



A
Project Synopsis
On:-

E-Medicare

WORK CARRIED OUT BY-

- 1. Rushika R***
- 2. Sheetal Dhamannavar***
- 3. Abhijeet Kumar Singh***
- 4. Aditi Gawade***
- 5. Malanabi Kammar***
- 6. Busa Hanuman***
- 7. Achut M Hublikar***

INDEX CONTENT

1. INTRODUCTION	
2. AIM AND OBJECT.....	
3. HARDWARE AND SOFTWARE REQUIREMENTS	
4. MODULE DESCRIPTION.....	
5. ER-DIAGRAM	
6.FORMS	
7.CODING.....	
8. SCREENSHOTS.....	
9.CONCLUSION	

1.Introduction to Medicines:-

Medicine is the science and practice of caring for a patient, managing the diagnosis, prognosis, prevention, treatment, palliation of their injury or disease, and promoting their health. Medicine encompasses a variety of health care practices evolved to maintain and restore health by the prevention and treatment of illness. Contemporary medicine applies biomedical sciences, biomedical research, genetics, and medical technology to diagnose, treat, and prevent injury and disease, typically through pharmaceuticals or surgery, but also through therapies as diverse as psychotherapy, external splints and traction, medical devices, biologics, and ionizing radiation, amongst others

2.Aims and Objectives:-

Our vision is to be a leader in collaborative research and education in the quality use of medicines and improved health outcomes.

Our objectives are:

- To be recognised nationally and internationally as a leader in improving medication outcomes and pharmacy practice research and education.
- To establish relationships with key individuals and organisations to improve the quality use of medicines and health outcomes.
- To provide quality research and education that can add to the quality of life of members of the community.
- To deliver research findings that translate to cost savings for the community.

Online pharmacies offer **better pricing than offline stores, with increased access, lower transaction and product costs, convenience and greater anonymity for consumers**. They offer accessibility to people with limited mobility and people in remote areas.

The objective of e-pharmacy is **to deliver medicine at an affordable cost to all places**.

1. **HARDWARE AND SOFTWARE REQUIREMENTS:**

- Hardware requirements:

Processor-Intel Core
i3 RAM-4GB or
greater Hard disk-
40GB or more.

- Software requirements:

Web Browser- Google Chrome,
Microsoft Edge Operating System-
Windows
Frontend-Angular10, HTML5, CSS3, Bootstrap
Middleware- Springboot, JPA, Hibernate, JUnit
Backend-Oracle
Server-Tomcat Server.

MODULE DESCRIPTION:-

Medicare management system allows admin to manage daily Medicines record keeping work. The main purpose of Medicare management project is to eliminate difficulties of manual Medical record keeping. Medicare agency needs to keep track of customer payment and the Medicines. This is very easy to manage with the use of this application.

It has Following Modules.

- 1. User Module**
- 2.Admin Module**

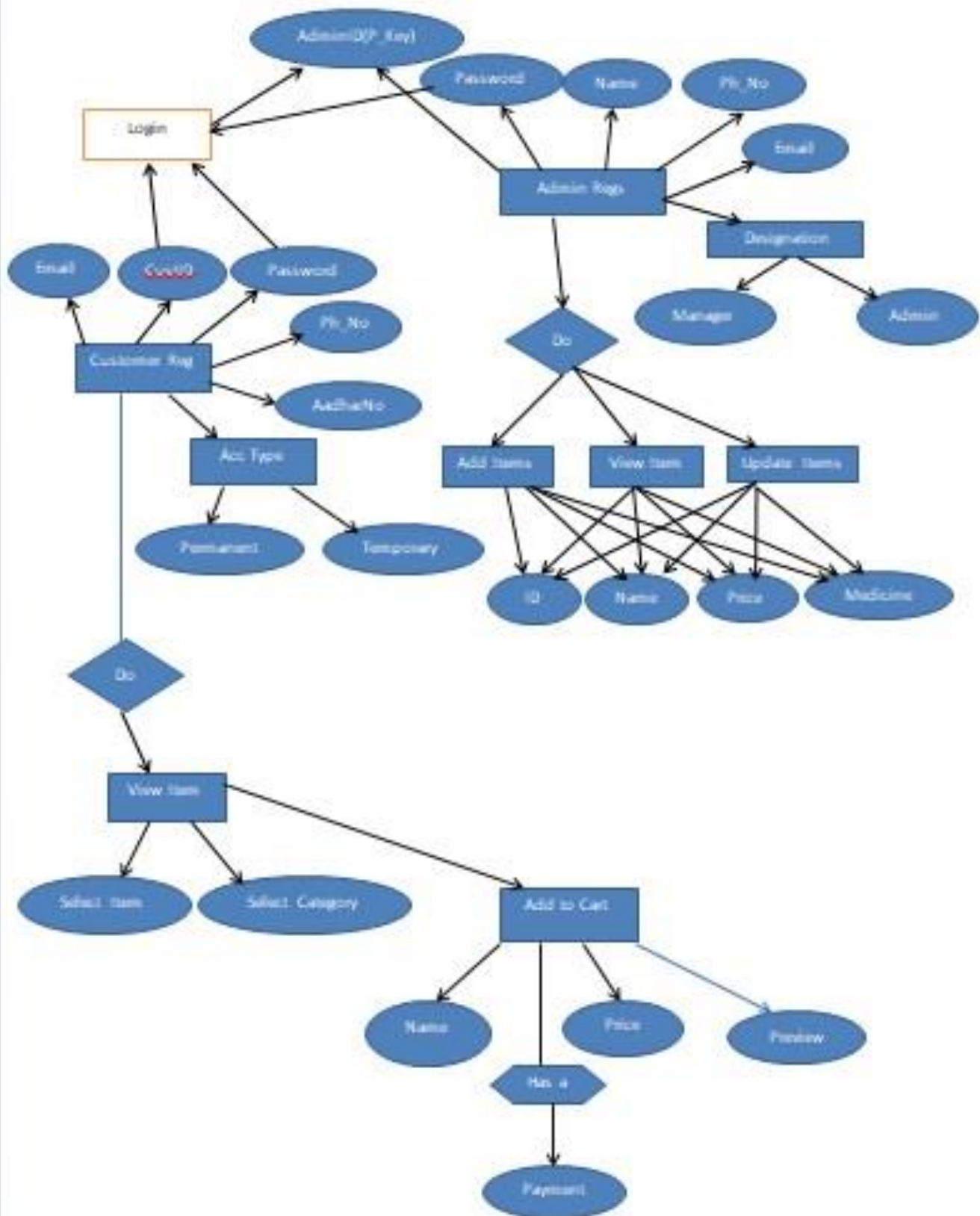
1. In the user module we have following modules

- a. User Registration
- b. User login
- c .payment
- d .we can view the medicines

2.in the Admin module we have following modules

- a. Admin Registration
- b. Admin login
- c. We can add the items
- d. we can view the Stock items

ER DIAGRAM



Forms:1

User Registration Form:

Name:	<input type="text"/>
Phone No:	<input type="text"/>
Enter Email:	<input type="text"/>
Adhar Number:	<input type="text"/>
Account Type:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Register"/>	

Forms:2

Admin Registration Form:

Name:	
Phone Number:	
Enter Email:	
Enter Adhar:	
Account Type:	
Password:	
	<input type="button" value="Register"/>

Forms:3

Login Form:

Type User:	<input type="text"/>
Enter Id:	<input type="text"/>
Enter Password:	<input type="password"/>
Forgot Password:	<input type="text"/>
<input type="button" value="LOGIN"/>	

Forms:4

VIEW ITEMS

SELECT CATEGORY:

SELECT PRODUCT :

YOUR CART ITEMS

NAME:

PRICE:

PREVIEW:

Forms:5

PAYMENT PAGE

FIRST NAME :

USER NAME :

EMAIL(OPTIONAL):

ADDRESS :

ADDRESS 2(OPTIONAL):

COUNTRY: STATE: ZIP:

PAYMENT

- ☐ CREDIT CARD
- ☐ DEBIT CARD

NAME ON THE CARD:	CREDIT CARD NUMBER :
<input type="text"/>	<input type="text"/>
Expiration :	CW :
<input type="text"/>	<input type="text"/>

Pay Now

FORM:6

ADD ITEMS

ID:

NAME:

PRICE

CATEGORY:

ADD

FORM:7

UPDATE ITEMS

ID:

NAME:

PRICE

CATEGORY:

UPDATE

FORM:8

VIEW STOCK

ID:	NAME	PRICE	MEDICINE
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

CODING:

Home:

```
<app-header></app-header>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-  
scale=1">
```

```
<link rel="stylesheet"  
href="https://www.w3schools.com/w3css/4/w3.css">
```

```
<link rel="stylesheet"  
href="https://fonts.googleapis.com/css?family=Inconsolata">
```

```
<body>
```

```
<br>
```

```
<br>
```

```
<h1>A Single Tablet Can Save Our Life</h1>
```

```
<h2>Life changes!! By Choosing A Good Medicines</h2>
```

```
<div class="container">
```

```
<a class="float-center playBtn" href="aboutus"><span>KNOW  
MORE</span></a>
```

```
</div>
```

MODULE.TS

```
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http';
import { CustomerRegisterComponent } from './customer-register/customer-register.component';
import { EmployeeRegisterComponent } from './employee-register/employee-register.component';
import { FormsModule } from '@angular/forms';
import { LoginComponent } from './login/login.component';
import { HeaderComponent } from './header/header.component';
import { WelcomeComponent } from './welcome/welcome.component';
import { WelcomeemployeeComponent } from './welcomeemployee/welcomeemployee.component';

import { HomeComponent } from './home/home.component';
import { NgModule } from '@angular/core';
```



```
import { AboutusComponent } from './aboutus/aboutus.component';
import { FooterComponent } from './footer/footer.component';
import { WelcomeheaderComponent } from './welcomeheader/welcomeheader.component';
import { AdditemComponent } from './additem/additem.component';
import { PaymentComponent } from './payment/payment.component';
import { SuccessComponent } from './success/success.component';
import { AdminstockComponent } from './adminstock/adminstock.component';
import { AdminstockUpdateComponent } from './adminstock-update/adminstock-update.component';
import { AdminstockViewComponent } from './adminstock-view/adminstock-view.component';
import { ViewitemComponent } from './viewitem/viewitem.component';
```

```
@NgModule({
  declarations: [
    AppComponent,
    CustomerRegisterComponent,
    EmployeeRegisterComponent,
    LoginComponent,
    HeaderComponent,
    WelcomeComponent,
    WelcomeemployeeComponent
```

```

    HomeComponent,
    AboutusComponent,
    FooterComponent,
    WelcomeheaderComponent,
    AdditemComponent,
    PaymentComponent,
    SuccessComponent,
    AdminstockComponent,
    AdminstockUpdateComponent,
    AdminstockViewComponent,
    ViewitemComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModuleModule,
    HttpClientModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

ROUTING.TS

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { AboutusComponent } from './aboutus/aboutus.component';
import { CustomerRegisterComponent } from './customer-register/customer-register.component';
import { EmployeeRegisterComponent } from './employee-register/employee-register.component';
import { HomeComponent } from './home/home.component';
import { LoginComponent } from './login/login.component';
import { WelcomeComponent } from './welcome/welcome.component';
import { WelcomeemployeeComponent } from './welcomeemployee/welcomeemployee.component';
import { AdditemComponent } from './additem/additem.component';
import { PaymentComponent } from './payment/payment.component';
import { SuccessComponent } from './success/success.component';
import { AdminstockComponent } from './adminstock/adminstock.component';
import { AdminstockUpdateComponent } from './adminstock-update/adminstock-update.component';
import { AdminstockViewComponent } from './adminstock-view/adminstock-view.component';

```

```

const routes: Routes = [
  { path: '', component: LoginComponent},
  { path: 'addemployee', component: EmployeeRegisterComponent},
  { path: 'addcustomer', component: CustomerRegisterComponent},
  { path: 'welcome/:userID', component: WelcomeComponent},
  { path: 'welcomeemployee/:userID', component: WelcomeemployeeComponent},

  { path: 'home', component: HomeComponent },
  { path: 'aboutus', component: AboutusComponent},
  { path: 'additem', component: AdditemComponent},
  { path: 'payment', component: PaymentComponent},
  { path: 'success', component: SuccessComponent},
  { path: 'adminstock-view', component: AdminstockViewComponent},
  { path: 'adminstock-update', component: AdminstockUpdateComponent},
  { path: 'adminstock', component: AdminstockComponent},
  {path: 'edit/:id', component: AdminstockUpdateComponent},

];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModuleModule { }

```

CUSTOMER.JAVA

```

package in.bank.project.model;

import javax.persistence.Entity;
import javax.persistence.Id;

import lombok.Data;
@Data
@Entity
public class Customer {
    public Customer() {
        super();
    }
    @Id
    public String customerid;
    public String name;
    public String phone;
    public String email;
    public String password;
    public String adhar;
}

```

```

        public Customer(String customerid, String name, String phone, String email, String
password, String adhar,
                        String signin) {
            super();
            this.customerid = customerid;
            this.name = name;
            this.phone = phone;
            this.email = email;
            this.password = password;
            this.adhar = adhar;
            this.signin = signin;
        }
        public String getCustomerid() {
            return customerid;
        }
        public void setCustomerid(String customerid) {
            this.customerid = customerid;
        }
        public String getName() {
            return name;
        }
        public void setName(String name) {
            this.name = name;
        }
        public String getPhone() {
            return phone;
        }
        public void setPhone(String phone) {
            this.phone = phone;
        }
        public String getEmail() {
            return email;
        }
        public void setEmail(String email) {
            this.email = email;
        }
        public String getPassword() {
            return password;
        }
        public void setPassword(String password) {
            this.password = password;
        }
        public String getAdhar() {
            return adhar;
        }
        public void setAdhar(String adhar) {
            this.adhar = adhar;
        }
        public String getSignin() {
            return signin;
        }

```

```

        this.signin = signin;
    }

}

```

EMPLOYEE.JAVA

```

package in.bank.project.model;

import javax.persistence.Entity;
import javax.persistence.Id;

import lombok.Data;
@Data
@Entity
public class Customer {
    public Customer() {
        super();
    }
    @Id
    public String customerid;
    public String name;
    public String phone;
    public String email;
    public String password;
    public String adhar;
    public String signin;
    public Customer(String customerid, String name, String phone, String email, String
password, String adhar,
        String signin) {
        super();
        this.customerid = customerid;
        this.name = name;
        this.phone = phone;
        this.email = email;
        this.password = password;
        this.adhar = adhar;
        this.signin = signin;
    }
    public String getCustomerid() {
        return customerid;
    }
    public void setCustomerid(String customerid) {
        this.customerid = customerid;
    }
    public String getName() {
        return name;
    }
}

```

```

        this.name = name;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getAdhar() {
        return adhar;
    }
    public void setAdhar(String adhar) {
        this.adhar = adhar;
    }
    public String getSignin() {
        return signin;
    }
    public void setAccounttype(String signin) {
        this.signin = signin;
    }
}
package in.bank.project.model;

```

```

import javax.persistence.Entity;
import javax.persistence.Id;

```

```

import lombok.Data;

```

```

@Data
@Entity
public class Employee {

```

```

public String name;
public String phone;
public String email;
public String password;
public String employeeetype;

public String getEmployeeid() {
    return employeeid;
}
public void setEmployeeid(String employeeid) {
    this.employeeid = employeeid;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getPhone() {
    return phone;
}
public void setPhone(String phone) {
    this.phone = phone;
}
public String getEmail() {
    return email;
}
public void setEmail(String email) {
    this.email = email;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}

public String getEmployeeetype() {
    return employeeetype;
}
public void setEmployeeetype(String employeeetype) {
    this.employeeetype = employeeetype;
}
public Employee(String employeeid, String name, String phone, String email, String password,
    String employeeetype) {
    super();
    this.employeeid = employeeid;
    this.name = name;
    this.phone = phone;
    this.email = email;
    this.password = password;
}

```

```

        this.employeetype = employeetype;
    }
    public Employee() {
        super();
    }
}

```

STOCKS.JAVA

```

package in.bank.project.model;
import javax.persistence.Entity;
import javax.persistence.Id;

import lombok.Data;
@Data
@Entity
public class Stock {

    @Id
    public String id;
    public String name;
    public String price;
    public String medicine;

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }
    public String getMedicine() {
        return medicine;
    }
}

```



```
public Stock(String id, String name, String price, String medicine
    ) {
    super();
    this.id = id;
    this.name = name;
    this.price = price;
    this.medicine = medicine;
}
public Stock() {
    super();
}
```

```
}
```

Screenshots:

Home Page:



Admin Registration:

■ E-Medicare Home Registration ▾ Login AboutUs

Admin Registration

Name:

Phone Number :

Email :

Designation :

Enter Password:

Create your own password

Register



Customer Registration:

[E-Medicare](#) [Home](#) [Registration](#) [Login](#) [AboutUs](#)

Customer Registration E-medicare

Name:

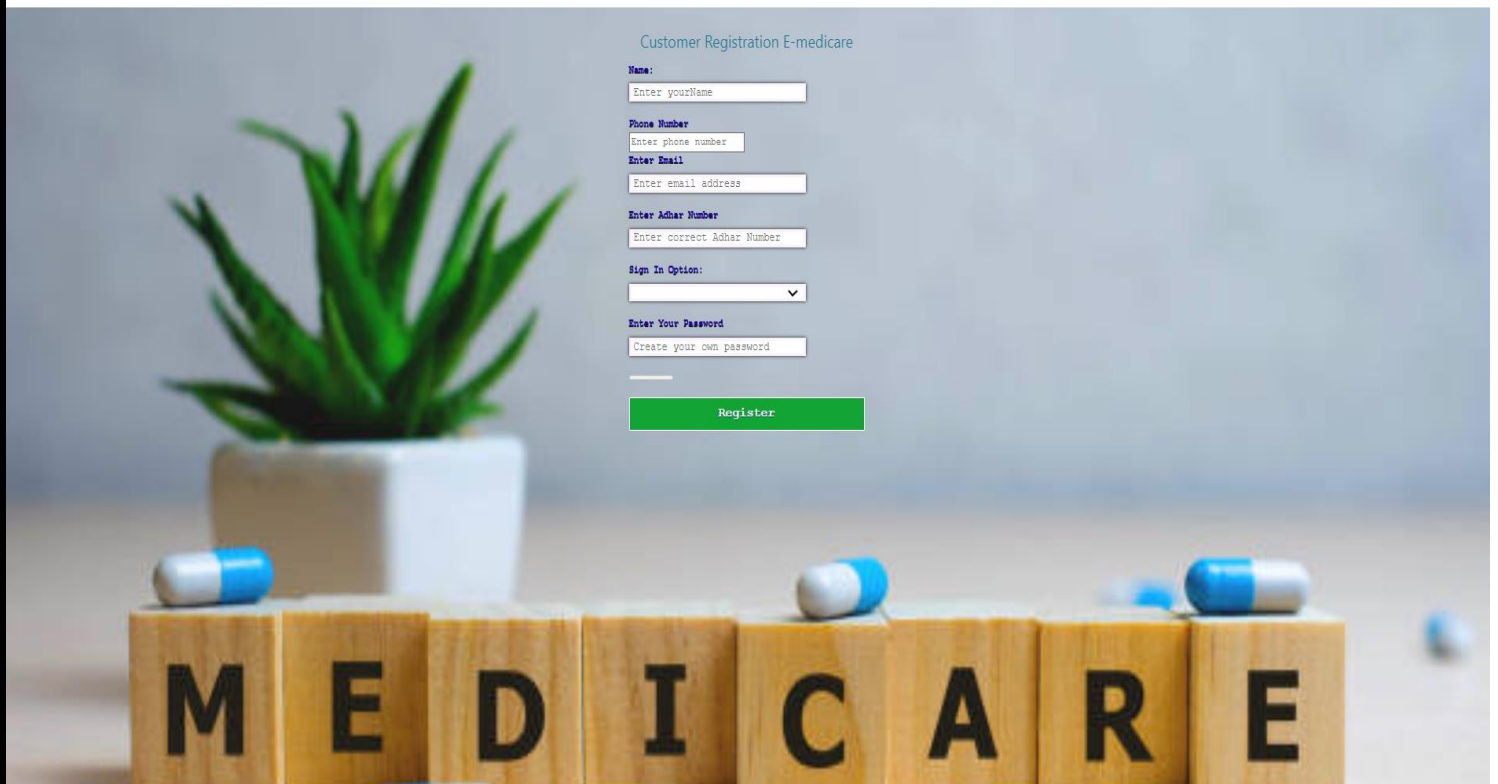
Phone Number

Enter Email

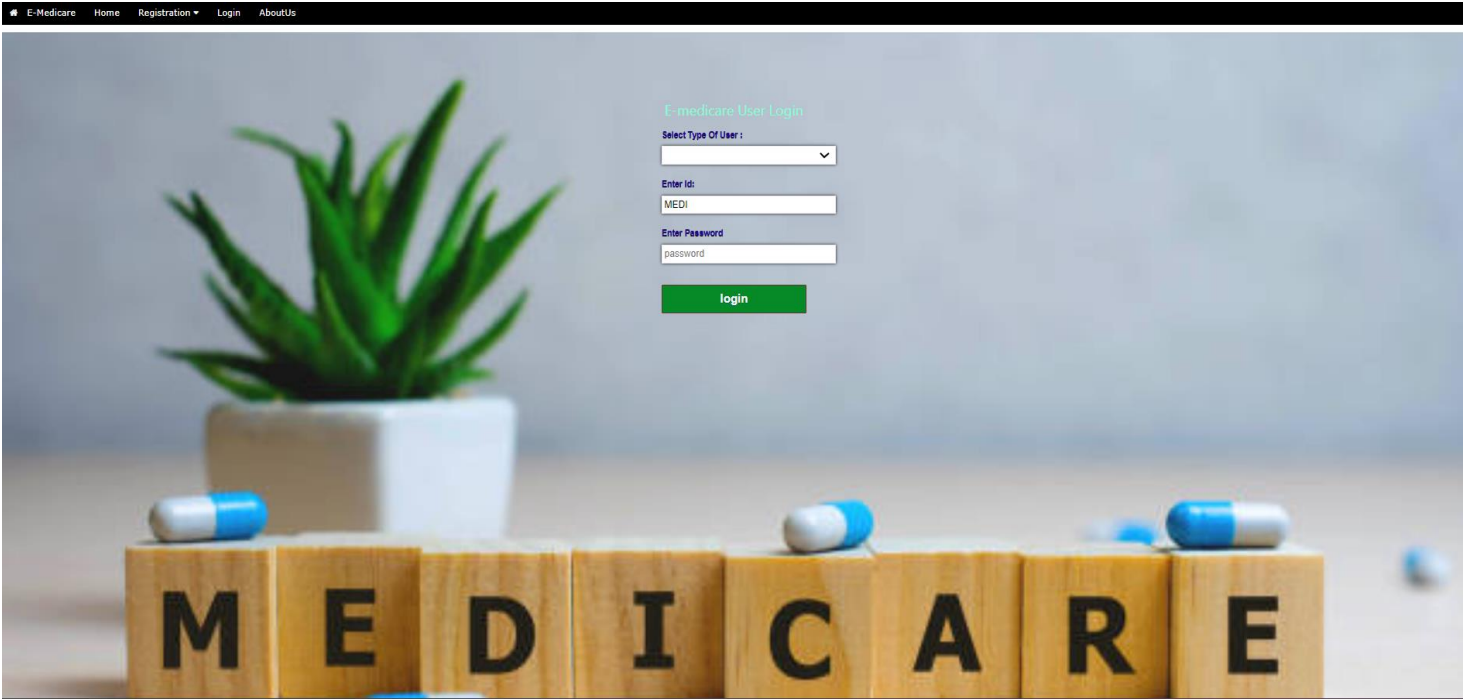
Enter Adhar Number

Sign In Option:

Enter Your Password



Login Page:



Add items Page :

E-MedicareHomeStockLogout

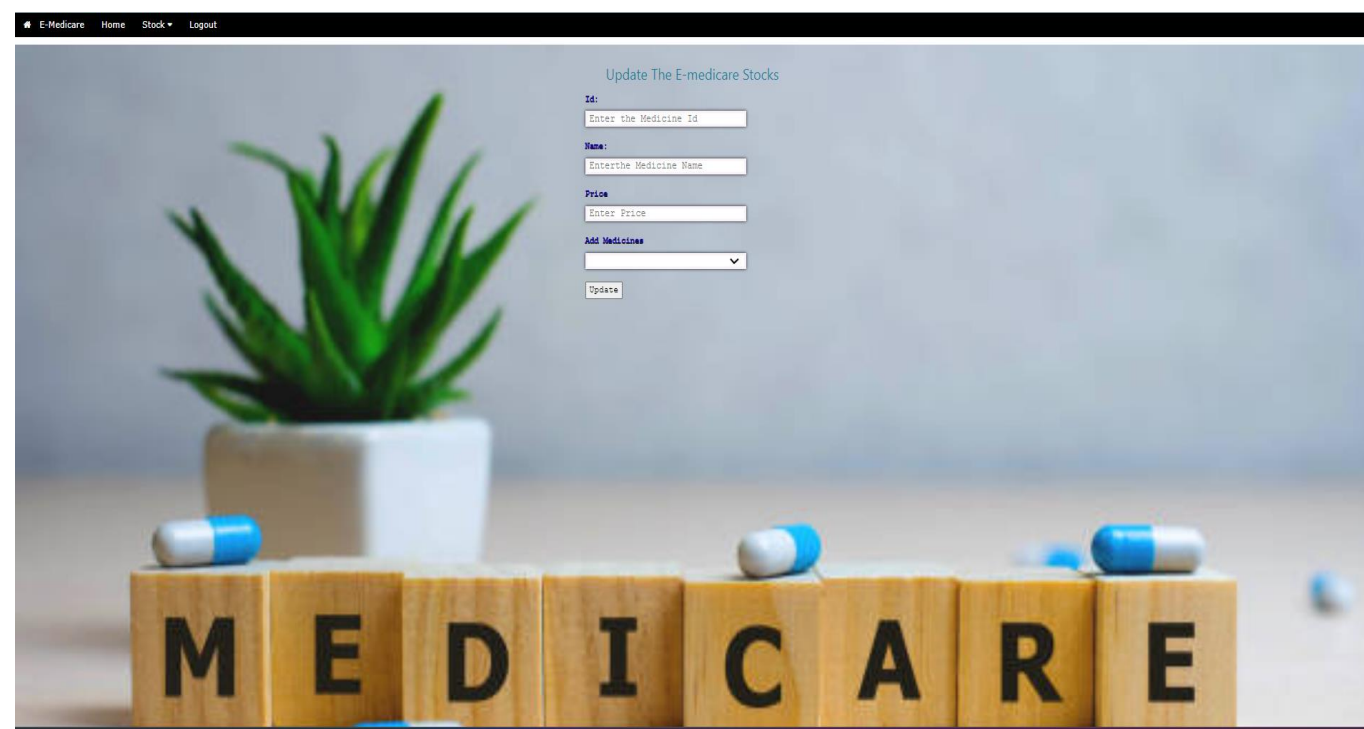
Add Items E-medicare

Add Medicines

Add

MEDICARE

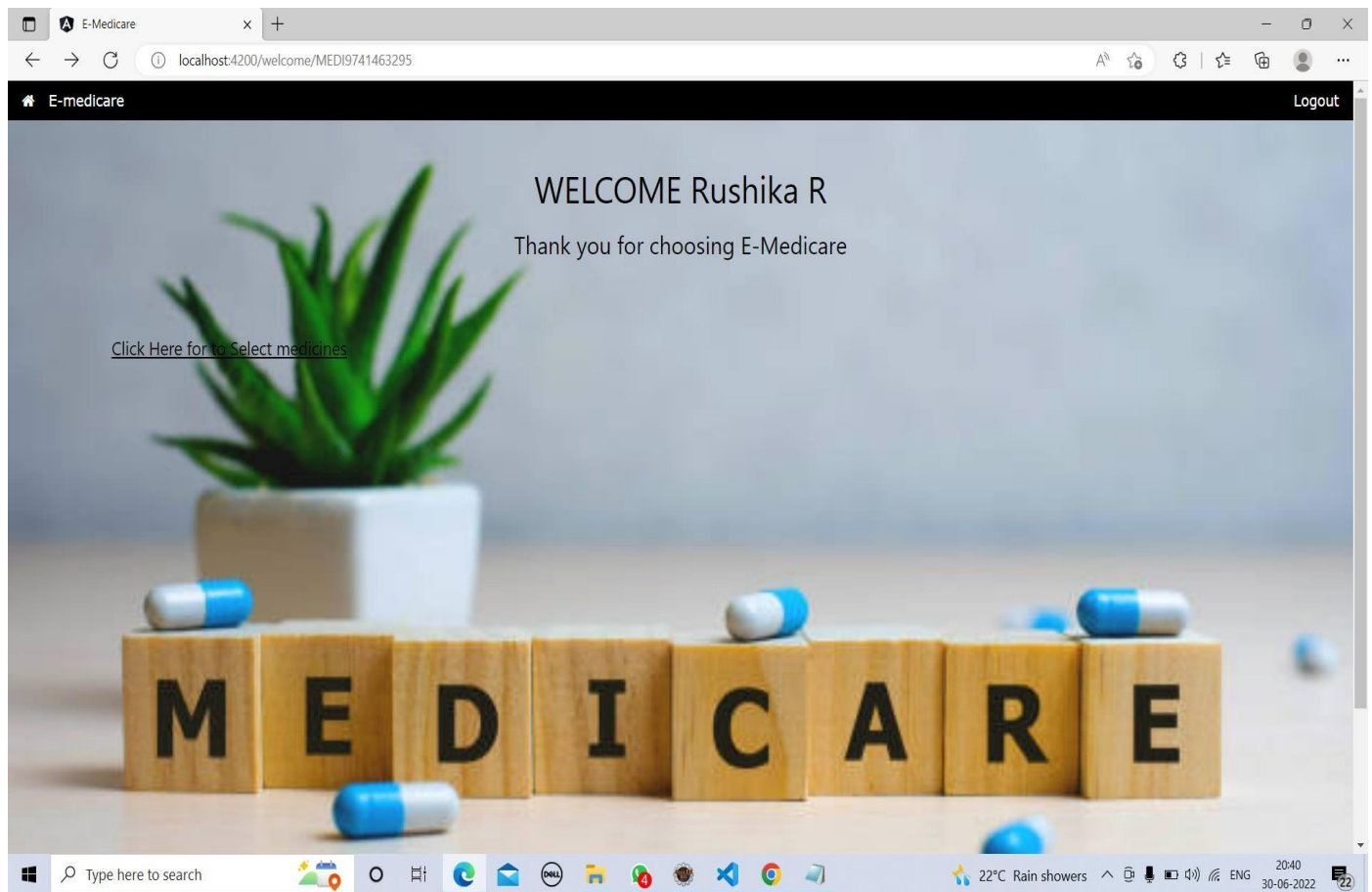
Update Page :



Stock View :



WELCOME CUSTOMER:



Add to Cart Page:

 E-medicaire

Logout

Select a Category

Select a Category

▼

Select a Product

Select a Product

▼

Preview

₹0.00

Add to Cart

[0] Your Cart Items

Payment

Follow Us

View Cart Items Page :

E-medicare

Logout

Select a Category

General

Select a Product

Losartan

Preview

Losartan

₹100.55

Add to Cart

Your Cart Items

Name	Price	Delete
Losartan	100.55	

[1] Your Cart Items

Payment

Follow Us

Payment Page :

E-medicare

Logout

INVOICE

Billing

Full Name

Name

Email

john@example.com

Address

542 W. 15th Street

City

city

State

State

PinCode

000000

Total

000

☒ Shipping address same as billing

Payment

Accepted Cards

VISA

MasterCard

Amex

Card Holder Name

Name

Credit card number

1111-2222-3333-4444

Exp Month

Month

Exp Year

year

CVV

000

Pay Now

Follow Us

f

t

G+

@

in

Page 36

About Us Page :

Home

Registration

Login

About Us

Terms & Conditions:

Medicines:

• Take the prescribed dose at the same time each day.

• Don't skip doses or take half doses to save money.

• Take medicine until it's finished or until your doctor says it's okay to stop.

Ayurveda:

• Avoid dry and spicy food items.

• Any medicine, Ayurvedic or otherwise, requires the stomach to be at the "right state of mind".

• Focus on nutrition is one of its essential components.

Payments:

• We request and warrant that if you are purchasing something from us or from Members that:


• (i) any credit information you supply is true and complete.

• (ii) charges incurred by you will be honored by your credit card company.

• (iii) You will pay the charges incurred by you at the posted price, including any applicable taxes.


OUR TEAM

Meet the team - our office staff:




Surenranath Tagore

CEO




Rebecca Flex

DIRECTOR



Jan Ringo

MEMO




Kai Ringo


MANAGER


Our Location


Contact us on 8676543210




Follow Us









Page 37

CONCLUSION

- This software has made **working in a medical store** a much easy job. This software maintains a log of **every customer bill**.
- It has **purchase stock** according to companies.
- It gives a lot of details about the **stock sell**. In short, it is a **medical inventory** that has made lives easier.
- This software is called the **medicare management system**.
- It provides every small to big details like **customer details, purchase details, sell details, bill details, etc**

Thank You