24K-0559

BSCS2H

OOP THEORY ASSIGNMENT3:

CODE:

```
//24k-0559 Bazil-Uddin-Khan
#include<iostream>
#include<fstream>
#include<string>
#include<exception>
using namespace std;

class User
{
   protected:
   string UserName;
   string DateOfJoining;
   string email;
   string phoneNumber;

   public:
   User() : UserName(""),DateOfJoining(" "),email(" "),phoneNumber(" ")
   {}

   User(string name,string date,string Email,string number) :
UserName(name),DateOfJoining(date),email(Email),phoneNumber(number)
```

```
{}

virtual  void DisplayInfo()
{
   cout << " UserS  Details Are For Verification " << endl;
   cout << "User Name is = " << UserName << endl;
   cout << "User Date Of Joining  is = " << DateOfJoining << endl;
   cout << "User Email is = " << email << endl;
   cout << "User Phone Number  is = " << phoneNumber << endl;
}

string GetName()
{
   return UserName;
}

string GetEmail()
{
   return email;
}

string GetDateOfJoining()
{
   return DateOfJoining;
}
```

```
};

class Routes
{
   private:
   string RoutesPickDown[25] =
{"Saffora","SindhBaloach","KamranChorangi","Jamalipull","Maskan","PowerHo
use","DolminMall","bardadari","Golbarq","Donisel","clifton","Bahria","Dha","
UniversityRoad","Korangi","lyaripull","Motimehal","Sheefaisal","Balochistansaj
ji","Kdm","Mehran","Lalqila","Tariqroad","Balochistanx","bax"};


   string RoutesPickUp[25] = {
      "Nazimabad", "North Karachi", "WaterPump", "Ancholi", "Liaquatabad",
      "Gulshan-e-Maymar", "AyeshaManzil", "PECHS", "GuruMandir",
"Shahra-e-Faisal",
      "Malir Halt", "Model Colony", "SafariPark", "JoharMor", "Askari-IV",
      "SafooraChowrangi", "Defence Phase 2", "SeaView", "HillPark", "Gizri",
      "ManzoorColony", "NewTown", "Khokharapar", "Landhi 89","Saddar"
   };


   double Distance[25] =
   {
     10.5, 20.3, 15.7, 8.2, 30.6,
     25.9, 12.4, 18.8, 5.5, 22.1,
     16.0, 9.7, 14.2, 27.3, 19.5,
```

```
  11.6, 24.7, 7.8, 13.9, 21.0,
  6.4, 17.2, 23.5, 28.1, 4.9
};

public:
Routes()
{}

int CheckRoute(string Routeu,string Transport)
{
   if(Transport == "Nadeem")
   {
      for(int i=0; i < 25; i++)
     {
      if(RoutesPickUp[i] == Routeu|| RoutesPickDown[i] == Routeu)
       {
          return 1;
       }
     }

     cout << " Not Found " << endl;
    return -1;
   }
   else if(Transport == "Zulfiqar")
   {
```

```
        for(int i=0; i < 25; i++)
        {
         if(RoutesPickUp[i] == Routeu || RoutesPickDown[i] == Routeu)
         {
            return 1;
         }
        }

        cout << " Not Found " << endl;
       return -1;

      }
      cout << " Not Found " << endl;
       return -1;
   }

   string GetRouteName(string RouteNumber)
   {
      for(int i =0; i < 25;i++)
      {
         if(RoutesPickUp[i] == RouteNumber || RoutesPickDown[i] ==
RouteNumber)
         {
            string val = RoutesPickUp[i];
            return(val);
```

```
        }
    }
    return " ";
}

void DisplayInfo()
{
    cout << " Routes Details Are " << endl;
    for( int y =0; y< 24;y++)
    {
        cout << RoutesPickUp[y] << "  ";
    }
    cout<< endl;
    cout << " Routes Pick Down  " << endl;
     for(int y =0; y< 24;y++)
    {
        cout << RoutesPickDown[y] << "  ";
    }
    cout << endl;
    cout << "Distances Are " << endl;
     for(int y =0; y< 24;y++)
    {
        cout << Distance[y] << "  ";
    }
    cout << endl;
```

```
    }
};

class EntityFoundNot : public exception
{
    public:
    const  char* what() const noexcept override
    {
        return " Entity Is Not Present Exception ";
    }
};

class PaymentNotExcep : public exception
{
    public:
    const  char* what() const noexcept override
    {
        return " Payment Exception Not paid  ";
    }
};

class OutOfRangeExcep : public exception
{
    public:
    const  char* what() const noexcept override
```

```
    {
        return " Index Exception Out Of Range ";
    }
};

class WrongInputException : public exception
{
    public:
    const  char* what() const noexcept override
    {
        return " Wrong Input Exception  ";
    }
};

class RouteNotFoundExeption : public exception
{
    const  char* what() const noexcept override
    {
        return " Route Not Found  Exception  ";
    }

};

class PaymentFee
{
```

```
private:
double Fees;
string CardStatus;
string AcStatus;
const int ExtraFees = 2000;

public:
PaymentFee()
{
   Fees =0;
   CardStatus = "NotPaid";
   AcStatus = "No";
}
PaymentFee(double Fee,string status)
{
   AcStatus = status;
   Fees =0;
   if(AcStatus == "Yes")
   {

      Fees = Fees + Fee;
      this->Fees = Fees + ExtraFees;
   }
   else if(AcStatus == "No")
   {
```

```
        Fees = Fee;
    }
    else
    {
        throw WrongInputException();
    }
}
int PaymentFees(double fees)
{
    Fees = fees;
    if(Fees ==  fees)
    {
        this->Fees = fees;
        this->CardStatus = "Paid";
        cout << " Payment Susssfully made " << endl;
        return 1;
    }
    else
    {
        cout << " Payment Not Recieved " << endl;
        throw PaymentNotExcep();
        return 0;
    }
}
```

```
    void DisplayInfo()
    {
        cout << " --Fees Status Is-- " << endl;
        cout << " Fees Is " << Fees << endl;
        cout << " Card Status Is " << CardStatus << endl;
    }

    string GetStatus()
    {
        return CardStatus;
    }
};

class Driver : virtual public User
{
    protected:
    string driverName;
    string TransportType;
    string LiscenceType;
    string RouteAssigned;

    public:
    Driver() : driverName(" "),TransportType(" "),LiscenceType("
"),RouteAssigned(" ")
    {}
```

```
   Driver(string name,string type,string lis,string route,string date,string
Email,string number) :
driverName(name),TransportType(type),LiscenceType(lis),RouteAssigned(route),
User(name,date,Email,number)
   {}

   void SetAttributes()
   {
      string name;
      cout << "Enter Driver Name" << endl;
      cin >> name;
      driverName = name;

      string Transporttype;
      cout << "Enter Transportype like Bus/Coaster" << endl;
      cin >> Transporttype;
      TransportType = Transporttype;

      string ltype;
      cout << "Enter Lisence like (Full/Intermediate/Learner)" << endl;

      cin >> ltype;
      LiscenceType = ltype;
      string routeassign;
```

```
        cout << "Enter RouteAssigned To You " << endl;
        cin >> routeassign;
        RouteAssigned = routeassign;
    }

    void displayDetails()
    {
        cout << " --Driver Info-- " << endl;
        cout << "Driver Name Is " << driverName << endl;
        cout << "Transporter Type Is " << TransportType << endl;
        cout << "Driver Lisence Is " << LiscenceType << endl;
        cout << "Driver Route Is " << RouteAssigned << endl;
    }

    string GetName()
    {
        return driverName;
    }
};

class Student : virtual public User
{
    private:
    string StudentId;
    string StudentName;
```

```
    string StudentContactNumber;
    string BatchNo;
    string SemesterYear;
    string RouteNumber;
    string StopName;
    static int TotalStudents;
    string CardNumber;
    int Status;
    Routes route;

    public:
    Student()
    {
        StudentId = " ";
        StudentName = " ";
        StudentContactNumber = " ";
        BatchNo = " ";
        SemesterYear = " ";
        RouteNumber = " ";
        StopName = " ";
        CardNumber =" ";
    }
    Student(string StudentId,string StudentName,string StudentContactNumber,
string BatchNo,string SemesterYear,string RouteNumber,string StopName,string
```

```
CardNumber,string date,string Email,string number) :
User(StudentName,date,Email,number),Status(-1)
  {
    this->StudentId = StudentId;
    this->StudentName = StudentName;
    this->StudentContactNumber = StudentContactNumber;
    this->BatchNo = BatchNo;
    this->SemesterYear = SemesterYear;
    this->RouteNumber = RouteNumber;
    this->StopName = StopName;
    this->CardNumber = CardNumber;
  }

  void SetAttributes(string StudentId,string StudentName,string
StudentContactNumber, string BatchNo,string SemesterYear,string
RouteNumber,string StopName,string CardNumber)
  {
    this->StudentId = StudentId;
    this->StudentName = StudentName;
    this->StudentContactNumber = StudentContactNumber;
    this->BatchNo = BatchNo;
    this->SemesterYear = SemesterYear;
    this->RouteNumber = RouteNumber;
    this->StopName = StopName;
    this->CardNumber = CardNumber;
```

```
}

void StudentRegisteration()
{
   string StudentId;
   cout << " Enter Your Fast Id like this(21k-0678) " << endl;
   cin >> StudentId;

   string StudentName;
   cout << " Enter Your Student Name " << endl;
   cin.ignore();
   getline(cin,StudentName);

   string StudentContactNumber;
   cout << " Enter Student Contact Number " << endl;

   cin >> StudentContactNumber;
   string BatchNo;
   cout << " Enter Batch No like 2022 " << endl;
   cin >> BatchNo;

   string SemesterYear;
   cout << " Enter Semester(Fall/Spring)Year like Fall2024 ";
   cin >> SemesterYear;
```

```
    cout << "Routes Location Display" << endl;
    route.DisplayInfo();

    string RouteNumber;
    cout << " Enter Route Name Like (UniversityRoad Etc) from above highlighted Routes " << endl;
    cin >> RouteNumber;

    string AcStatus;
    cout << "Tell In Yes/No If You Need Ac Or Not " << endl;
    cin >> AcStatus;

    PaymentFee payment(21200,AcStatus);

    string transport;
    cout << "Enter Transport name Nadeem/Zulfiqar " << endl;
    cin >> transport;
    int Routenumber = route.CheckRoute(RouteNumber,transport);
    if(Routenumber != -1)
    {
       this->RouteNumber = RouteNumber;
       string RouteName = route.GetRouteName(RouteNumber);
       if(RouteName != " " && payment.GetStatus() != "NotPaid")
       {
          this->StopName = RouteName;
```

```
            cout << " Succesfully Registered " << endl;
            CardNumber = ("0"+ to_string(TotalStudents));

SetAttributes(StudentId,StudentName,StudentContactNumber,BatchNo,Semest
erYear,RouteNumber,RouteName,CardNumber);
            TotalStudents++;
            Status = 1;
        }
        else
        {
            cout << " Failed To Register " << endl;
            throw PaymentNotExcep();
        }
    }
    else
    {
        cout << " Route Not Found " << endl;
        throw RouteNotFoundExeption();
    }
}

string GetContactNumber() const
{
    return StudentContactNumber;
}
```

```
string GetName() const
{
    return StudentName;
}

static int GetTotalStudents()
{
    return TotalStudents;
}

string GetUserId() const
{
    return StudentId;
}
```

```
void SaveStudentToFile(Student& student)
{
  ofstream file("students.txt", ios::app);
  if (file.is_open())
  {
    cout << "file Sucesfully Opened" << endl;
    file << "Name: " << StudentName << endl;
```

```
    file << " Id: " << StudentId << endl;
    file << "Contact: " << StudentContactNumber << endl;
    file << "Route: " << RouteNumber << endl;
    file.close();
  }
  else
  {
    cerr << "Sorry!. Unable to open file for writing!" << endl;
   }
}

int getstatus()
{
    return Status;
}

void DisplayInfo()
{
    cout << "--Student Details  Is--" << endl;
    cout << " StudentId  Is  " << StudentId <<  endl;
    cout << " Student Name is " << StudentName << endl;
    cout << " Student ContactNumber  Is  "<< StudentContactNumber << endl;
   cout << " Student BatchNo  Is  " << BatchNo << endl;
   cout << " Student SemesterYear  Is  " << SemesterYear << endl;
```

```
    cout <<"Route Number Is " <<  RouteNumber << "And Stop Name Is "  << StopName << " And Card Number Is " << CardNumber << endl;
  }
};

int Student :: TotalStudents =0;

class Faculty : virtual public User
{
  private:
  double MonthlyFees;
  double Salary;
  string RouteNumber;
  int Status;
  Routes route;

  public:
  Faculty(){}

  Faculty(string name,string date,string Email,string number,double fees,string Routenumber,double salary) :Status(-1), MonthlyFees(fees),User(name,date,Email,number),RouteNumber(Routenumber),Salary(salary)
  {}

  void DisplayInfo()
```

```
{
   cout << endl;
   cout << " __Teacher Details Is_ _" << endl;
   cout << " Teacher Name Is " << UserName << endl;
   cout << " Teacher Email Is " << email << endl;
   cout << " Teacher Date Of 📅 Joining Is " << DateOfJoining << endl;
   cout << " Teacher Fees Is " << MonthlyFees << endl;
   cout << " Teacher Number Is " << phoneNumber << endl;
}

void TeacherRouteRegisteration()
{
   string AcStatus;
   cout << "Tell In Yes/No If You Need Ac Or Not " << endl;
   cin >> AcStatus;
   PaymentFee payment(MonthlyFees,AcStatus);
   int result = payment.PaymentFees(MonthlyFees);
   if(result !=0)
   {
      string transport;
      cout << "Enter Transport name Nadeem/Zulfiqar " << endl;
      cin >> transport;
      int Routenumber = route.CheckRoute(RouteNumber,transport);
      if(Routenumber != -1)
      {
```

```
            string RouteName = route.GetRouteName(RouteNumber);
            if(RouteName != " " && payment.GetStatus() != "NotPaid")
            {
              cout << " Teacher Succesfully Registered " << endl;
              Status =1;
            }
            else
            {
              cout << "Teacher Failed To Register " << endl;
              throw PaymentNotExcep();
            }
          }
          else
          {
           cout << " Route Not Found " << endl;
           throw RouteNotFoundExeption();
           }
        }

        else
        {
           cout << " Sorry Teacher But You Cannot Avail Transport Please Clear Fees "
<< endl;
        }
    }
```

```
string getname()
{
   return RouteNumber;
}

int getstatus()
{
   return Status;
}

void SaveFacultyToFile(Faculty& faculty)
{
  ofstream file("faculty.txt",ios::app);
  if(file.is_open())
  {
    cout << "Name is " << UserName << endl;
    file << "Route: " << RouteNumber << endl;
    file << "Email : " << email << endl;
    file.close();
  }
  else
  {
    cout << "Sorry!. Unable to open file for writing!" << endl;
   }
```

```
    }
};

class Vehicle
{
    protected:
    string AssignedRoute;
    string AssignedDriver;
    int FacultySeats;
    int StudentSeats;
    Driver driver;



    public:
    Vehicle() : AssignedRoute(" "),AssignedDriver("
"),FacultySeats(),StudentSeats()
    {}


    Vehicle(string rou,string dr,int fse,int se) :
AssignedRoute(rou),AssignedDriver(dr),FacultySeats(fse),StudentSeats(se)
    {}


     void displayinfo();
```

```
};

class SeatsNotAvailableException : public exception
{
   const  char* what() const noexcept override
   {
      return " Unsuccesfull Because seats not available Exception ";
   }

};

class Bus : public Vehicle
{
   string BusName;
   string DriverName;
   string model;
   const int Seats=52;

   public:
   Bus() : BusName(" "),DriverName(" "),model(" ")
   {}

   Bus(string bname,string dname,string mo,string rou,string dr,int fse,int se) :
BusName(bname),DriverName(dname),model(mo),Vehicle(rou,dr,fse,se)
   {
```

```
    if(se > Seats)
    {
       throw SeatsNotAvailableException();
    }
  }

  void SetAttributes(string bname,string dname,string mdel)
  {
    BusName = bname;
    model = mdel;
    DriverName = dname;
    FacultySeats =  15;
    StudentSeats = 52-15 ;
  }

  void displayInfo(string name)
  {
    cout << "--Bus Info--" << endl;
    cout << " BusName is " << BusName << endl;
    cout << " Driver Name is " << DriverName << endl;
    cout << " Model is " << model << endl;
    cout << " Bus Contains : " << FacultySeats << " Seats for faculty " << endl;
    cout << " Bus Contains : " << StudentSeats << " Seats for student " << endl;
    cout << " Bus : " << BusName << " Assigned Succesfully To " << name <<
endl;
```

```
    }

    void SafeBusDataToFile()
    {
        ofstream file("bus.txt",ios::app);
        if(!file)
        {
            cerr << "Camt Open File" << endl;
            return ;
        }

        file << "Bus name: "<< BusName << endl;
        file << " Driver Name: " << DriverName << endl;
        file << " Model is " << model << endl;

        file.close();
    }
};

class Coaster : public Vehicle
{
    string CoasName;
    string DriverName;
    string model;
    const int seats = 32;
```

```
public:
Coaster() : CoasName(" "),DriverName(" "),model(" ")
{}

Coaster(string bname,string dname,string mo,string rou,string dr,int fse,int se)
: CoasName(bname),DriverName(dname),model(mo),Vehicle( rou,dr,fse, se)
{
   if(seats < se)
   {
      throw SeatsNotAvailableException();
   }
}

void SetAttributes(string bname,string dname,string mdel)
{
   CoasName = bname;
   model = mdel;
   DriverName = dname;
   FacultySeats = 10;
   StudentSeats = 32-10;

}

void displayInfo(string name)
```

```
  {
     cout << "--Coaster Info--"<<endl;
     cout << " CoasterName is " << CoasName << endl;
     cout << " Driver Name is " << DriverName << endl;
     cout << " Model is " << model << endl;
     cout << " Coaster Contains : " << 32-FacultySeats << " Seats for faculty " <<
endl;
     cout << " Coaster Contains : " << StudentSeats << " Seats for student " <<
endl;
     cout << " Coaster : " << CoasName << " Assigned Succesfully To " << name <<
endl;
  }

  void SafeCoasterDataToFile()
  {
     ofstream file("coaster.txt",ios::app);
     if(!file)
     {
        cerr << "Camt Open File" << endl;
        return ;
     }

     file << "Coaster name: "<< CoasName << endl;
     file << " Driver Name: " << DriverName << endl;
     file << " Model is " << model << endl;
```

```
        file.close();
    }
};

class NoBookingException : public exception
{
    const  char* what() const noexcept override
    {
        return " Unsuccesfull Booking Exception  ";
    }

};

class Booking
{
    protected:
    int BookingNum;
    string BookingName;
    int BookingDate;
    int BookingMonth;
    int BookingYear;
    int BookingStatus;

    public:
```

```
   Booking() :
BookingDate(0),BookingMonth(0),BookingYear(0),BookingStatus(-1),BookingNu
m(0)
   {
      BookingName = " ";
      BookingStatus =-1;
   }

   Booking(string name) :
BookingName(name),BookingDate(0),BookingMonth(0),BookingYear(0),Booking
Status(-1),BookingNum(0)
   {}

   void SetBookingName(string name)
   {
      BookingName = name;
   }

   void SetBookingNumber(int number)
   {
      BookingNum = number;
   }

   void SetAttributes()
   {
   }
```

```
  void PerformBooking()
  {
     cout << " Enter Details To Do Booking. Booking Number is : " <<
BookingNum << endl;
     int month;
     int date;
     int year;

     cout << " Enter Month of Booking " << endl;
     cin >> month;
     cout << " Enter Date of booking " << endl;
     cin >> date;
     if(date >=0 && date<32 && month > 0 && month < 13)
     {
        double Fee;
        cout << " Enter Fee " << endl;
        cin >> Fee;
        if(Fee < 0)
        {
           cout << "Cant Be negative " << endl;
           return;
        }

        string Acneed;
```

```
        cout << "Enter In Yes/No if You need Ac " << endl;
        cin >> Acneed;
        PaymentFee payment(Fee,Acneed);

        if(Acneed != "No")
        {
          int result = payment.PaymentFees(Fee + 2000);
          if(result !=0)
          {
           BookingMonth = month;
           BookingYear = year;
           BookingDate = date;
           string bookingname;
           cout << " Enter Booking Name " << endl;
           cin >> bookingname;
           SetBookingName(bookingname);
           SetBookingNumber(BookingNum+1);
           cout << "Succesfully!. Booking Proceeded " << " Booking Number is " <<
BookingNum << " and Booking Name is " << BookingName << endl;
           BookingStatus = 1;
          }
          else
          {
           cout << " Sumbit Fees First " << endl;
           throw PaymentNotExcep();
```

```
        }
      }
      else
      {
        int result = payment.PaymentFees(Fee);
        if(result !=0)
        {
          BookingMonth = month;
          BookingYear = year;
          BookingDate = date;

          string bookingname;
        cout << " Enter Booking Name " << endl;
        cin >> bookingname;
        SetBookingName(bookingname);
        SetBookingNumber(BookingNum+1);
          cout << "Succesfully!. Booking Proceeded " << " Booking Number is "
<< BookingNum << " and Booking Name is " << BookingName <<   endl;
          BookingStatus = 1;
        }
        else
        {
        cout << " Sumbit Fees First " << endl;
        throw PaymentNotExcep();
        }
```

```
        }
      }
      else
      {
       cout << " Cant Do For Booking Now " << endl;
       throw NoBookingException();
      }
   }

   int GetStatus()
   {
      return BookingStatus;
   }

   string GetBookingName()
   {
      return BookingName;
   }
};

class NadeemTransport
{
   private:
   int Points;
   int TotalDrivers;
```

```
    int trackdriver;
    Driver * driver;
    Bus * bus;
    Coaster * coaster;
    int TotalVehicles;
    int TotalCoaster;
    int TotalBus;
    int trackbus;
    int trackcoast;
    int trackvehicle;
    string AssignedRoutes[25] =
    {
    "Route1A", "Route1", "Route2", "Route3", "Route4",
    "Route5", "Route6", "Route7", "Route8", "Route9",
    "Route10", "Route11", "Route12", "Route13A", "Route13B",
    "Route15", "Route16", "Route17", "Route18", "Route18B",
    "Route19","Route20", "Route21", "Route22","Route24"
    };
     int MaxDriver = 25;


    public:
    NadeemTransport() :
trackdriver(0),TotalVehicles(0),trackvehicle(0),TotalDrivers(MaxDriver),TotalBus(
25),TotalCoaster(25)
    {
```

```
    driver = new Driver[25];
    bus = new Bus[TotalBus];
   coaster = new Coaster[TotalCoaster];
}

NadeemTransport(int tdriver, int tbus, int tcoaster, int tvehicle)
: TotalDrivers(tdriver), TotalBus(tbus), TotalCoaster(tcoaster),
  TotalVehicles(tvehicle), trackdriver(0), trackvehicle(0),
  trackbus(0), trackcoast(0)
{

    driver = new Driver[TotalDrivers];
    bus = new Bus[TotalBus];
   coaster = new Coaster[TotalCoaster];

}

void SetAttributes()
{
}

void ProvideVehicleToDriver()
{
    string vehicle;
```

```
    cout << " Enter Choice Of Whether A Bus Or Coaster Is To Be Assigned " <<
endl;
    cin >> vehicle;
    if(vehicle == "Bus")
    {
       if(trackbus < TotalBus)
       {
          string driverName;
          cout << " Enter Driver Name " << endl;
          cin >> driverName;

          string coastern;
          cout << "Enter coastername: " << endl;
          cin >> coastern;

          string mdel;
          cout << "Enter model name: " << endl;
          cin >> mdel;
          bus[trackbus].SetAttributes(coastern,driverName,mdel);
          string name = driver[trackdriver].GetName();
          bus[trackbus].displayInfo(name);
         driver[trackdriver].displayDetails();
          trackbus++;
       }
    }
```

```
else if(vehicle == "Coaster")
{
   if(trackcoast < TotalCoaster)
   {
       string driverName;
      cout << " Enter Driver Name " << endl;
      cin >> driverName;

      string coastern;
      cout << "Enter coastername: " << endl;
      cin >> coastern;

      string mdel;
      cout << "Enter modelname: " << endl;
      cin >> mdel;



      coaster[trackcoast].SetAttributes(coastern,driverName,mdel);
      string name = driver[trackdriver].GetName();
      coaster[trackcoast].displayInfo(name);

     driver[trackdriver].displayDetails();
      trackcoast++;
   }
   else
```

```
        {
            cout << "Less Space"<<endl;
        }
    }
    else
    {
        throw EntityFoundNot();
    }
}

void AddDriver()
{
  TotalDrivers = MaxDriver;
  if(trackdriver < TotalDrivers)
  {
      driver[trackdriver].SetAttributes();
      ProvideVehicleToDriver();
      trackdriver++;
  }
  else
  {
      cout << "Sorry. Cant Add!."<<endl;
      throw OutOfRangeExcep();
  }
}
```

```
    string GetDriverName()
    {
        if (trackdriver > 0)
        {
            return driver[trackdriver -  1].GetName();
        }
        else
        {
            return "No driver";
        }
    }

    string GetRouteName(int i)
    {
        return AssignedRoutes[i];
    }

};

class ZulfiqarTransport
{
    int Points;
    int TotalDrivers;
    int TotalCoaster;
```

```
   int TotalBus;
   string AssignedRoutes[25] =
   {
   "Route1A", "Route1", "Route2", "Route3", "Route4",
   "Route5", "Route6", "Route7", "Route8", "Route9",
   "Route10", "Route11", "Route12", "Route13A", "Route13B",
   "Route15", "Route16", "Route17", "Route18", "Route18B",
   "Route19","Route20", "Route21", "Route22","Route24"
   };
   Driver * driver;
   Bus * bus;
   Coaster * coaster;
   int trackbus;
   int trackcoaster;
   int trackdriver;
    int MaxDriver = 25;


   public:
   ZulfiqarTransport()
:trackdriver(0),trackbus(0),trackcoaster(0),TotalDrivers(MaxDriver),TotalBus(25),
TotalCoaster(25)
   {
      driver = new Driver[25];
      bus = new Bus[TotalBus];
      coaster = new Coaster[TotalCoaster];
```

```
    }


    ZulfiqarTransport(int Tdrive,int Tcoas,int tbus) :
TotalDrivers(Tdrive),TotalBus(tbus),TotalCoaster(Tcoas),trackdriver(0),trackbus(0)
,trackcoaster(0)
    {
        driver = new Driver[Tdrive];
        bus = new Bus[TotalBus];
        coaster = new Coaster[TotalCoaster];
    }


    void SetAttributes()
    {
    }


     void ProvideVehicleToDriver()
    {
        string vehicle;
        cout << " Enter Choice Of Whether A Bus Or Coaster Is To Be Assigned " <<
endl;
        cin >> vehicle;
        if(vehicle == "Bus")
        {
            if(trackbus < TotalBus)
            {
```

```
        string driverName;
        cout << " Enter Driver Name " << endl;
        cin >> driverName;

        string coastern;
        cout << "Enter coastername: " << endl;
        cin >> coastern;

        string mdel;
        cout << "Enter model: " << endl;
        cin >> mdel;

        bus[trackbus].SetAttributes(coastern,driverName,mdel);
        string name = driver[trackdriver].GetName();
        bus[trackbus].displayInfo(name);
        bus[trackbus].SafeBusDataToFile();
        trackbus++;
         driver[trackdriver].displayDetails();
    }
}
else if(vehicle == "Coaster")
{
    if(trackcoaster < TotalCoaster)
    {
        string driverName;
```

```
        cout << " Enter Driver Name " << endl;
        cin >> driverName;

        string coastern;
        cout << "Enter coastername: " << endl;
        cin >> coastern;

        string mdel;
        cout << "Enter model: " << endl;
        cin >> mdel;


        coaster[trackcoaster].SetAttributes(coastern,driverName,mdel);
        string name = driver[trackdriver].GetName();
        coaster[trackcoaster].displayInfo(name);
        coaster[trackcoaster].SafeCoasterDataToFile();
        trackcoaster++;
        driver[trackdriver].displayDetails();
    }
    else
    {
        cout << "Less Space"<<endl;
    }
}
else
```

```
    {
        throw EntityFoundNot();
    }
}

void AddDriver()
{
    TotalDrivers = MaxDriver;
   if(trackdriver < TotalDrivers)
   {
       driver[trackdriver].SetAttributes();
        ProvideVehicleToDriver();
          trackdriver++;
   }
   else
   {
       cout << "Cant Add!"<<endl;
       throw OutOfRangeExcep();
   }
}

 string GetDriverName()
 {
    if (trackdriver > 0)
    {
```

```
            return driver[trackdriver -  1].GetName();
        }
        else
        {
            return "No driver";
        }
    }

    string GetRouteName(int i)
    {
        return AssignedRoutes[i];
    }

};
template<typename T>
class GenericStructure
{
    protected:
    T * content[1000];
    int trackcontent;

    public:
    GenericStructure() : trackcontent(0)
    {}
```

```
   void AddContent()
   {
        if(trackcontent < 1000)
        {
           content[trackcontent] = new T();
         content[trackcontent]->SetAttributes();//My logic For Guidance: All
Empty Set Attributes functions are used to initiate the process.
          trackcontent++;
        }
        else
         {
          throw OutOfRangeExcep();
         }
   }

   T& operator[](int index)
   {
      if(index < 0 || index >=trackcontent)
      {
         throw OutOfRangeExcep();
      }
      return *content[index];
   }
};
```

```
class TransportManagementSystem
{
   private:
   const int BusSeats = 52;
   const int Coaster = 32;
   GenericStructure <NadeemTransport>  nadeem;
   GenericStructure <ZulfiqarTransport> zulfiqar;
   GenericStructure <Booking> bookings;
   GenericStructure <Vehicle> VEHICLE;
   GenericStructure <User> user;

   const int MaxBooking = 1000;
   int trackbooking;
   int tracknadeem;
   int trackzulfiqar;
   int TotalNadeemDriver;
   int TotalZulfiqarDriver;

   public:
   TransportManagementSystem() : trackbooking(0), tracknadeem(0),
trackzulfiqar(0)
   {}
```

```
   TransportManagementSystem(int totalnad, int totalzul) :
TotalNadeemDriver(totalnad), TotalZulfiqarDriver(totalzul), trackbooking(0),
tracknadeem(0), trackzulfiqar(0)
     {
      for (int i = 0; i < TotalNadeemDriver; i++)
      {
       nadeem.AddContent();
      }

      for (int i = 0; i < TotalNadeemDriver; i++)
      {
       zulfiqar.AddContent();
      }

       for (int i = 0; i < MaxBooking; i++)
      {
       bookings.AddContent();
      }
    }
    void ManageRoutes(string Transport)
   {
       if(Transport == "Nadeem")
       {
          cout << "Driver :"<<nadeem[tracknadeem].GetDriverName() << "Assigned
to route " << nadeem[tracknadeem].GetRouteName(tracknadeem) << endl;
```

```
    }
    else if(Transport == "Zulfiqar")
    {
        cout << "Driver :"<< zulfiqar[trackzulfiqar].GetDriverName() << "Assigned to route " << zulfiqar[trackzulfiqar].GetRouteName(tracknadeem) << endl;
    }
  }

  void AllocateSeats()
  {
    int randomBusSeatsFaculty = (rand() % 52 +1);

    int randomCoasterSeatsFaculty = (rand() %32 +1);
    cout << "Seats In The Vehicle(Bus) for faculty are " << randomBusSeatsFaculty << " and for students is "<<52-randomBusSeatsFaculty << endl;
    cout << "Seats In The Vehicle(Coaster) for faculty are " << randomCoasterSeatsFaculty << " and for students is "<< 32-randomCoasterSeatsFaculty<< endl;
  }

  void AssignDriver(string Transport)
  {
    if(Transport == "Zulfiqar")
    {
      if(trackzulfiqar < TotalZulfiqarDriver)
```

```
    {

     zulfiqar[trackzulfiqar].AddDriver();
     ManageRoutes(Transport);
     AllocateSeats();
     trackzulfiqar++;
    }
    else
    {
       cout << "Cant Add Sorry!." << endl;
    }
 }
else if(Transport == "Nadeem")
 {
    if(tracknadeem < TotalNadeemDriver)
    {
     nadeem[tracknadeem].AddDriver();

     ManageRoutes(Transport);
     AllocateSeats();
     tracknadeem++;
    }
    else
    {
       cout << "Cant Add Sorry!." << endl;
```

```
        }
    }
}

void SafeBookingDataToFile()
{
    ofstream file("booking.txt",ios::app);
    if(!file)
    {
        cerr << "Camt Open File" << endl;
        return ;
    }
    string name = bookings[trackbooking].GetBookingName();
    file << "Booking Name :"<< name << endl;

    file.close();
}

int HandleBooking()
{
    if(trackbooking < MaxBooking)
    {
        bookings[trackbooking].PerformBooking();
        if(bookings[trackbooking].GetStatus() == 1)
        {
```

```
            trackbooking++;
            cout << " Booking Succesfull " << endl;
            string name;
            cout << "Enter Booking Name write same name " << endl;
            cin >> name;
            bookings[trackbooking].SetBookingName(name);
            SafeBookingDataToFile();
            return 1;
          }
          else
          {
            cout << " Sorry.Please Do Payment " << endl;
            return 0;
          }
        }
        else
        {
          cout << "Error!"<<endl;
          return 0;
        }
    }

};

int main()
```

```
{
  cout << " Welcome Transport Manager To Fast Bus Transportation System Enter Detils Of The Day To Do Booking etc. " << endl;
  cout << endl;
  TransportManagementSystem manager(25,25);
  NadeemTransport nadeem(25,25,10,5);
  ZulfiqarTransport zulfiqar(25,15,10);

  while(1)
  {
   cout << endl;
   cout << " __Welcome Student/Faculty for going to register route__ " << endl;
   string Choice;
   cout << "__Enter You Are Student/Faculty/AddDriver/Exit__" << endl;
   cin >> Choice;
   if(Choice == "Student")
   {
      try
      {
         Student student;
         int Bookingresult =  manager.HandleBooking();
         if(Bookingresult == 1)
         {
            student.StudentRegisteration();
            int stats = student.getstatus();
```

```
        cout << stats <<endl;
        if(stats == 1)
        {
          student.SaveStudentToFile(student);
          cout << "__Booking Succesfully Confirmed__" << endl;
          cout << "__Verify Your Details BY looking At it__ " << endl;
          student.DisplayInfo();
         }
         else
        {
          cout << "Coudnt Add " << endl;
        }

    }
    else
    {
       cout << "_Cant Do Processing Now. " << endl;
    }
  }
  catch(const exception& e)
  {
    cerr << e.what() << '\n';
  }

}
```

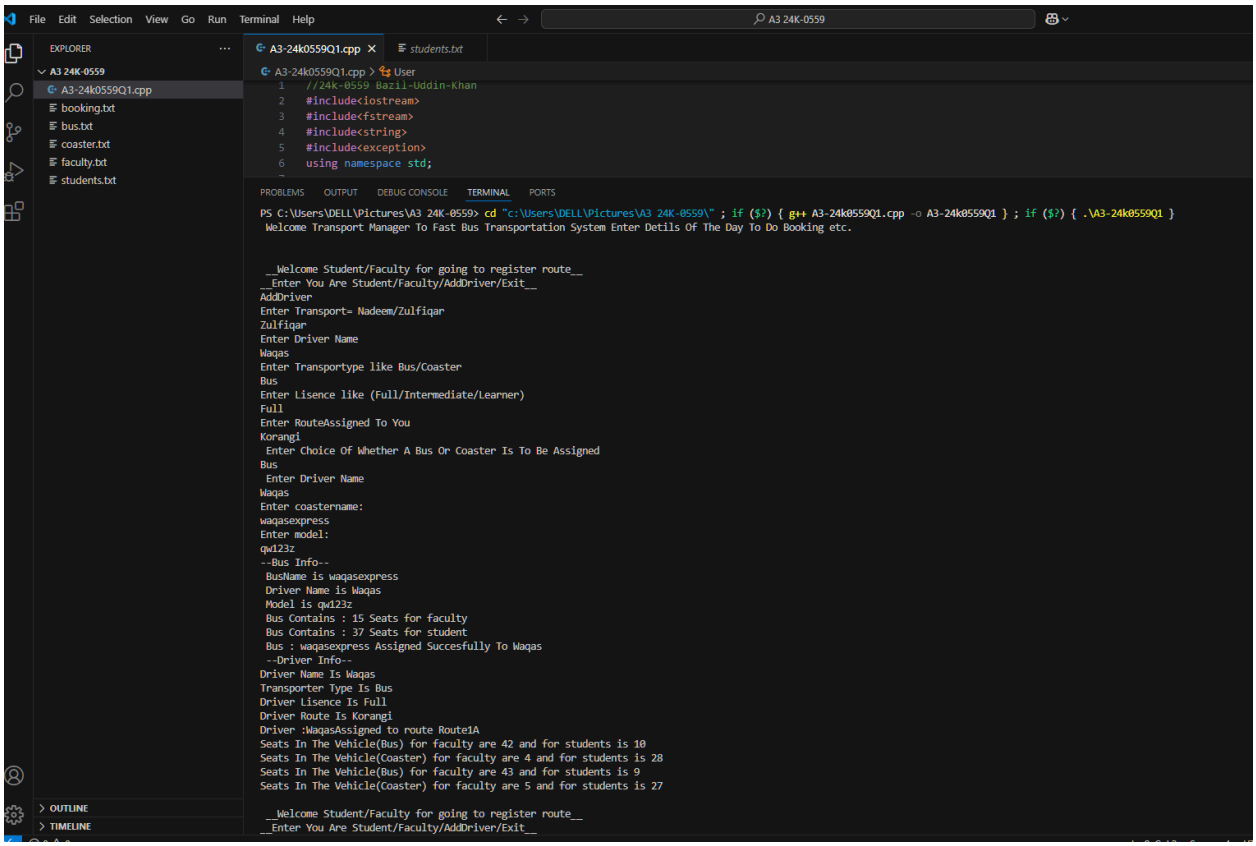
```
else if(Choice == "Faculty")
{
   string name;
   cout << "Enter Name " << endl;
   cin >> name;
   string date;
   cout << "Enter Date of join " << endl;
   cin >> date;
   string Email;
   cout << "Enter Email" << endl;
   cin >> Email;
   string number;
   cout << "Enter contact number" << endl;
   cin >> number;
   double fees;
   cout << "Enter fees" << endl;
   cin >> fees;
   string routenumber;
   cout << "Enter Route no " << endl;
   cin >> routenumber;
   double salary;
   cout << "Enter salary " << endl;
   cin >> salary;
   if(salary < 0 )
   {
```

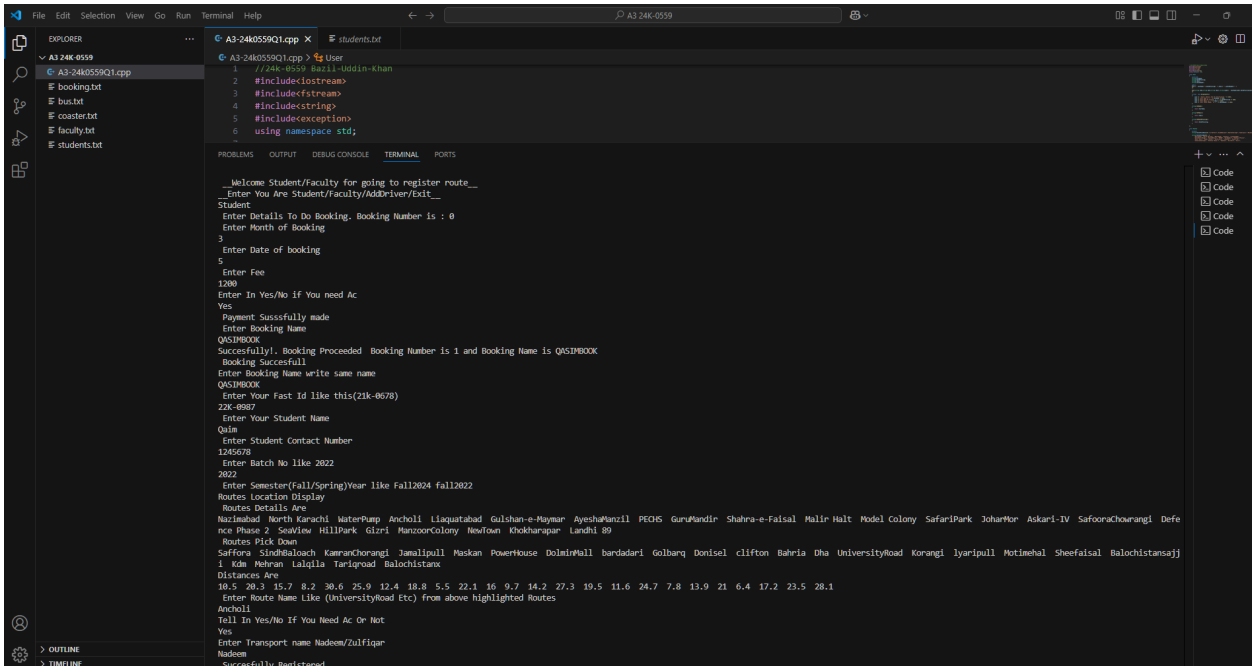
```
    break;
}

try
{

    Faculty faculty(name,date,Email,number,fees,routenumber,salary);
    int Bookresult = manager.HandleBooking();
    if(Bookresult == 1)
    {
      faculty.TeacherRouteRegisteration();
      int Status = faculty.getstatus();
       if(Status == 1)
       {
        faculty.SaveFacultyToFile(faculty);
        cout << "__Booking Succesfully Confirmed__" << endl;
        cout << "__Verify Your Details BY looking At it__" << endl;
        faculty.DisplayInfo();
        cout << "__Processing For Booking__And Savi g Data__" << endl;
     }
      else
      {
     cout << "_Coudnt Add. " << endl;
      }
   }
```

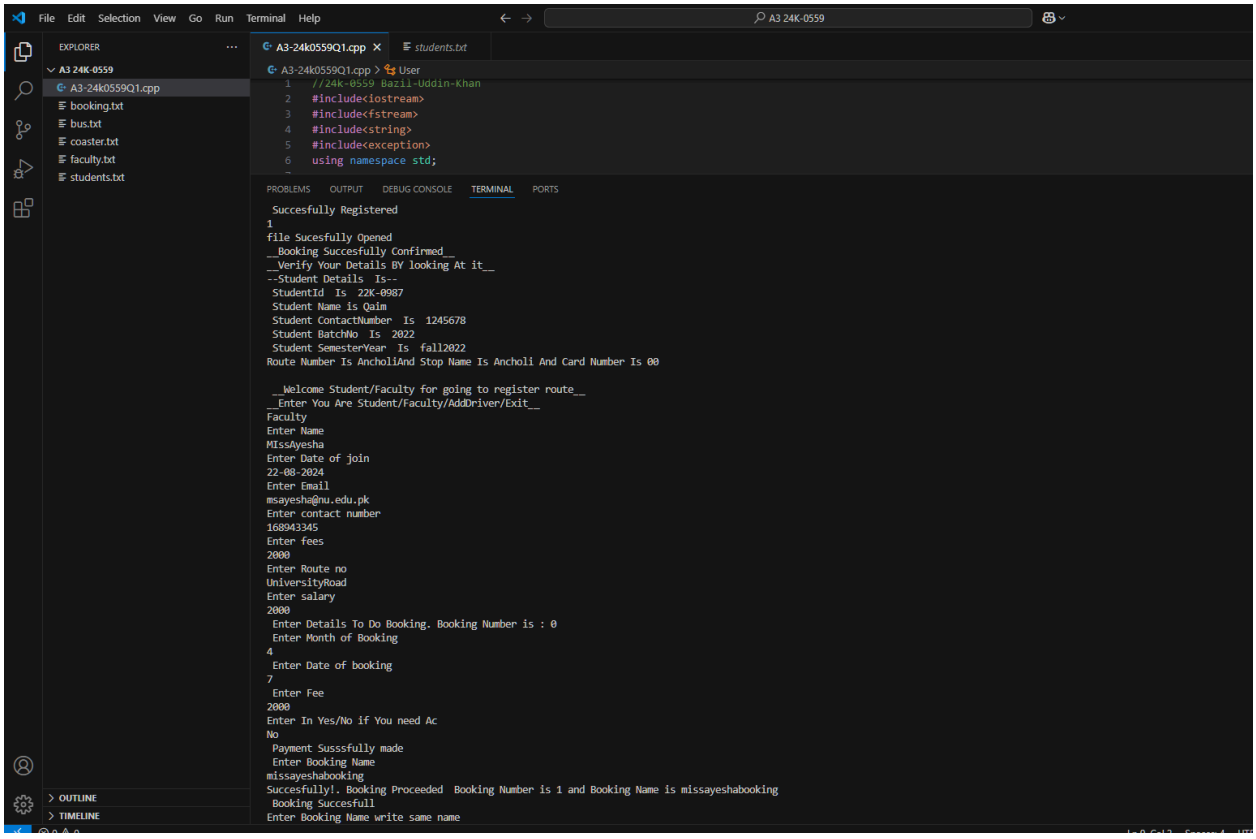
```
    }
    catch(const exception& e)
    {
        cerr << e.what() << '\n';
    }
}
else if(Choice == "AddDriver")
{
    try
    {
        string Transport;
        cout << "Enter Transport= Nadeem/Zulfiqar" << endl;
        cin >> Transport;

        manager.AssignDriver(Transport);
        manager.AllocateSeats();
    }
    catch(const exception& e)
    {
        cerr << e.what() << '\n';
    }
}
else if(Choice == "Exit")
{
    cout << " Thanks For Being With Us!. " << endl;
```

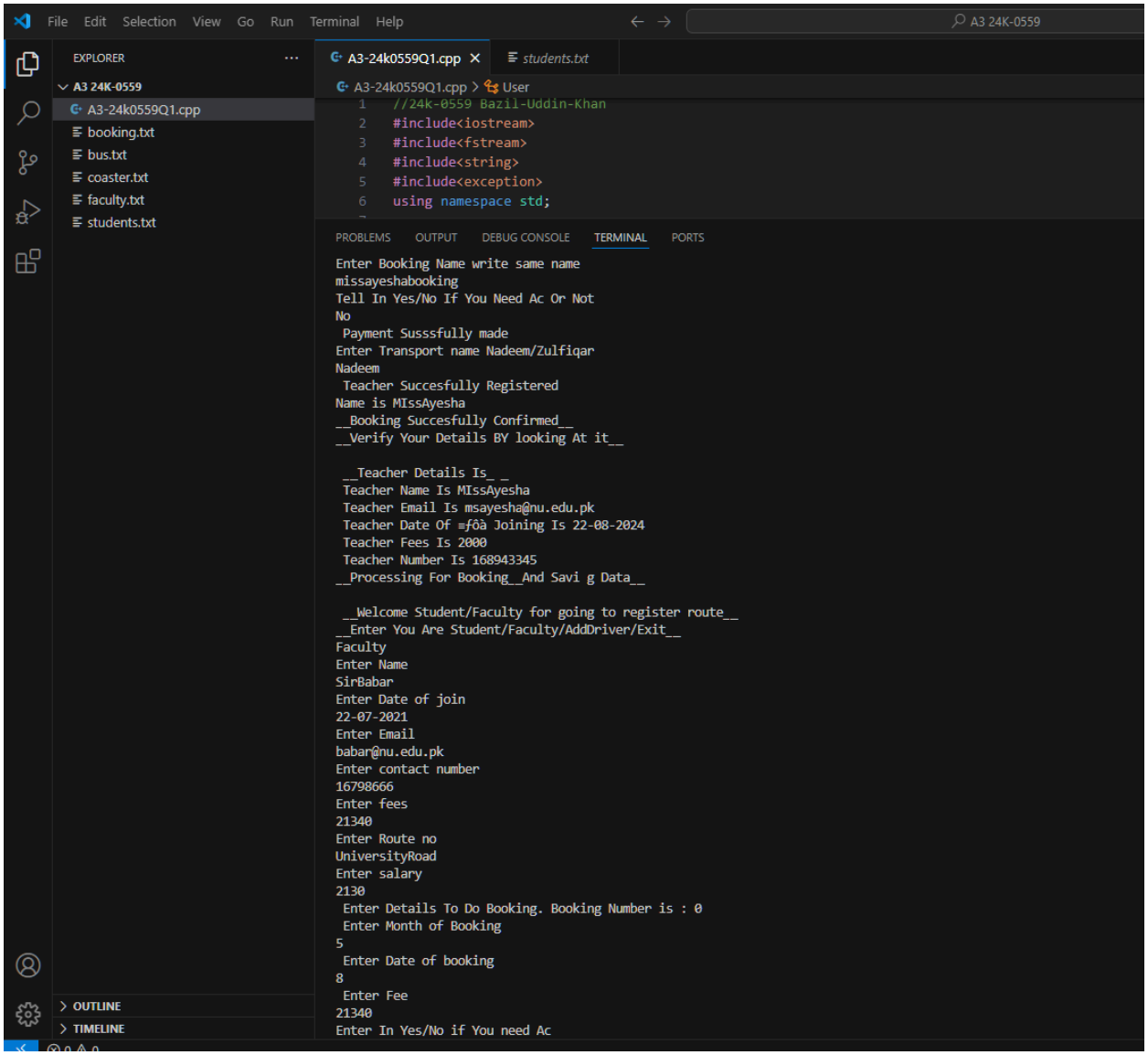
```
        return 0;
    }
    else
    {
        cout << " Wrong Retry! "<<endl;
    }
   }
}
```

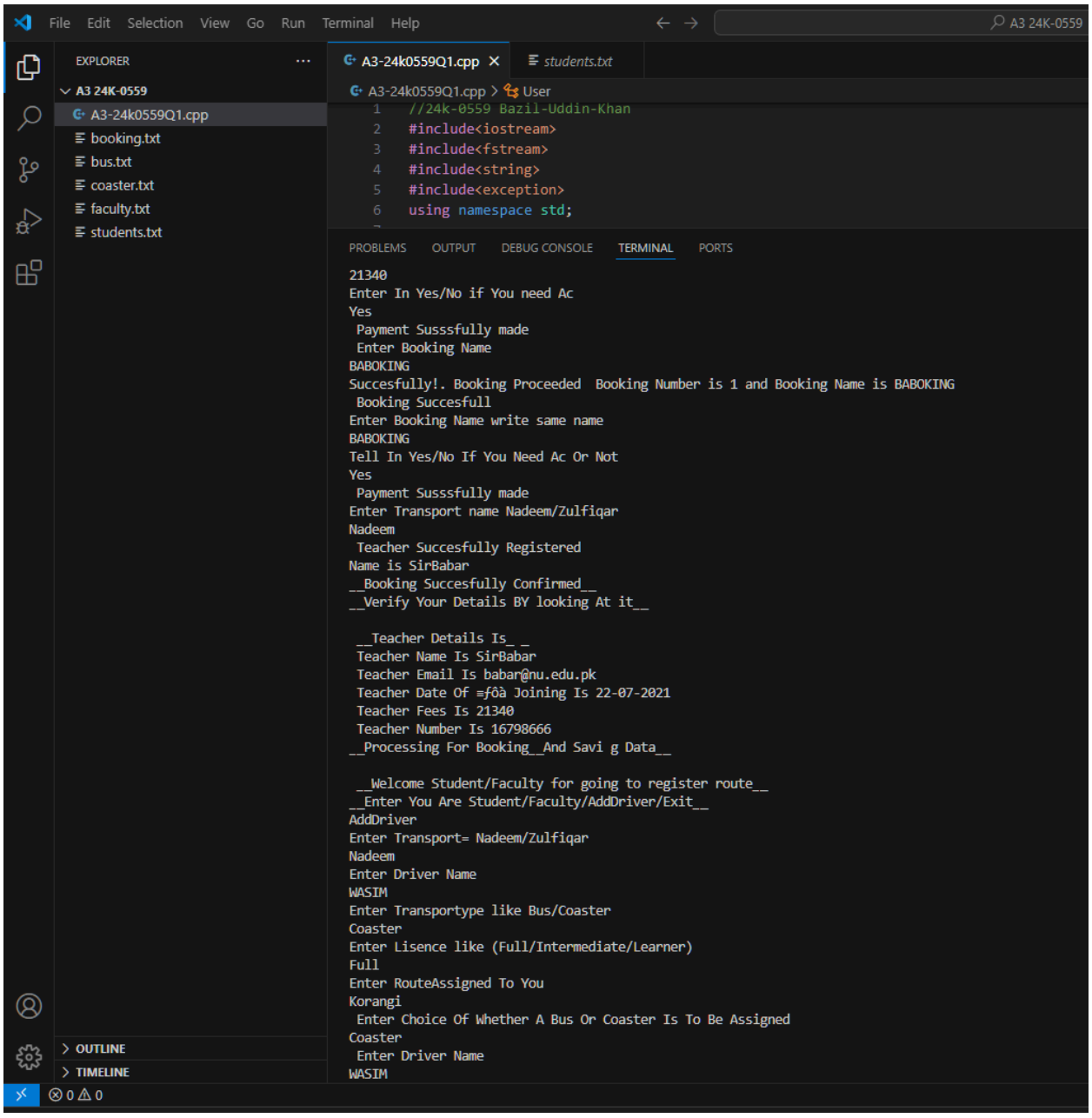
OUTPUT:

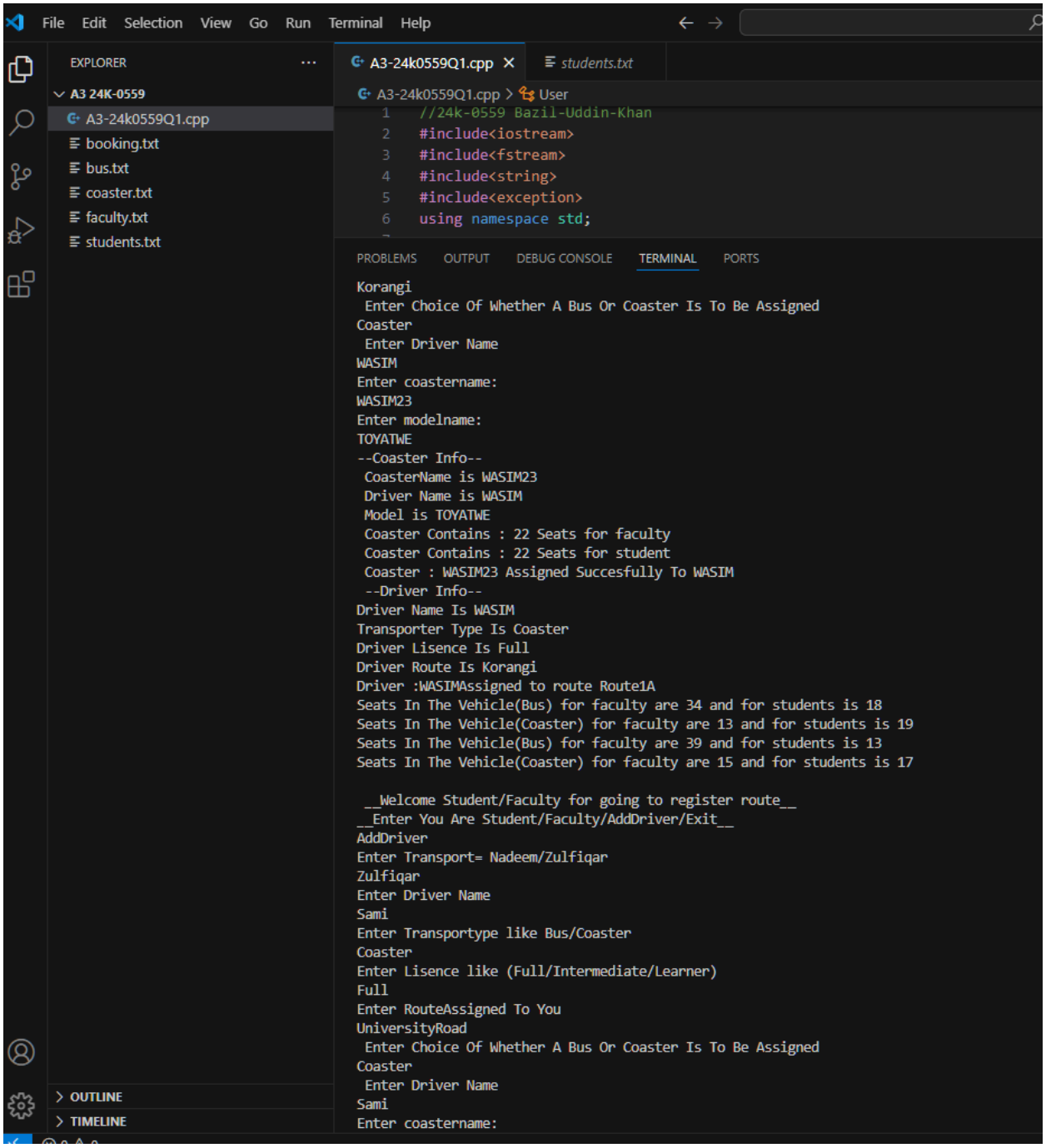

File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Pictures\A3 24K-0559> cd "c:\Users\DELL\Pictures\A3 24K-0559\" ; if ($?) { g++ A3-24k0559Q1.cpp -o A3-24k0559Q1 } ; if ($?) { .\A3-24k0559Q1 }
Welcome Transport Manager To Fast Bus Transportation System Enter Detils Of The Day To Do Booking etc.
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
AddDriver
Enter Transport= Nadeem/Zulfiqar
Zulfiqar
Enter Driver Name
Waqas
Enter Transportype like Bus/Coaster
Bus
Enter Lisence like (Full/Intermediate/Learner)
Full
Enter RouteAssigned To You
Korangi
Enter Choice Of Whether A Bus Or Coaster Is To Be Assigned
Bus
Enter Driver Name
Waqas
Enter coastername:
waqasexpress
Enter model:
qw123z
--Bus Info--
BusName is waqasexpress
Driver Name is Waqas
Model is qw123z
Bus Contains : 15 Seats for faculty
Bus Contains : 37 Seats for student
Bus : waqasexpress Assigned Succesfully To Waqas
--Driver Info--
Driver Name Is Waqas
Transporter Type Is Bus
Driver Lisence Is Full
Driver Route Is Korangi
Driver :WaqasAssigned to route Route1A
Seats In The Vehicle(Bus) for faculty are 42 and for students is 10
Seats In The Vehicle(Coaster) for faculty are 4 and for students is 28
Seats In The Vehicle(Bus) for faculty are 43 and for students is 9
Seats In The Vehicle(Coaster) for faculty are 5 and for students is 27
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
OUTLINE
TIMELINE

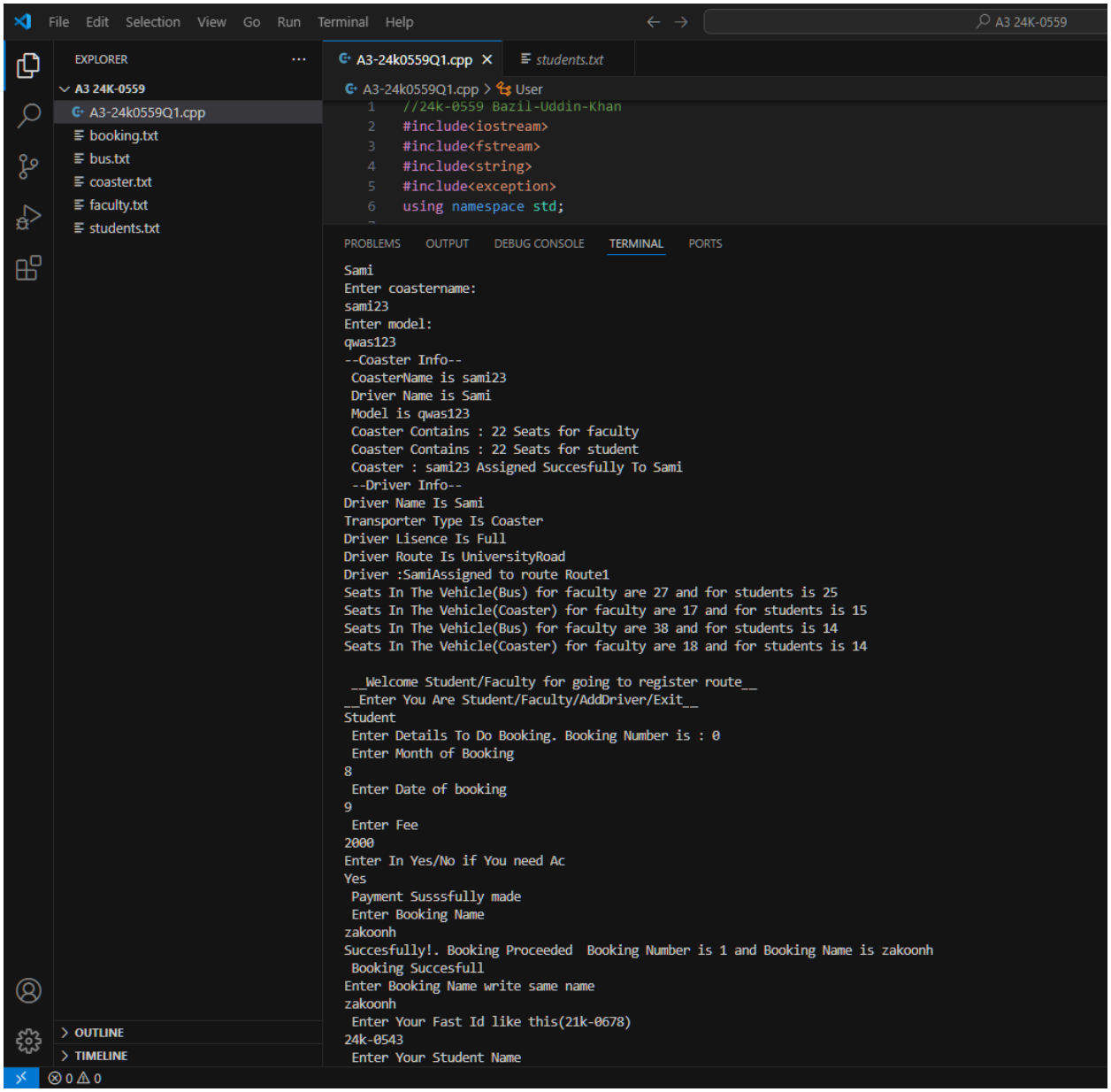

File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Student
Enter Details To Do Booking. Booking Number is : 0
Enter Month of Booking
3
Enter Date of booking
5
Enter Fee
1200
Enter In Yes/No if You need Ac
Yes
Payment Sussfully made
Enter Booking Name
QASIMBOOK
Succesfully!. Booking Proceeded Booking Number is 1 and Booking Name is QASIMBOOK
Booking Succesfull
Enter Booking Name write same name
QASIMBOOK
Enter Your Fast Id like this(21k-0678)
22K-0987
Enter Your Student Name
Qaim
Enter Student Contact Number
1245678
Enter Batch No like 2022
2022
Enter Semester(Fall/Spring)Year like Fall2024 fall2022
Routes Location Display
Routes Details Are
Nazimabad North Karachi WaterPump Ancholi Liaquatabad Gulshan-e-Maymar AyeshaManzil PECHS GuruMandir Shahra-e-Faisal Malir Halt Model Colony SafariPark JoharMor Askari-IV SafooraChowrangi Defence Phase 2 SeaView HillPark Gizri ManzoorColony NewTown Khokharapar Landhi 89
Routes Pick Down
Saffora SindhBaloach KamranChorangi Jamalipull Maskan PowerHouse DolminMall bardadari Golbarq Donisel clifton Bahria Dha UniversityRoad Korangi lyaripull Motimehal Sheefaisal Balochistansajjad Kdm Mehran Lalqila Tariqroad Balochistanx
Distances Are
10.5 20.3 15.7 8.2 30.6 25.9 12.4 18.8 5.5 22.1 16 9.7 14.2 27.3 19.5 11.6 24.7 7.8 13.9 21 6.4 17.2 23.5 28.1
Enter Route Name Like (UniversityRoad Etc) from above highlighted Routes
Ancholi
Tell In Yes/No If You Need Ac Or Not
Yes
Enter Transport name Nadeem/Zulfiqar
Nadeem
Succesfully Registered
OUTLINE
TIMELINE
Code
Code
Code
Code
Code

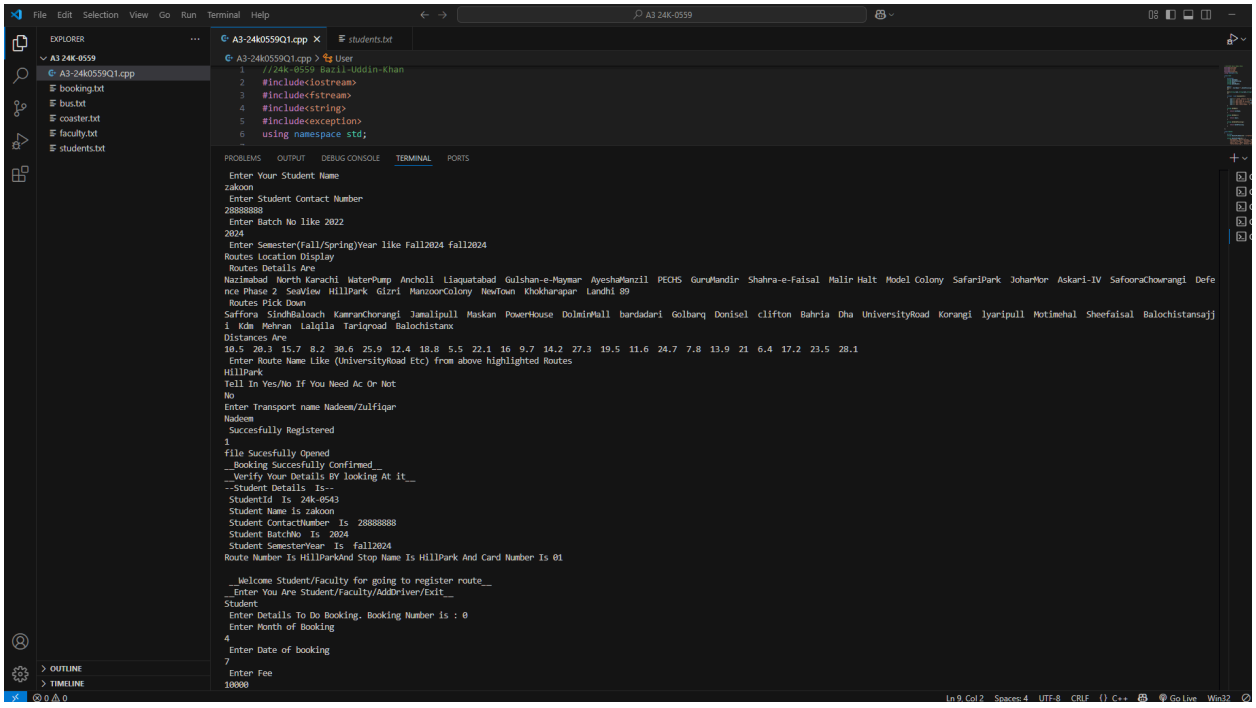

File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Succesfully Registered
1
file Sucesfully Opened
__Booking Succesfully Confirmed__
__Verify Your Details BY looking At it__
--Student Details Is--
StudentId Is 22K-0987
Student Name is Qaim
Student ContactNumber Is 1245678
Student BatchNo Is 2022
Student SemesterYear Is fall2022
Route Number Is AncholiAnd Stop Name Is Ancholi And Card Number Is 00
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Faculty
Enter Name
MIssAyesha
Enter Date of join
22-08-2024
Enter Email
msayesha@nu.edu.pk
Enter contact number
168943345
Enter fees
2000
Enter Route no
UniversityRoad
Enter salary
2000
Enter Details To Do Booking. Booking Number is : 0
Enter Month of Booking
4
Enter Date of booking
7
Enter Fee
2000
Enter In Yes/No if You need Ac
No
Payment Sussfully made
Enter Booking Name
missayeshabooking
Succesfully!. Booking Proceeded Booking Number is 1 and Booking Name is missayeshabooking
Booking Succesfull
Enter Booking Name write same name
OUTLINE
TIMELINE

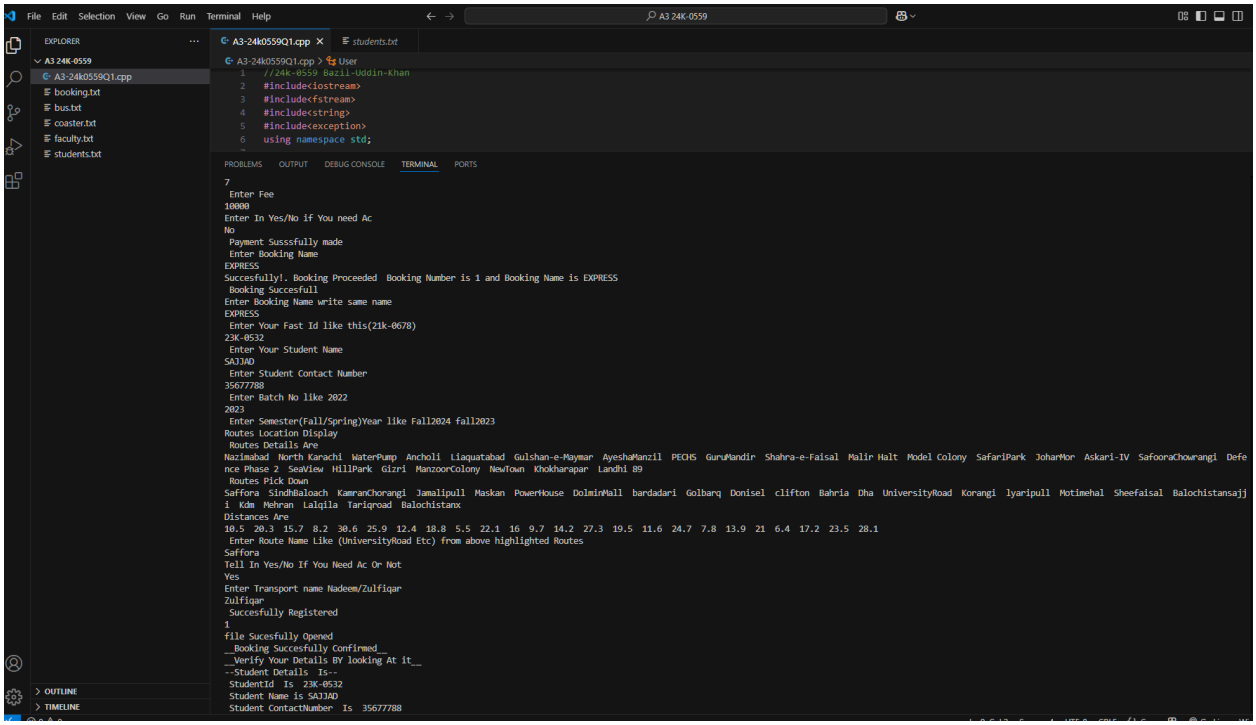

File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter Booking Name write same name
missayeshabooking
Tell In Yes/No If You Need Ac Or Not
No
Payment Sussfully made
Enter Transport name Nadeem/Zulfiqar
Nadeem
Teacher Succesfully Registered
Name is MIssAyesha
__Booking Succesfully Confirmed__
__Verify Your Details BY looking At it__
__Teacher Details Is_ _
Teacher Name Is MIssAyesha
Teacher Email Is msayesha@nu.edu.pk
Teacher Date Of ≡fôà Joining Is 22-08-2024
Teacher Fees Is 2000
Teacher Number Is 168943345
__Processing For Booking__And Savi g Data__
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Faculty
Enter Name
SirBabar
Enter Date of join
22-07-2021
Enter Email
babar@nu.edu.pk
Enter contact number
16798666
Enter fees
21340
Enter Route no
UniversityRoad
Enter salary
2130
Enter Details To Do Booking. Booking Number is : 0
Enter Month of Booking
5
Enter Date of booking
8
Enter Fee
21340
Enter In Yes/No if You need Ac
OUTLINE
TIMELINE

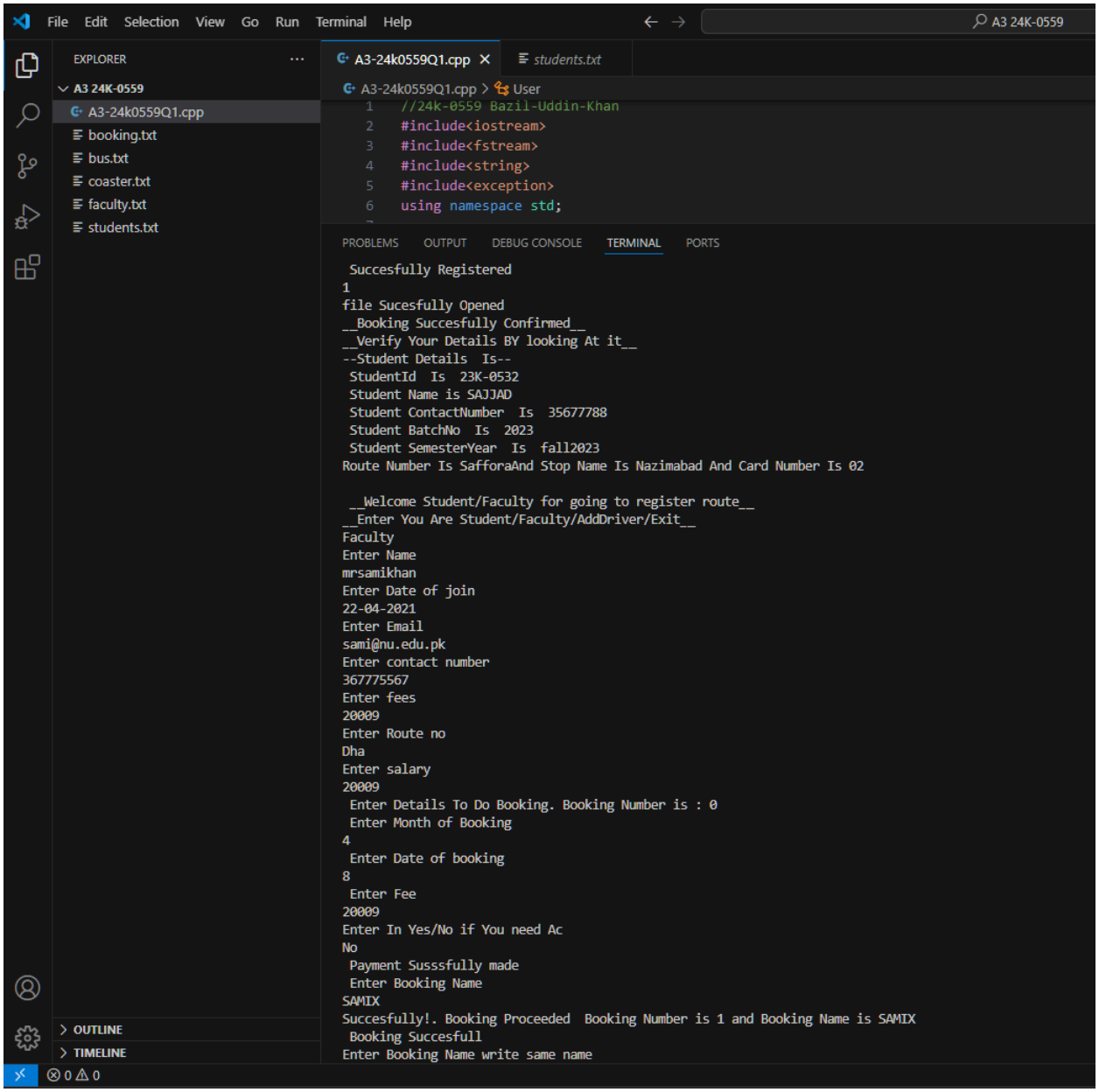

File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
21340
Enter In Yes/No if You need Ac
Yes
Payment Susssfully made
Enter Booking Name
BABOKING
Succesfully!. Booking Proceeded Booking Number is 1 and Booking Name is BABOKING
Booking Succesfull
Enter Booking Name write same name
BABOKING
Tell In Yes/No If You Need Ac Or Not
Yes
Payment Susssfully made
Enter Transport name Nadeem/Zulfiqar
Nadeem
Teacher Succesfully Registered
Name is SirBabar
__Booking Succesfully Confirmed__
__Verify Your Details BY looking At it__
__Teacher Details Is_ _
Teacher Name Is SirBabar
Teacher Email Is babar@nu.edu.pk
Teacher Date Of ≡ƒôà Joining Is 22-07-2021
Teacher Fees Is 21340
Teacher Number Is 16798666
__Processing For Booking__And Savi g Data__
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
AddDriver
Enter Transport= Nadeem/Zulfiqar
Nadeem
Enter Driver Name
WASIM
Enter Transportype like Bus/Coaster
Coaster
Enter Lisence like (Full/Intermediate/Learner)
Full
Enter RouteAssigned To You
Korangi
Enter Choice Of Whether A Bus Or Coaster Is To Be Assigned
Coaster
Enter Driver Name
WASIM
OUTLINE
TIMELINE
⊗ 0 ⚠ 0

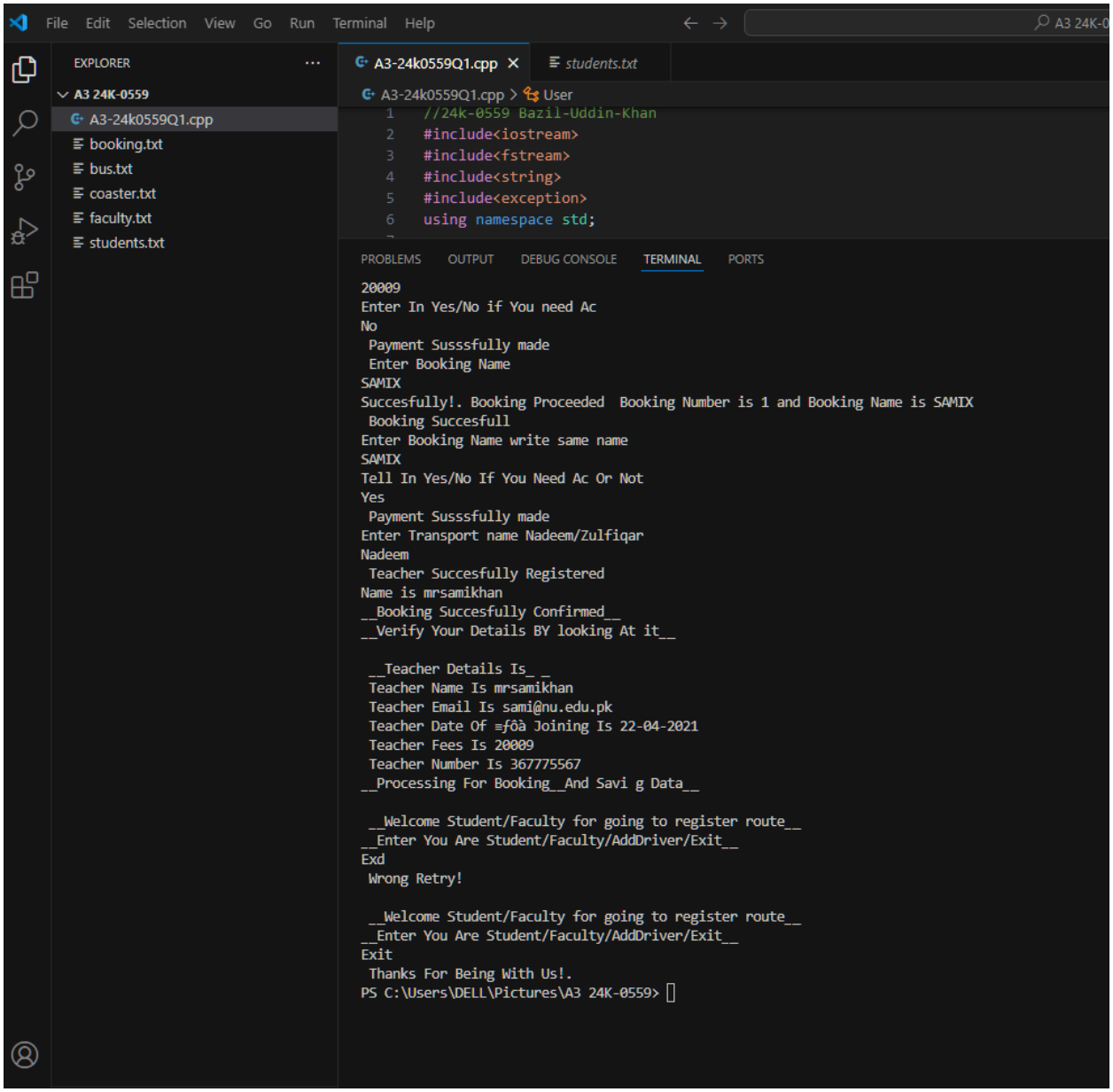

File Edit Selection View Go Run Terminal Help
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Korangi
Enter Choice Of Whether A Bus Or Coaster Is To Be Assigned
Coaster
Enter Driver Name
WASIM
Enter coastername:
WASIM23
Enter modelname:
TOYATWE
--Coaster Info--
CoasterName is WASIM23
Driver Name is WASIM
Model is TOYATWE
Coaster Contains : 22 Seats for faculty
Coaster Contains : 22 Seats for student
Coaster : WASIM23 Assigned Succesfully To WASIM
--Driver Info--
Driver Name Is WASIM
Transporter Type Is Coaster
Driver Lisence Is Full
Driver Route Is Korangi
Driver :WASIMAssigned to route Route1A
Seats In The Vehicle(Bus) for faculty are 34 and for students is 18
Seats In The Vehicle(Coaster) for faculty are 13 and for students is 19
Seats In The Vehicle(Bus) for faculty are 39 and for students is 13
Seats In The Vehicle(Coaster) for faculty are 15 and for students is 17
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
AddDriver
Enter Transport= Nadeem/Zulfiqar
Zulfiqar
Enter Driver Name
Sami
Enter Transportype like Bus/Coaster
Coaster
Enter Lisence like (Full/Intermediate/Learner)
Full
Enter RouteAssigned To You
UniversityRoad
Enter Choice Of Whether A Bus Or Coaster Is To Be Assigned
Coaster
Enter Driver Name
Sami
Enter coastername:
OUTLINE
TIMELINE


File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Sami
Enter coastername:
sami23
Enter model:
qwas123
--Coaster Info--
CoasterName is sami23
Driver Name is Sami
Model is qwas123
Coaster Contains : 22 Seats for faculty
Coaster Contains : 22 Seats for student
Coaster : sami23 Assigned Succesfully To Sami
--Driver Info--
Driver Name Is Sami
Transporter Type Is Coaster
Driver Lisence Is Full
Driver Route Is UniversityRoad
Driver :SamiAssigned to route Route1
Seats In The Vehicle(Bus) for faculty are 27 and for students is 25
Seats In The Vehicle(Coaster) for faculty are 17 and for students is 15
Seats In The Vehicle(Bus) for faculty are 38 and for students is 14
Seats In The Vehicle(Coaster) for faculty are 18 and for students is 14
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Student
Enter Details To Do Booking. Booking Number is : 0
Enter Month of Booking
8
Enter Date of booking
9
Enter Fee
2000
Enter In Yes/No if You need Ac
Yes
Payment Susssfully made
Enter Booking Name
zakoonh
Succesfully!. Booking Proceeded Booking Number is 1 and Booking Name is zakoonh
Booking Succesfull
Enter Booking Name write same name
zakoonh
Enter Your Fast Id like this(21k-0678)
24k-0543
Enter Your Student Name
OUTLINE
TIMELINE
0 0


File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter Your Student Name
zakoon
Enter Student Contact Number
28888888
Enter Batch No like 2022
2024
Enter Semester(Fall/Spring)Year like Fall2024 fall2024
Routes Location Display
Routes Details Are
Nazimabad North Karachi WaterPump Ancholi Liaquatabad Gulshan-e-Maymar AyeshaManzil PECHS GuruMandir Shahra-e-Faisal Malir Halt Model Colony SafariPark JoharMor Askari-IV SafooraChowrangi Defence Phase 2 SeaView HillPark Gizri ManzoorColony NewTown Khokharapar Landhi 89
Routes Pick Down
Saffora SindhBaloach KamranChorangi Jamalipull Maskan PowerHouse DolminMall bardadari Golbarq Donisel clifton Bahria Dha UniversityRoad Korangi lyaripull Motimehal Sheefaisal Balochistansajji Kdm Mehran Lalqila Tariqroad Balochistanx
Distances Are
10.5 20.3 15.7 8.2 30.6 25.9 12.4 18.8 5.5 22.1 16 9.7 14.2 27.3 19.5 11.6 24.7 7.8 13.9 21 6.4 17.2 23.5 28.1
Enter Route Name Like (UniversityRoad Etc) from above highlighted Routes
HillPark
Tell In Yes/No If You Need Ac Or Not
No
Enter Transport name Nadeem/Zulfiqar
Nadeem
Succesfully Registered
1
file Sucesfully Opened
__Booking Succesfully Confirmed__
__Verify Your Details BY looking At it__
--Student Details Is--
StudentId Is 24k-0543
Student Name is zakoon
Student ContactNumber Is 28888888
Student BatchNo Is 2024
Student SemesterYear Is fall2024
Route Number Is HillParkAnd Stop Name Is HillPark And Card Number Is 01
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Student
Enter Details To Do Booking. Booking Number is : 0
Enter Month of Booking
4
Enter Date of booking
7
Enter Fee
10000
OUTLINE
TIMELINE

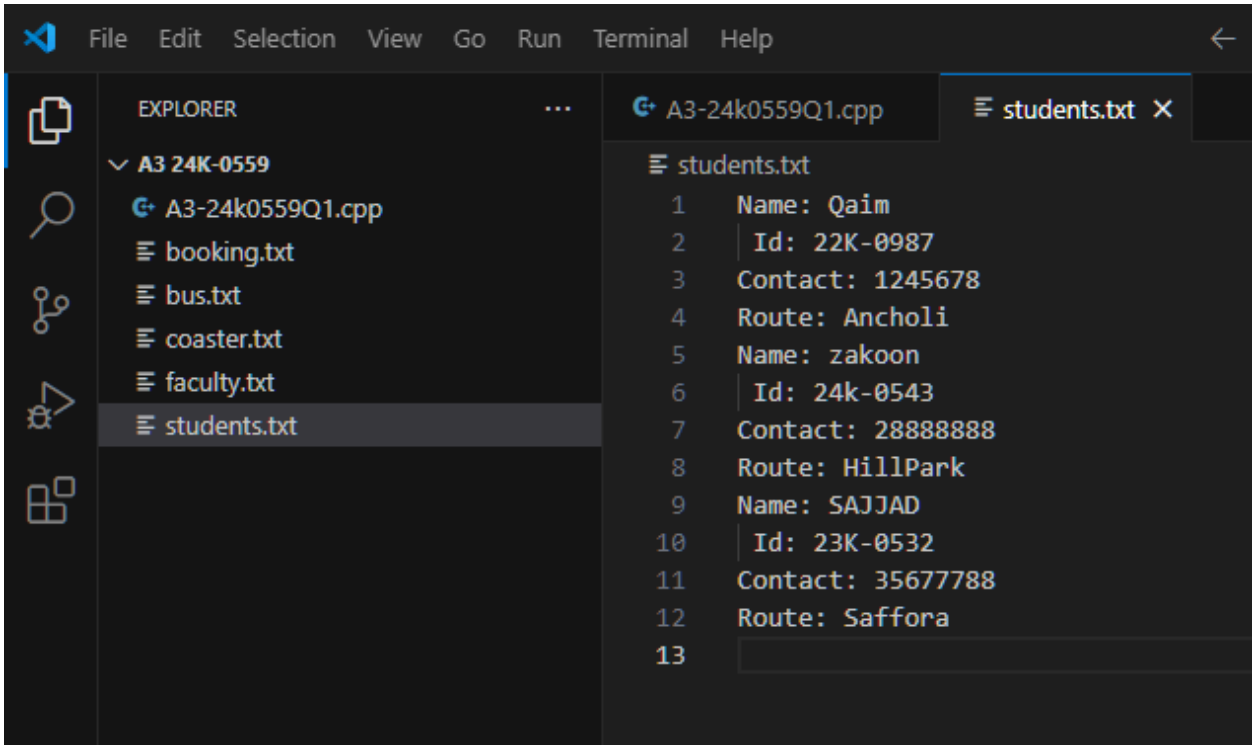

File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
7
Enter Fee
10000
Enter In Yes/No if You need Ac
No
Payment Sussfully made
Enter Booking Name
EXPRESS
Succesfully!. Booking Proceeded Booking Number is 1 and Booking Name is EXPRESS
Booking Succesfull
Enter Booking Name write same name
EXPRESS
Enter Your Fast Id like this(21k-0678)
23K-0532
Enter Your Student Name
SAJJAD
Enter Student Contact Number
35677788
Enter Batch No like 2022
2023
Enter Semester(Fall/Spring)Year like Fall2024 fall2023
Routes Location Display
Routes Details Are
Nazimabad North Karachi WaterPump Ancholi Liaquatabad Gulshan-e-Maymar AyeshaManzil PECHS GuruMandir Shahra-e-Faisal Malir Halt Model Colony SafariPark JoharMor Askari-IV SafooraChowrangi Defence Phase 2 SeaView HillPark Gizri ManzoorColony NewTown Khokharapar Landhi 89
Routes Pick Down
Saffora SindhBaloach KamranChorangi Jamalipull Maskan PowerHouse DolminMall bardadari Golbarq Donisel clifton Bahria Dha UniversityRoad Korangi lyaripull Motimehal Sheefaisal Balochistansajji Kdm Mehran Lalqila Tariqroad Balochistanx
Distances Are
10.5 20.3 15.7 8.2 30.6 25.9 12.4 18.8 5.5 22.1 16 9.7 14.2 27.3 19.5 11.6 24.7 7.8 13.9 21 6.4 17.2 23.5 28.1
Enter Route Name Like (UniversityRoad Etc) from above highlighted Routes
Saffora
Tell In Yes/No If You Need Ac Or Not
Yes
Enter Transport name Nadeem/Zulfiqar
Zulfiqar
Succesfully Registered
1
file Sucesfully Opened
__Booking Succesfully Confirmed__
__Verify Your Details BY looking At it__
--Student Details Is--
StudentId Is 23K-0532
Student Name is SAJJAD
Student ContactNumber Is 35677788
OUTLINE
TIMELINE


File Edit Selection View Go Run Terminal Help
A3 24K-0559
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Succesfully Registered
1
file Sucesfully Opened
__Booking Succesfully Confirmed__
__Verify Your Details BY looking At it__
--Student Details Is--
StudentId Is 23K-0532
Student Name is SAJJAD
Student ContactNumber Is 35677788
Student BatchNo Is 2023
Student SemesterYear Is fall2023
Route Number Is SafforaAnd Stop Name Is Nazimabad And Card Number Is 02
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Faculty
Enter Name
mrsamikhan
Enter Date of join
22-04-2021
Enter Email
sami@nu.edu.pk
Enter contact number
367775567
Enter fees
20009
Enter Route no
Dha
Enter salary
20009
Enter Details To Do Booking. Booking Number is : 0
Enter Month of Booking
4
Enter Date of booking
8
Enter Fee
20009
Enter In Yes/No if You need Ac
No
Payment Susssfully made
Enter Booking Name
SAMIX
Succesfully!. Booking Proceeded Booking Number is 1 and Booking Name is SAMIX
Booking Succesfull
Enter Booking Name write same name
OUTLINE
TIMELINE
0 0

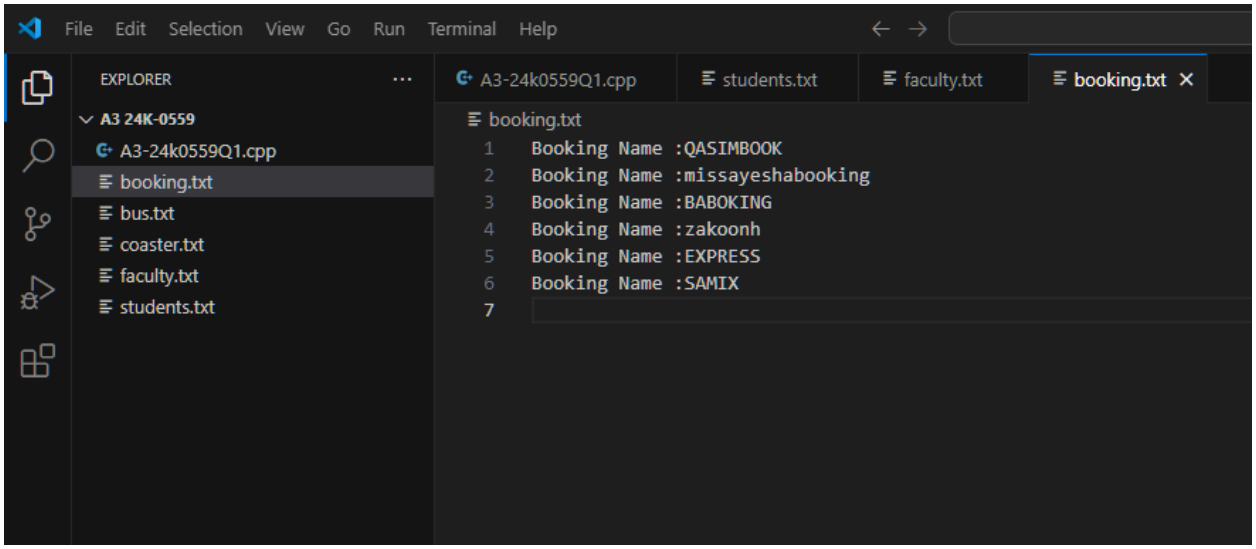

File Edit Selection View Go Run Terminal Help
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
A3-24k0559Q1.cpp > User
1 //24k-0559 Bazil-Uddin-Khan
2 #include<iostream>
3 #include<fstream>
4 #include<string>
5 #include<exception>
6 using namespace std;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
20009
Enter In Yes/No if You need Ac
No
Payment Susssfully made
Enter Booking Name
SAMIX
Succesfully!. Booking Proceeded Booking Number is 1 and Booking Name is SAMIX
Booking Succesfull
Enter Booking Name write same name
SAMIX
Tell In Yes/No If You Need Ac Or Not
Yes
Payment Susssfully made
Enter Transport name Nadeem/Zulfiqar
Nadeem
Teacher Succesfully Registered
Name is mrsamikhan
__Booking Succesfully Confirmed__
__Verify Your Details BY looking At it__
__Teacher Details Is__
Teacher Name Is mrsamikhan
Teacher Email Is sami@nu.edu.pk
Teacher Date Of =ƒôà Joining Is 22-04-2021
Teacher Fees Is 20009
Teacher Number Is 367775567
__Processing For Booking__And Savi g Data__
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Exd
Wrong Retry!
__Welcome Student/Faculty for going to register route__
__Enter You Are Student/Faculty/AddDriver/Exit__
Exit
Thanks For Being With Us!.
PS C:\Users\DELL\Pictures\A3 24K-0559>

FILES PICTURE:

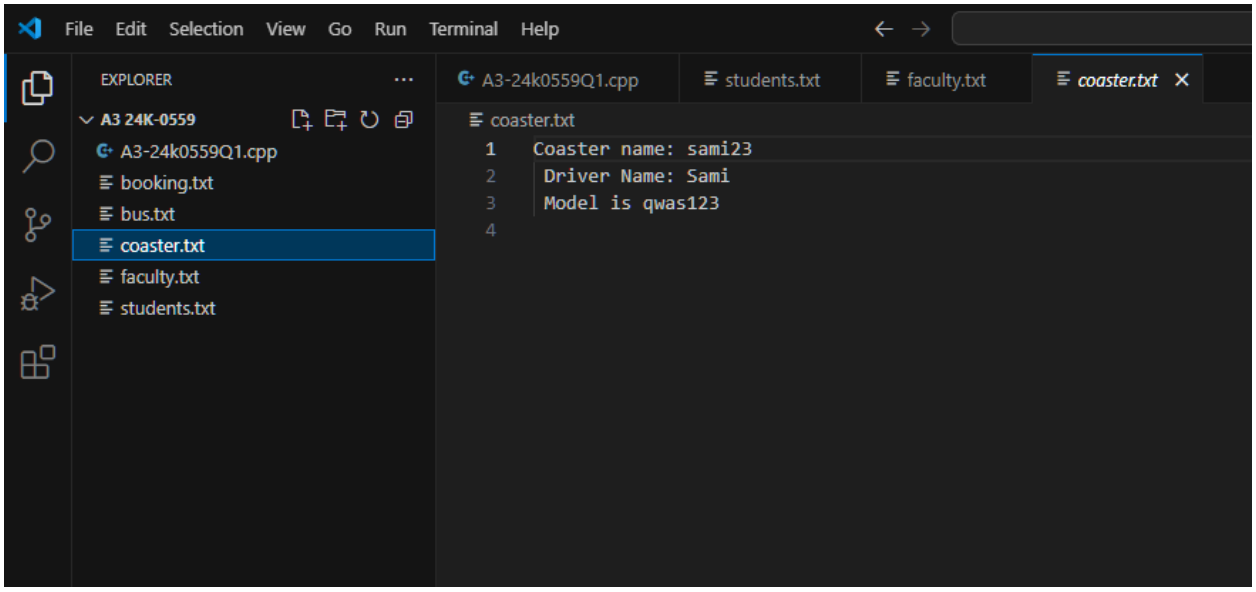
students.txt:



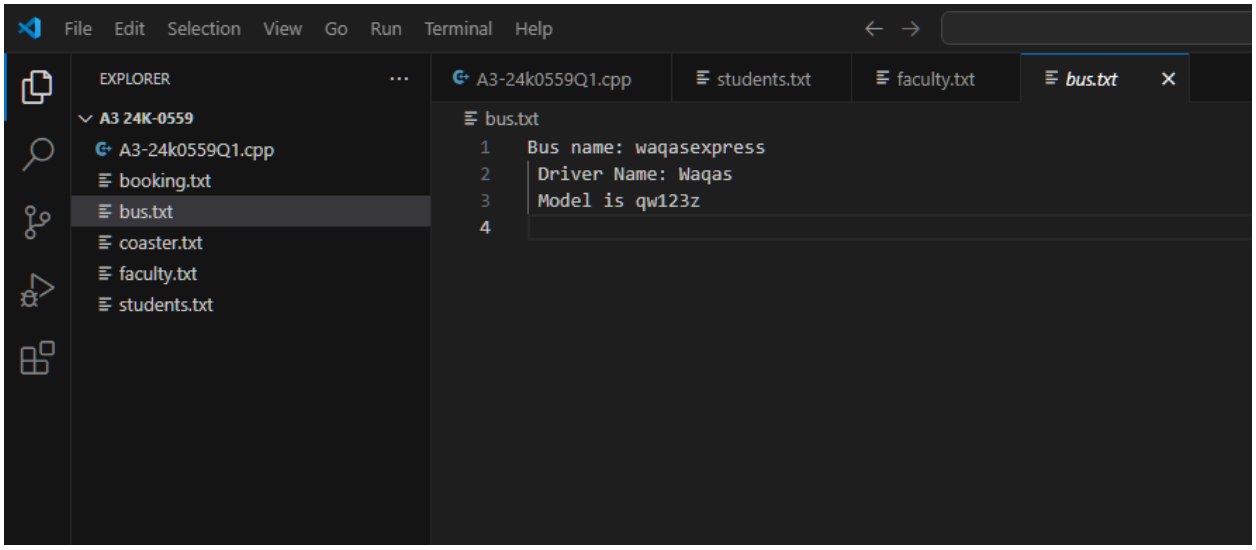
booking.txt:



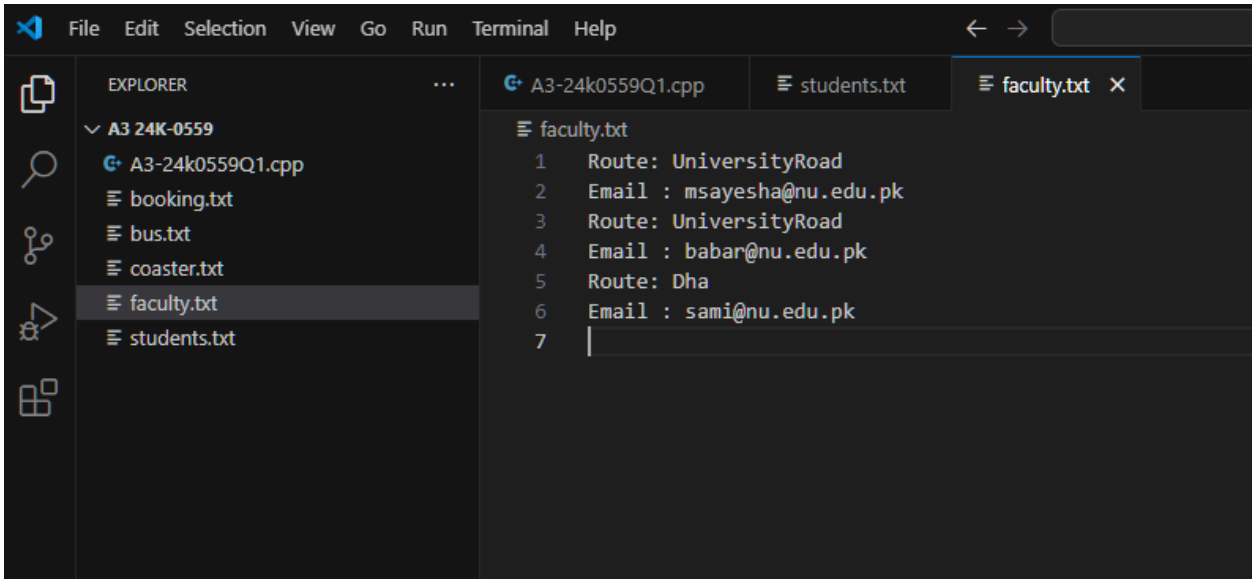
coaster.txt:



bus.txt:



faculty.txt:


File Edit Selection View Go Run Terminal Help
EXPLORER
A3 24K-0559
A3-24k0559Q1.cpp
booking.txt
bus.txt
coaster.txt
faculty.txt
students.txt
A3-24k0559Q1.cpp
students.txt
faculty.txt
faculty.txt
1 Route: UniversityRoad
2 Email : msayesha@nu.edu.pk
3 Route: UniversityRoad
4 Email : babar@nu.edu.pk
5 Route: Dha
6 Email : sami@nu.edu.pk
7