimbus-cricket-kit-men-size-500x500.png",

price: 350

},

{

equipment\_name: "Hockey",

equipment\_available: 40,

equipment\_img: "https://m.media-amazon.com/images/I/71Q-IbUOQlL.\_AC\_SL1500\_.jpg",

price: 300

},

{

equipment\_name: "Badminton",

equipment\_available: 50,

equipment\_img: "https://m.media-amazon.com/images/I/81-4rp3jnxL.\_SL1500\_.jpg",

price: 150

},

{

equipment\_name: "Golf",

equipment\_available: 10,

equipment\_img: "https://i.insider.com/5f21b84524381734ea40df21?width=1136&format=jpeg",

price: 700

},

]);

}

**4.3 Queries and Frontend**

**4.3.1 Triggers**

const trigger = async () => {

const trigger\_decrement = await connection.query('CREATE TRIGGER decrement AFTER INSERT ON issuepages' +

' FOR EACH ROW' +

' BEGIN' +

' UPDATE equipmentpages SET equipment\_available = equipment\_available - NEW.number\_of\_eq WHERE equipment\_no = NEW.equipmentId;' +

'END;')

const trigger\_increment = await connection.query('CREATE TRIGGER increment AFTER INSERT ON returnpages' +

' FOR EACH ROW' +

' BEGIN' +

' UPDATE equipmentpages SET equipment\_available = equipment\_available + NEW.number\_of\_eq WHERE equipment\_no = NEW.equipmentId;' +

'END;')

}

trigger();

**CHAPTER 5**

**RESULTS AND DISCUSSION**

**5.1 Signup Page**

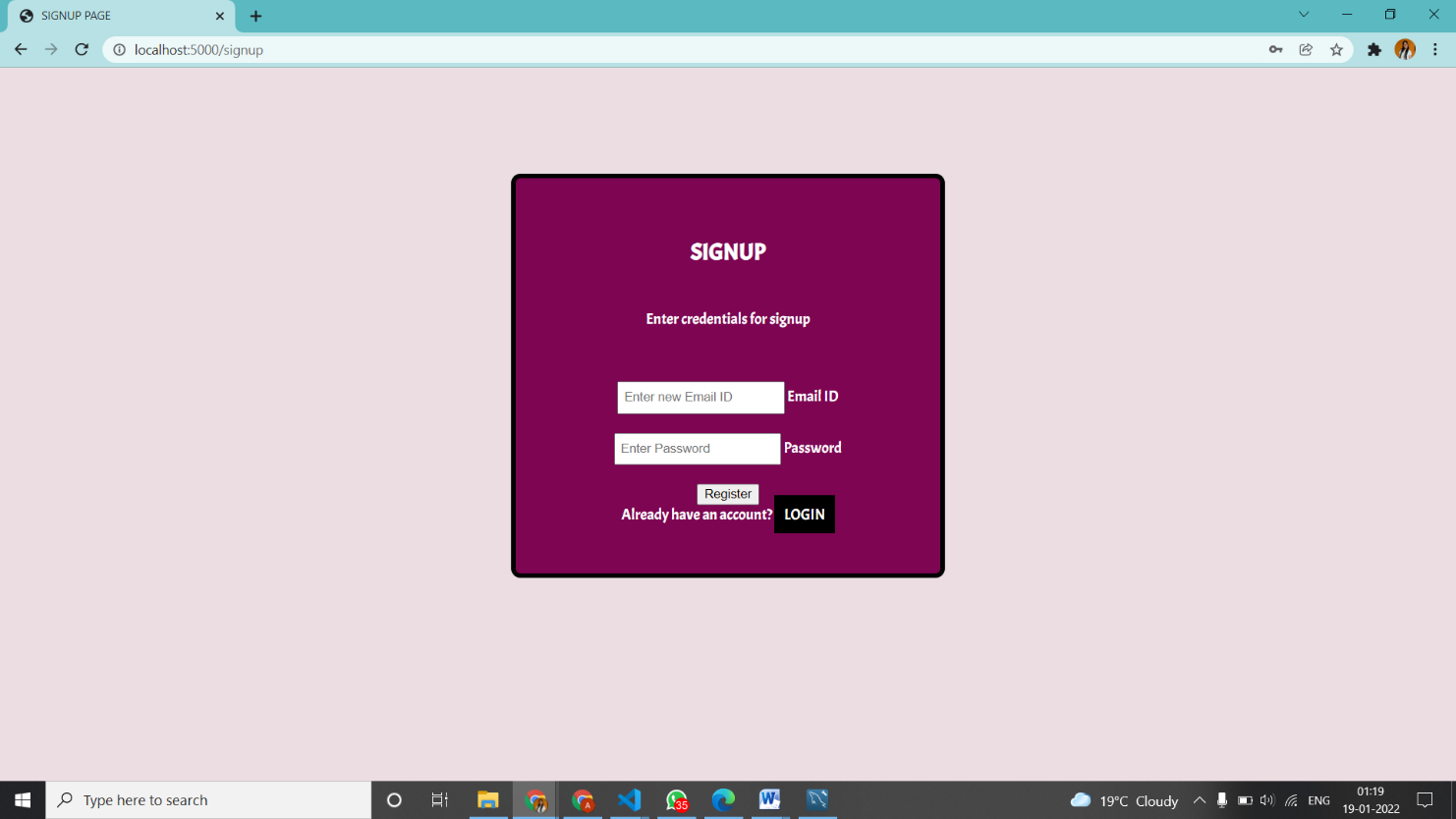
****

Fig 5.1 Signup Page

* This is a signup page where the user registers themselves by providing their email ID and password.
* The password must have a minimum of 8 characters.
* On clicking “Register” the user is redirected to the login page.
* If the user already has an account, then they can directly login by clicking on the “Login” button which takes them to the login page

**5.2 Login page**

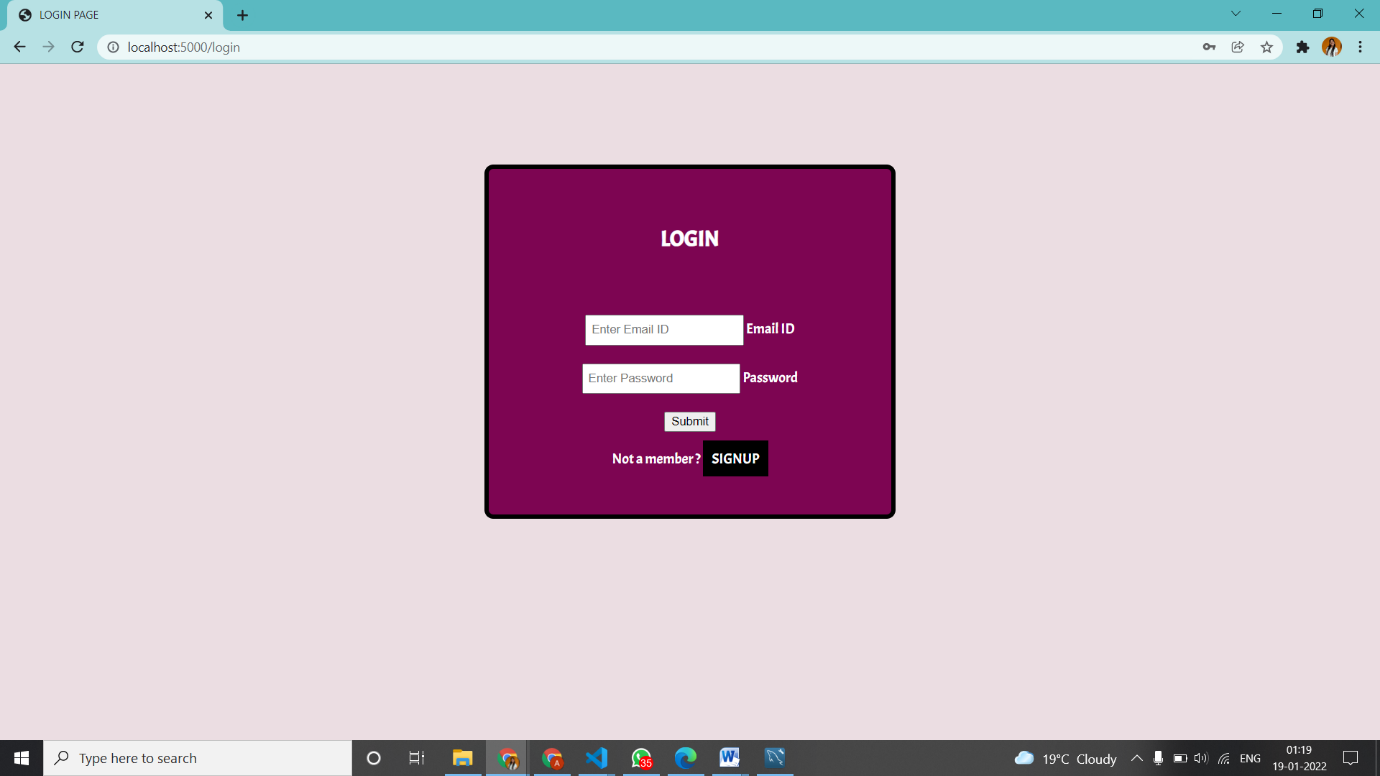
****

Fig 5.2 Login Page

* A user can login only if they have registered previously. Else they can register themselves by clicking on the signup button which takes them to the signup page.
* The email ID and password of the user should match the ones given during registration.
* Once the user types in their credentials and submits they are redirected to the equipment page.

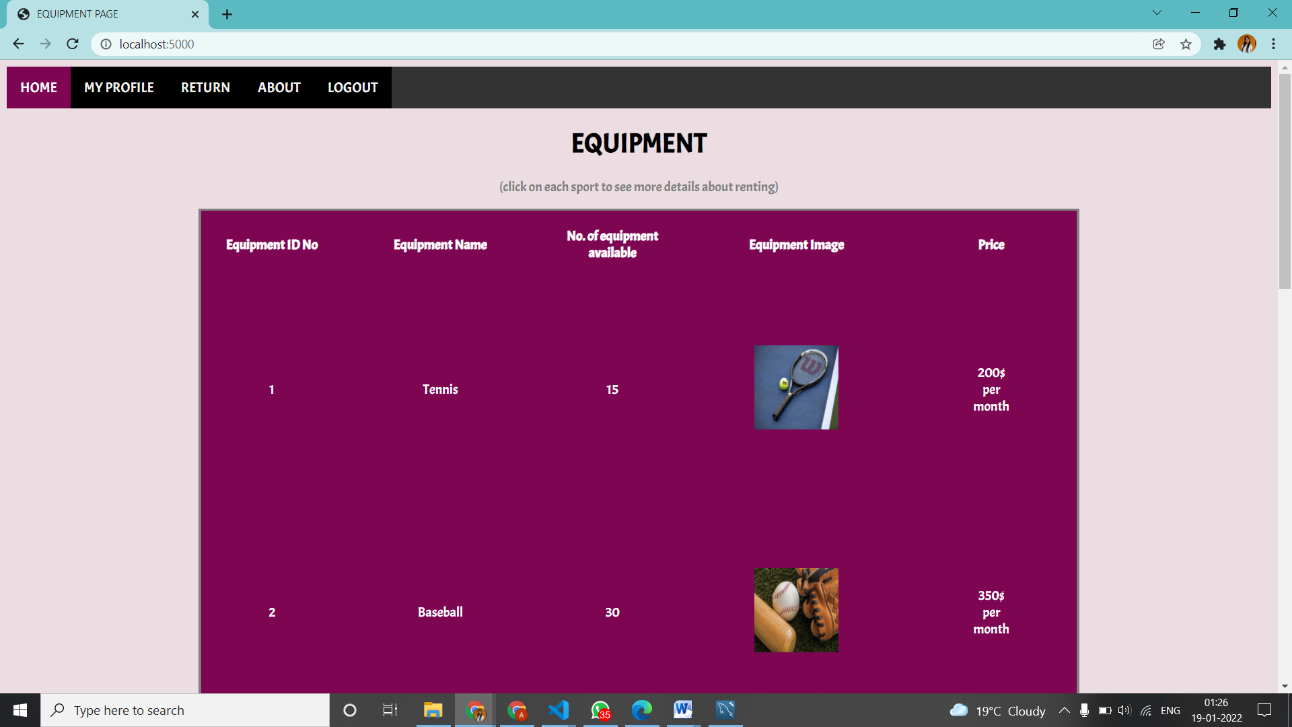
**5.3 Equipment Page **

Fig 5.3.1

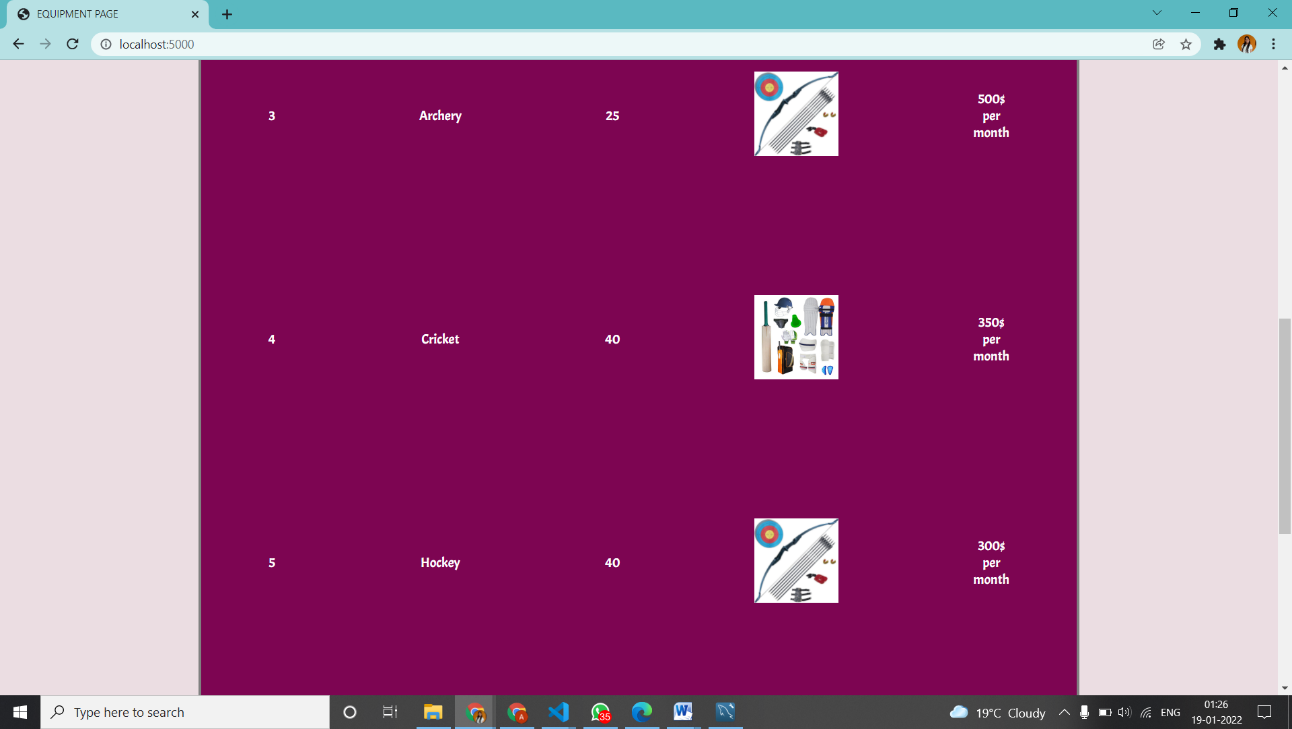
****

Fig 5.3.2

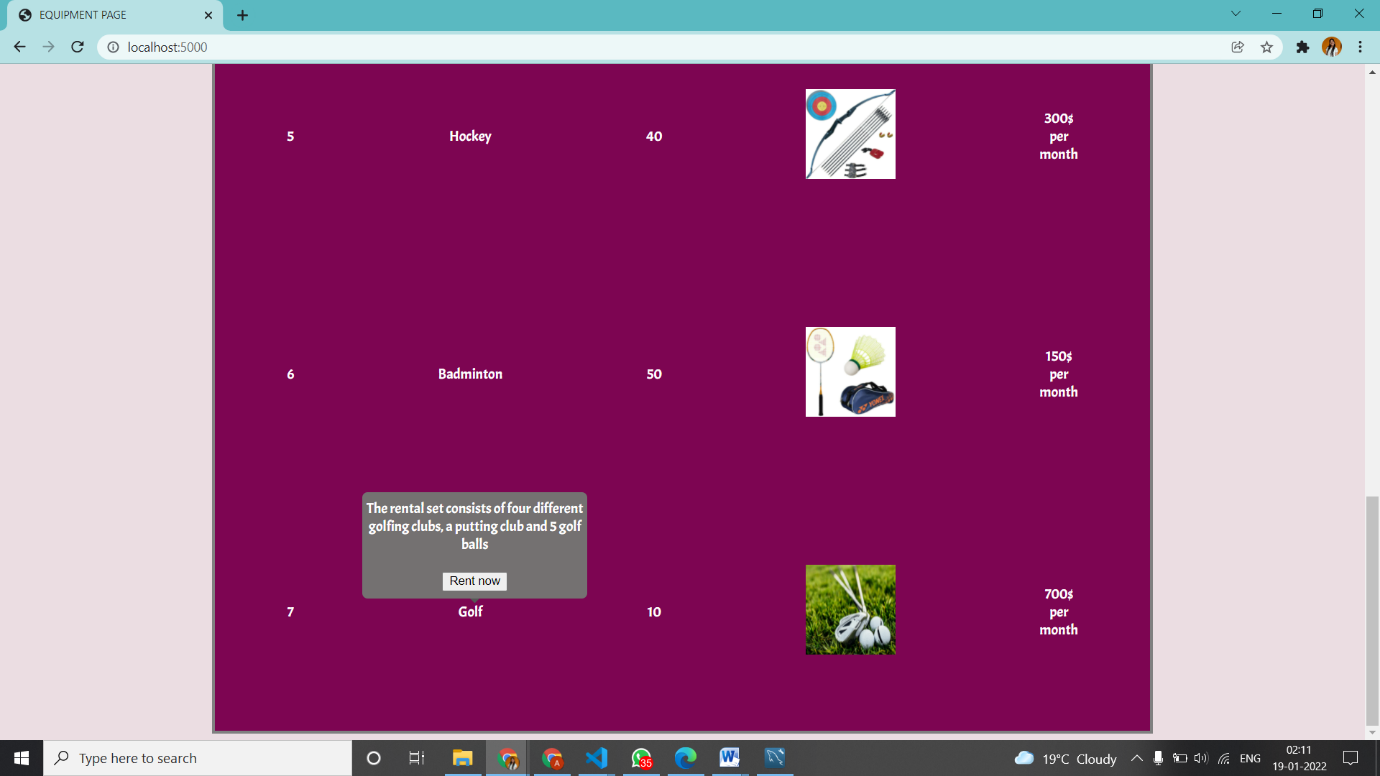
****

Fig 5.3.3

Equipment page

* This is the homepage of the project.
* User can select an equipment to rent by clicking on the name of the sport and selecting the “Rent now” option.
* The user is now redirected to the Issue page.

**5.4 Issue Page**

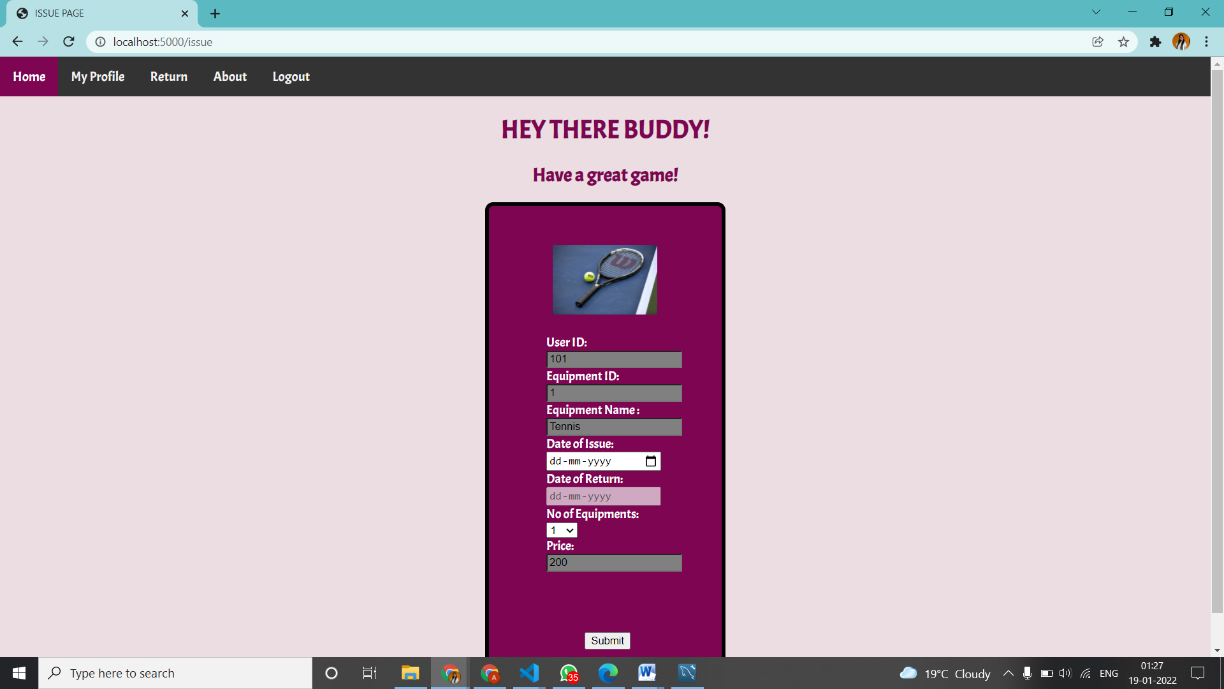


Fig 5.4 Issue Page

* The values for user ID, equipment ID and equipment name were previously stored in the database and fetched onto this page
* The user can select a date of issue using which the date of return is calculated and displayed as a month from the date of issue.
* The user then selects the number of equipments that they would be renting.
* The price is calculated accordingly and displayed.
* Upon clicking the submit button, the user is redirected to the equipment page.

**5.5 Billing Page**

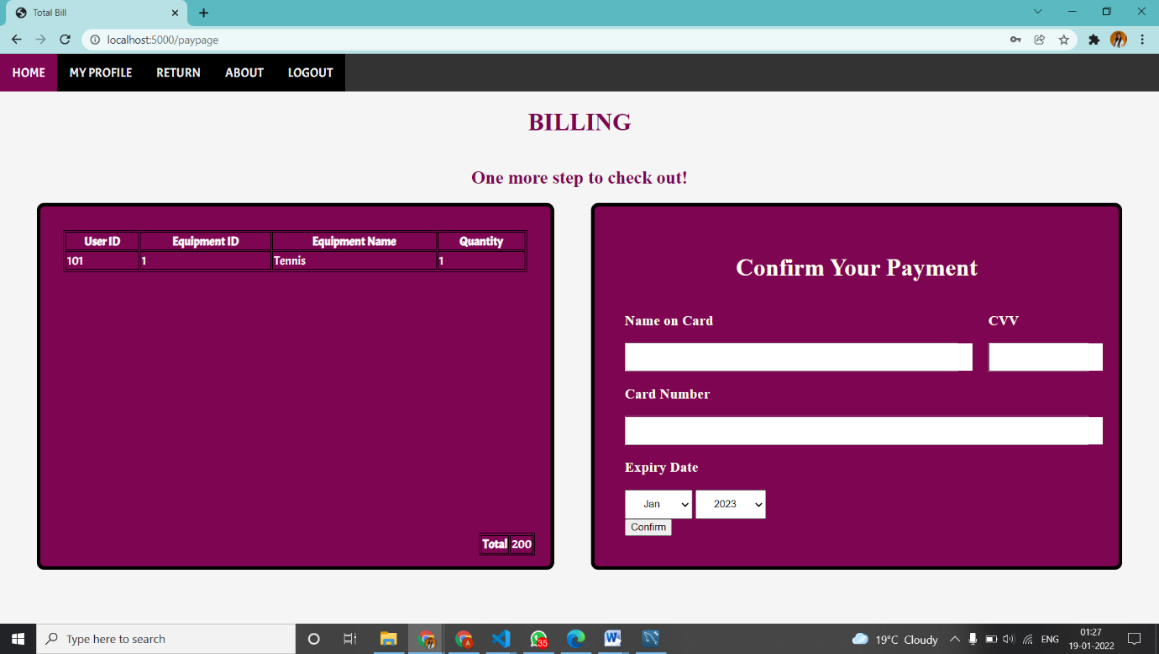
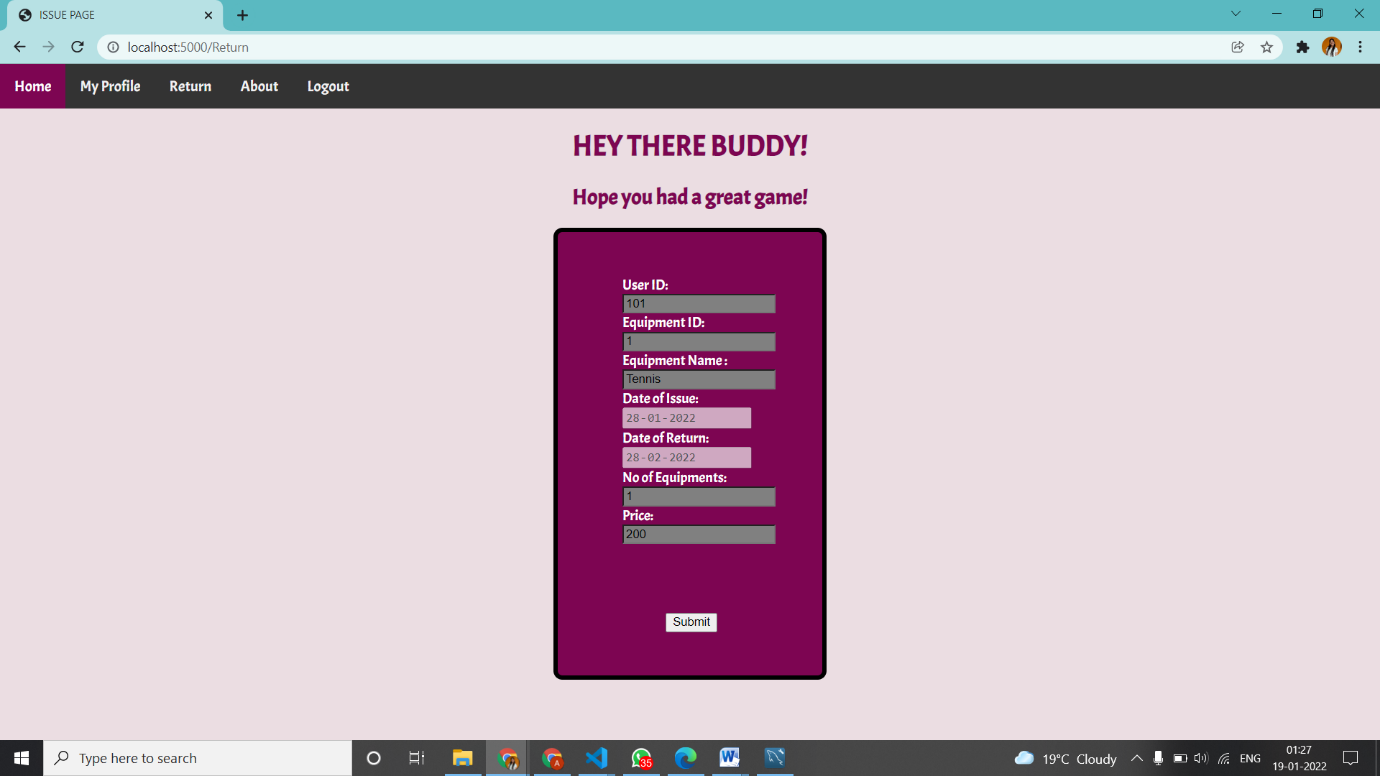
****

Fig 5.5 Billing Page

* The bill is displayed using the data stored in the backend that is fetched onto this page.
* The user can make the payment by providing his card details.
* On clicking the confirm button the user is redirected to the home page.

**5.6 Return page**

****Fig 5.6 Return Page

* All the details for this page has been fetched from the database and displayed

**5.7 My Profile Page**

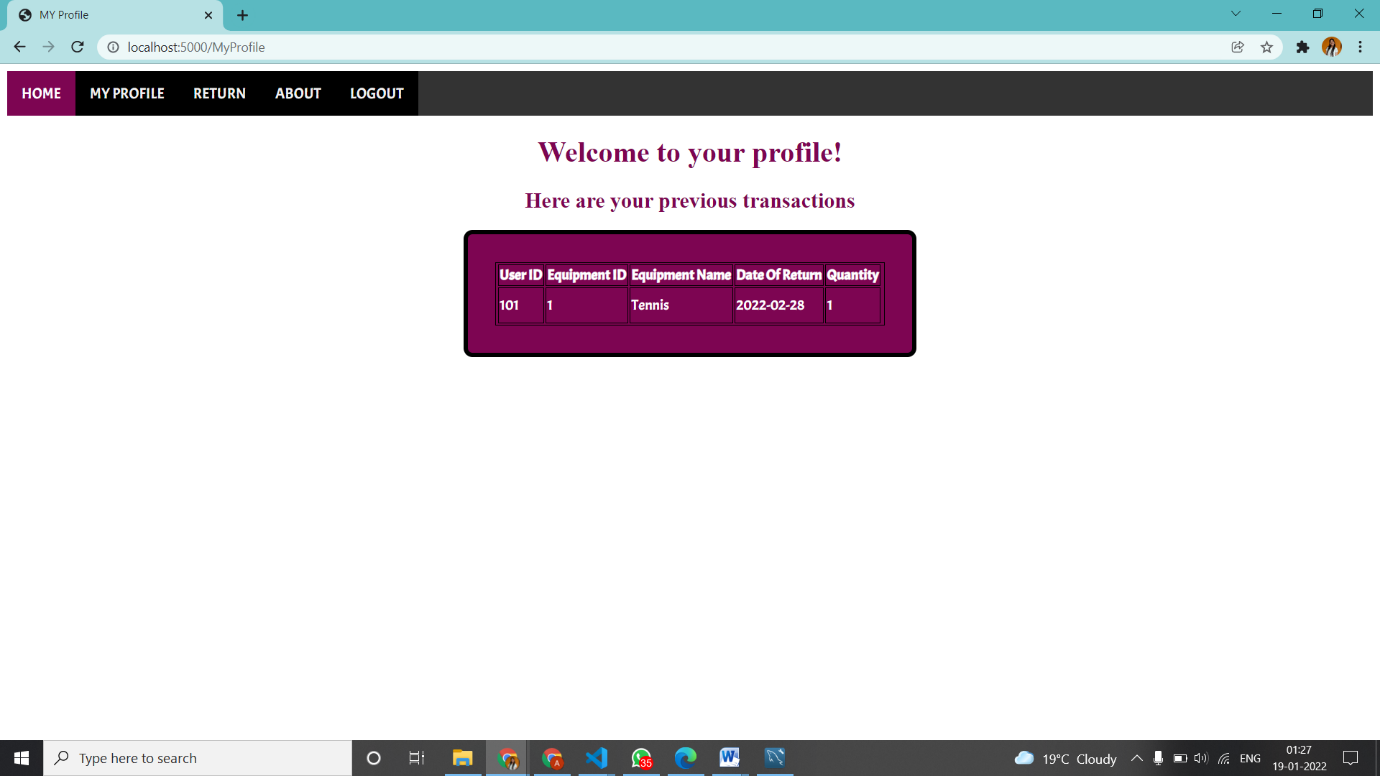
****

Fig 5.7 My profile Page

* This page displays all the previous sports gears rented out by the user

**5.8 About us page**

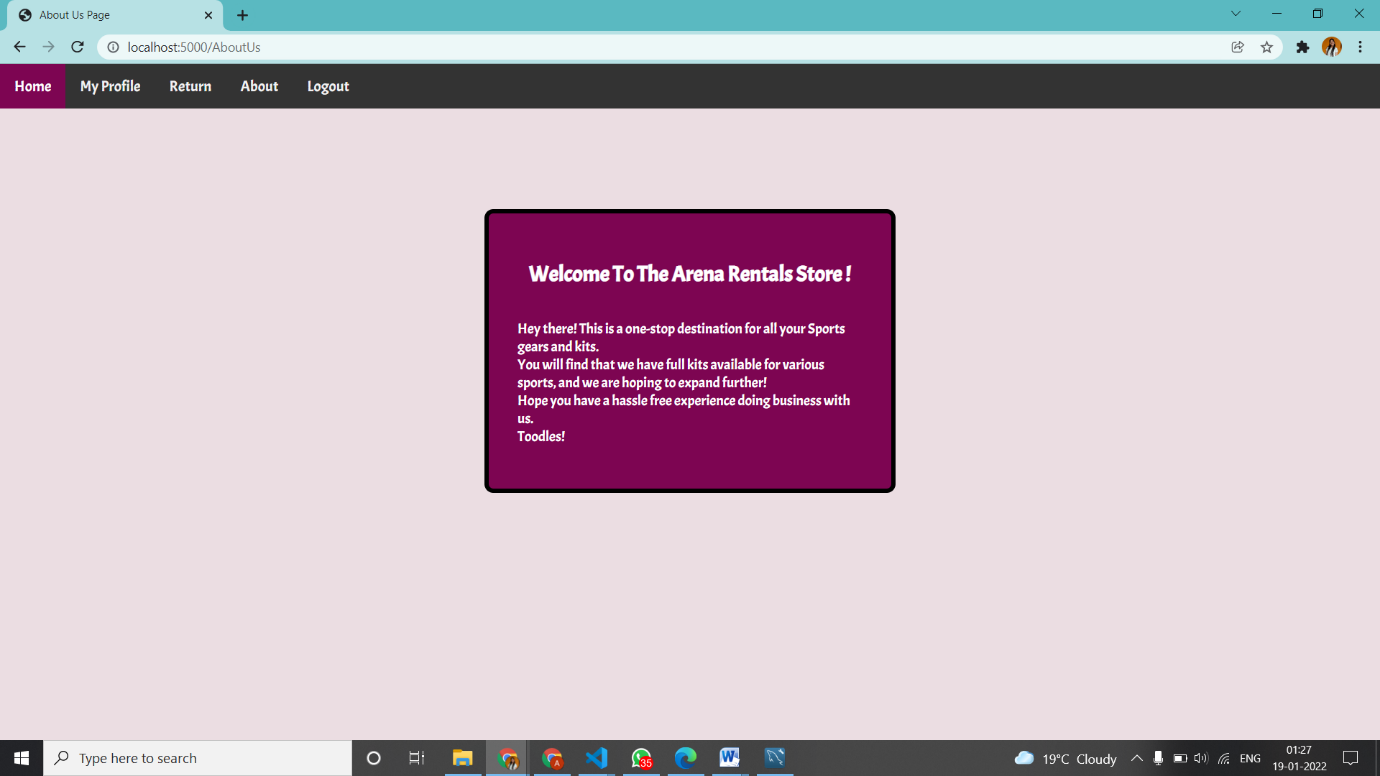
****

Fig 5.8 About us Page

**CONCLUSION & FUTURE ENHANCEMENT**

**Conclusion**

Arena Rental Stores is a web application for renting various sports equipment. This project has some features implemented using front end tools such as html, css and JavaScript and backend tools such as nodejs and sequelize for the storage of data in the database. As seen above, a user can signup, login, issue and return sports equipment according to their need. This web application is a small scale implementation of what one day might be a system used by everyone for their convenience.

**BIBLIOGRAPHY**

**Web References**

[1] https://www.w3schools.in/dbms/intro/

[2] https://beginnersbook.com/2015/04/dbms-introduction/

[3] https://getbootstrap.com/

[4] https://developer.mozilla.org/en-US/

[5] https://www.npmjs.com

[6] https://ejs.co/#docs

[7] https://expressjs.com/

[8] https://sequelize.org/master/

[9] https://www.google.com/

**Book References**

[1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson

[2] Ramakrishnan and Gehrke, “Database management systems”,3rd Edition McGraw Hill 2014

[3] SilberschatzKorth and Sudharshan, “Database System Concepts”, 6th Edition Mc-Graw Hill, 2013.