University of Karachi

Department of Computer Science (UBIT)

Project title: Web Mining & Knowledge Discovery

Final year project prepared by MCS final 4th semester batch 2020 students:

Muhammad Usman Khan EP-19101056.
&
Muneeb Ashfaq EP-19101059.

Supervisor: Professor Khalid Jamal

Contents

Chapter 01

Introduction

Web Mining

Web Mining is the process of Data Mining techniques to automatically discover and extract information from Web documents and services. The use of web mining typically involves several steps: collecting data, selecting the data before processing, knowledge discovery and analysis

The main purpose of web mining is discovering useful information from the World-Wide Web and its usage patterns. Web mining is used to predict user behavior. For example, create filters or search fields for user to search for required information.

Categories of Web Mining:

1. **Web content mining:** This is the process of mining useful information from the contents of Web pages and Web documents, which are mostly text, images and audio/video files. Techniques used in this discipline have been heavily drawn from natural language processing (NLP) and information retrieval.

2. **Web structure mining:** This is the process of analyzing the nodes and connection structure of a website through the use of graph theory. There are two things that can be obtained from this:
   the structure of a website in terms of how it is connected to other sites and the document structure of the website itself, as to how each page is connected.

3. **Web usage mining:** This is the process of extracting patterns and information from server logs to gain insight on user activity including where the users are from, how many clicked what item on the site and the types of activities being done on the site.

Knowledge Discovery

- Knowledge Discovery in Databases (KDD) is the non-trivial extraction of implicit, previously unknown and potentially useful knowledge from data.
- Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data. Process of semi-automatically analyzing large databases to find patterns that are:
  - Valid: The patterns hold in general.
  - Novel: We did not know the pattern beforehand.
  - Useful: We can devise actions from the patterns.
  - Understandable: We can interpret and comprehend the patterns.
- Why? Find trends and correlations in existing databases, that help you change structures and processes in human organizations:
  - to be more effective
  - to save time
  - to make more money
  - to improve product quality etc.

Applications of Web Mining & Knowledge Discovery

1. **Banking: Bank applications**
   - Banking applications

2. **Customer relationship management:**
   - identify those who are likely to leave for a competitor.

3. **Targeted marketing:**
   - identify likely responders to promotions

4. **Fraud detection: telecommunications, financial transactions**
   - from an online stream of event identify fraudulent events

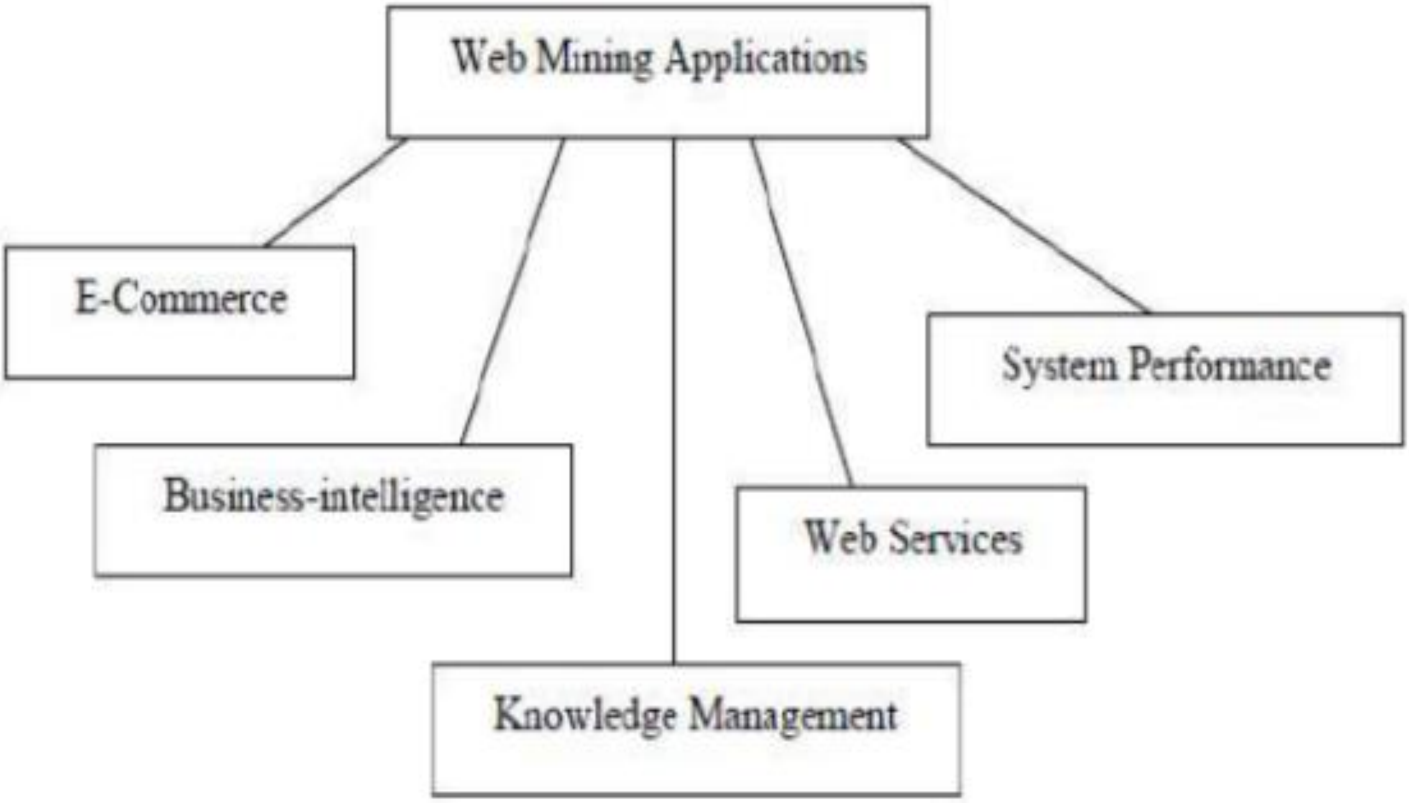
5. **Manufacturing and production:**
   - automatically adjust knobs when process parameter change

6. **Medicine: disease outcome, effectiveness of treatments**
   - analyze patient disease history: find relationship between diseases

7. **Molecular/Pharmaceutical: identify new drugs**

8. **Scientific data analysis:**
   - identify new galaxies by searching for sub clusters

9. **Web site/store design and promotion:**
   - find affinity of visitor to pages and modify layout.


Web Mining Applications
E-Commerce
Business-intelligence
Knowledge Management
Web Services
System Performance

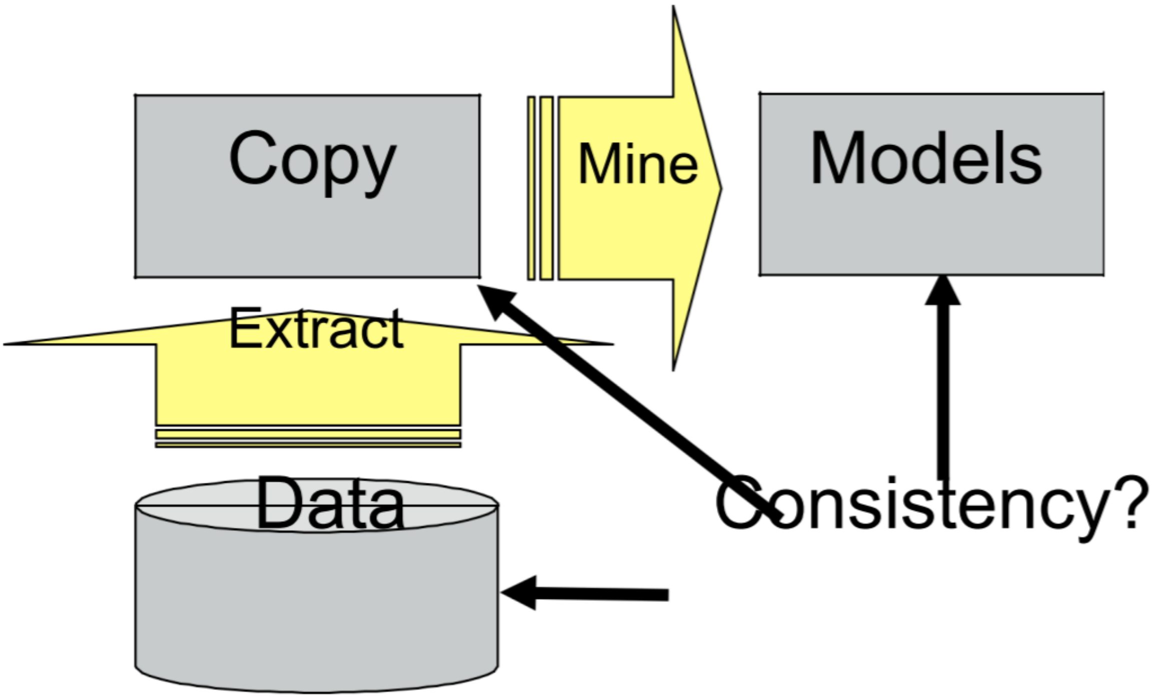
Chapter 02

Web Mining and Knowledge Process Steps

Why we need it now?

- Data is being produced
- Data is being warehoused
- The computing power is available
- The computing power is affordable
- The competitive pressures are strong
- Commercial products are available

Why Integrate Web Mining & Knowledge Discovery with DB'S?



Integration Objectives

From Analysts (users) point of view

- Avoid isolation of querying from mining
  - –Difficult to do “ad-hoc” mining
- Provide simple programming approach to creating and using DM models

From Knowledge Discovery Vendors point of view:

- Make it possible to add new models
- Make it possible to add new, scalable algorithms

Why Knowledge Discovery (KD)?

Can we solve all problems with KD?

Four questions to be answered:

1. Can the problem clearly be defined?
2. Does potentially meaningful data exists?
3. Does the data contain hidden knowledge or useful only for reporting purposes?
4. Will the cost of processing the data will be less then the likely increase in profit from the knowledge gained from applying any data mining project

Knowledge Discovery:

- Knowledge Discovery in Databases (KDD) is the non-trivial extraction of implicit, previously unknown and potentially useful knowledge from data.
- Knowledge Discovery is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data. Process of semi-automatically analyzing large databases to find patterns that are:
  - Valid: The patterns hold in general.
  - Novel: We did not know the pattern beforehand.
  - Useful: We can devise actions from the patterns.
  - Understandable: We can interpret and comprehend the patterns.
- Why? Find trends and correlations in existing databases, that help you change structures and processes in human organizations:
- be more effective

- to save time
- to make more money
- to improve product quality etc.

KDD Process:

- Problem formulation
- Data collection
  - subset data: sampling might hurt if highly skewed data
  - feature selection: principal component analysis, heuristic search
- Pre-processing: cleaning
  - name/address cleaning, different meanings (annual, yearly), duplicate removal, supplying missing values
- Transformation:
  - map complex objects e.g. time series data to features e.g. frequency
- Choosing mining task and mining method
- Result evaluation and visualization

Web Mining and Knowledge Discovery Process Steps (detailed)

1. **Goal identification:**
   - Define problem
   - relevant prior knowledge and goals of application

2. **Creating a target data set:** data selection

3. **Data preprocessing:** (may take 60%-80% of effort!)
   - removal of noise or outliers
   - strategies for handling missing data fields
   - accounting for time sequence information

4. **Data reduction and transformation:**
   - Find useful features, dimensionality/variable reduction, invariant representation

5. **Data Mining:**
   - **Choosing functions of data mining:**
     - summarization, classification, regression, association, clustering
   - **Choosing the mining algorithm(s):**
     - which models or parameters
   - **Search for patterns.**

6. **Presentation and evaluation:**
   - visualization, transformation, removing redundant patterns, etc.

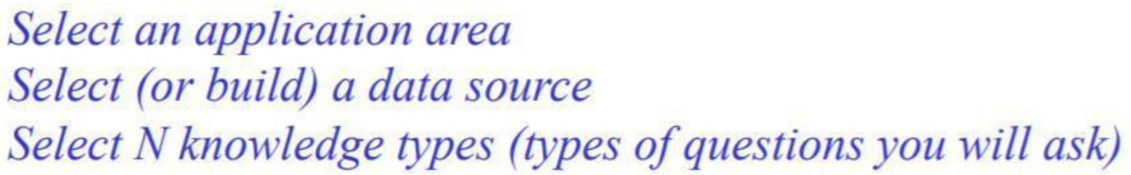
7. **Taking action:**
   - incorporating into the performance system
   - documenting
   - reporting to interested parties

8. **Structure of Web According to Database fields:**
   - Display of data in a required way

- Submitting of data for insertion and update.
- Get data, prepare report and print it or export it.

Mechanism:

Web mining and Knowledge Discovery mechanisms are not application-specific, they depend on the target knowledge type:



Select an application area
Select (or build) a data source
Select N knowledge types (types of questions you will ask)

Advantages of Data Mining

Marketing / Retail: Data mining helps marketing companies build models based on historical data to predict who will respond to the new marketing campaigns such as direct mail, online marketing campaign…etc. Through the results, marketers will have an appropriate approach to selling profitable products to targeted customers. Data mining brings a lot of benefits to retail companies in the same way as marketing. Through market basket analysis, a store can have an appropriate production arrangement in a way that customers can buy frequent buying products together with pleasant. In addition, it also helps the retail companies offer certain discounts for particular products that will attract more customers.

Finance / Banking: Data mining gives financial institutions information about loan information and credit reporting. By building a model from historical customer's data, the bank, and financial institution can determine good and bad loans. In addition, data mining helps banks detect fraudulent credit card transactions to protect credit card's owner.

Manufacturing: By applying data mining in operational engineering data, manufacturers can detect faulty equipment and determine optimal control parameters. For example, semiconductor manufacturers have a challenge that even the conditions of manufacturing environments at different wafer production plants are similar, the quality of wafer are a lot the same and some for unknown reasons even has defects. Data mining has been applying to determine the ranges of control parameters that lead to the production of the golden wafer. Then those optimal control parameters are used to manufacture wafers with desired quality.

Governments: Data mining helps government agency by digging and analyzing records of the financial transaction to build patterns that can detect money laundering or criminal activities.

Conclusion:

Data mining brings a lot of benefits to businesses, society, governments as well as the individual. However, privacy, security, and misuse of information are the big problems if they are not addressed and resolved properly.

Advantages of Knowledge Discovery

Marketing:In marketing, the primary application is database marketing systems,which analyze customer databases to identify different customer groups and forecast their behavior. Business Week (Berry 1994) estimated that over half of all retailers are using or planning to use database marketing, and those who do use it have good results; for example, American Express reports a 10 to 15 percent increase in credit-card use. Another notable marketing application is market-basket analysis (Agrawal et al. 1996) systems, which find patterns such as, “If customer bought X, he/she is also likely to buy Y and Z.” Such patterns are valuable to retailers.

Investment: Numerous companies use data mining for investment, but most do not describe their systems. One exception is LBS Capital Management. Its system uses expert systems, neural nets, and genetic algorithms to manage portfolios totaling $600 million; since its start in 1993, the system has outperformed the broad stock market (Hall, Mani, and Barr 1996).

Fraud detection: HNC Falcon and Nestor PRISM systems are used for monitoring credit card fraud, watching over millions of accounts .The FAIS system (Senator et al. 1995), from the U.S. Treasury Financial Crimes Enforcement Network, is used to identify financial transactions that might indicate money laundering activity.

Manufacturing: The CASSIOPEE troubleshooting system, developed as part of a joint venture between General Electric and SNECMA, was applied by three major European airlines to diagnose and predict problems for the Boeing 737. To derive families of faults, clustering methods are used. CASSIOPEE received the European first prize for innovtive applications (Manago and Auriol 1996).

Telecommunications: The telecommunications alarm-sequence analyzer (TASA) was built in cooperation with a manufacturer of telecommunications equipment and three telephone networks (Mannila, Toivonen, and Verkamo 1995). The system uses a novel framework for locating frequently occurring alarm episodes from the alarm stream and presenting them as rules. Large sets of discovered rules can be explored with flexible information-retrieval tools supporting interactivity and iteration. In this way, TASA offers pruning, grouping, and ordering tools to refine the results of a basic brute-force search for rules.

Data cleaning: The MERGE-PURGE system was applied to the identification of duplicate welfare claims (Hernandez and Stolfo 1995).

It was used successfully on data from the Welfare Department of the State of Washington. In other areas, a well-publicized system is IBM's ADVANCED SCOUT, a specialized data-mining system that helps National Basketball Association (NBA) coaches organize and interpret data from NBA games (U.S. News 1995). ADVANCED SCOUT was used by several of the NBA teams in 1996, including the Seattle Supersonics, which reached the NBA finals. Finally, a novel and increasingly important type of discovery is one based on the use of intelligent agents to navigate through an information-rich environment. Although the idea of active triggers has long been analyzed in the database field, really successful applications of this idea appeared only with the advent of the Internet. These systems ask the user to specify a profile of interest and search for related information among a wide variety of public-domain and proprietary sources. For example, FIREFLY is a personal music-recommendation agent: It asks a user his/her opinion of several music pieces and then suggests other music that the user might like (<http://www.ffly.com/>). CRAYON (<http://crayon.net/>) allows users to create their own free newspaper (supported by ads); NEWSHOUND (<http://www.sjmercury.com/hound/>) from the *San Jose Mercury News* and FARCAST (http://www.farcast.com/) automatically search information from a wide variety of sources, including newspapers and wire services, and e-mail relevant documents directly to the user.

Conclusion:

These are just a few of the numerous such systems & there advantages that use KDD techniques to automatically produce useful information from large masses of raw data.

Chapter 03
RDBMS in Knowledge Discovery

Why RDBMS in Knowledge Discovery
Software packages providing a whole set of data mining and machine learning algorithms are attractive because they allow experimentation with many kinds of algorithms in an easy setup. However, these packages are often based on main-memory data structures, limiting the amount of data they can handle.

Advantages of RDBMS

Data Structure: The table format is simple and easy for database users to understand and use. RDBMSs provide data access using a natural structure and organization of the data. Database queries can search any column for matching entries.

Multi-User Access: RDBMSs allow multiple database users to access a database simultaneously. Built-in locking and transactions management functionality allow users to access data as it is being changed, prevents collisions between two users updating the data, and keeps users from accessing partially updated records.

Privileges: Authorization and privilege control features in an RDBMS allow the database administrator to restrict access to authorized users, and grant privileges to individual users based on the types of database tasks they need to perform. Authorization can be defined based on the remote client IP address in combination with user authorization, restricting access to specific external computer systems.

Network Access: RDBMSs provide access to the database through a server daemon, a specialized software program that listens for requests on a network, and allows database clients to connect to and use the database. Users do not need to be able to log in to the physical computer system to use the database, providing convenience for the users and a layer of security for the database. Network access allows developers to build desktop tools and Web applications to interact with databases.

Speed: The relational database model is not the fastest data structure. RDBMS advantages, such as simplicity, make the slower speed a fair trade-off. Optimizations built into an RDBMS, and the design of the databases, enhance performance, allowing RDBMSs to perform more than fast enough for most applications and data sets. Improvements in technology, increasing processor speeds and decreasing memory and storage costs allow systems administrators to build incredibly fast systems that can overcome any database performance shortcomings.

Maintenance: RDBMSs feature maintenance utilities that provide database administrators with tools to easily maintain, test, repair and back up the databases housed in the system. Many of the functions can be automated using built-in automation in the RDBMS, or automation tools available on the operating system.

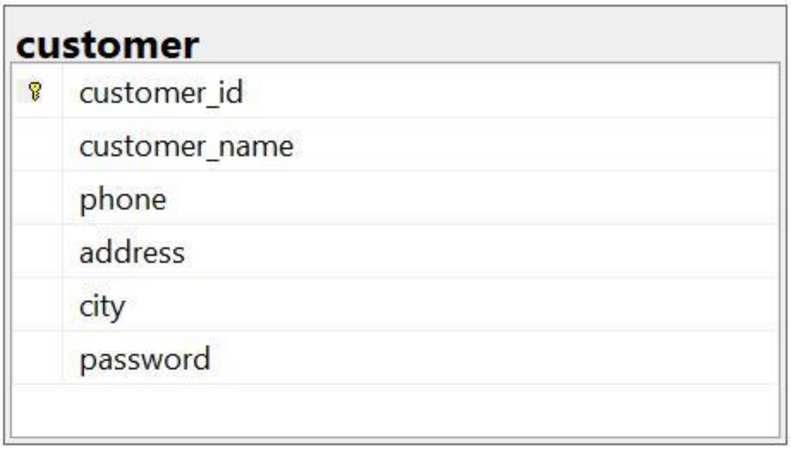
Language: RDBMSs support a generic language called "Structured Query Language" (SQL). The SQL syntax is simple, and the language uses standard English language keywords and phrasing, making it fairly intuitive and easy to learn. Many RDBMSs add non-SQL, database-specific keywords, functions and features to the SQL language.

Chapter 03

Database for Parcel Delivery with the use of Web Mining and Knowledge Discovery

Database for Parcel Delivery Application Based on Web Mining and Knowledge Discovery Concept (Not completely developed):

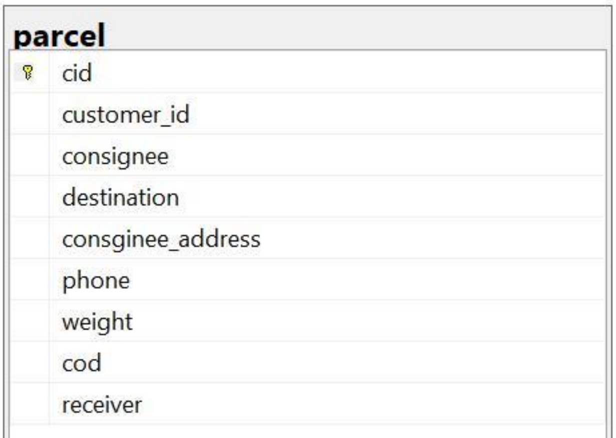
1. Table for Customer Registration



	customer
🔑	customer_id
	customer_name
	phone
	address
	city
	password

- customer_id is the primary key

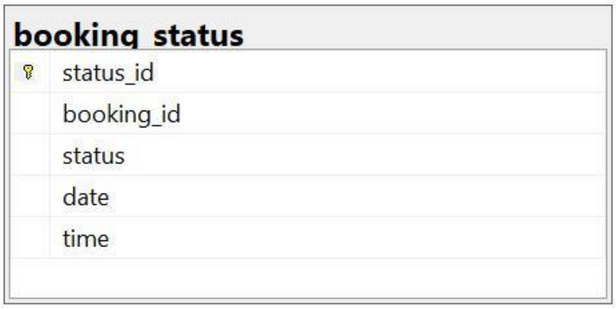
2. Table for Parcel Booking



	parcel
🔑	cid
	customer_id
	consignee
	destination
	consginee_address
	phone
	weight
	cod
	receiver

- cid is the primary key
- customer_id is the foreign key

3. Table for Booking/Parcel Delivery Status

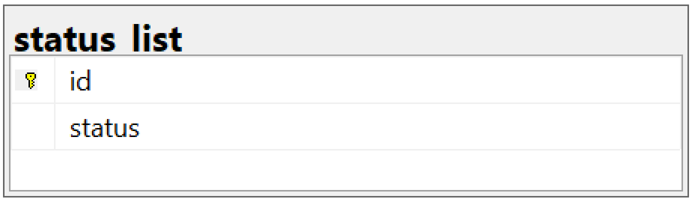


booking_status

🔑	status_id
	booking_id
	status
	date
	time

- status_id is the primary key
- booking_id is the foreign key
- status is the foreign key

4. Table for Status List

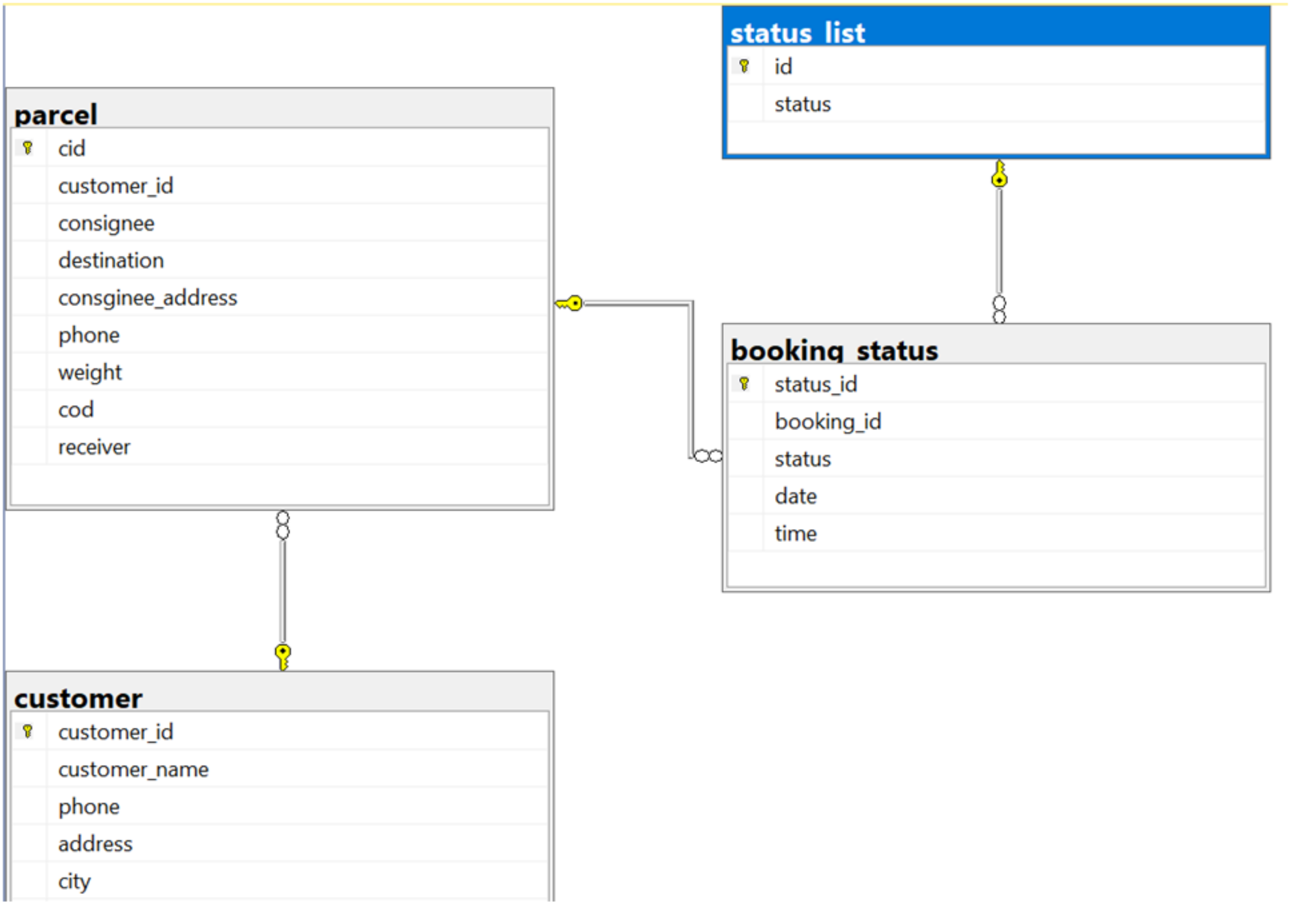


status_list

🔑	id
	status

1. id is the primary key

Entity Relationship Diagram



Chapter 04

Application Development with Web Mining and Knowledge Discovery

Introduction

As discussed above we will develop an application for Parcel Delivery System using the Web Mining and Knowledge Discovery. We will avoid the insertion of incorrect data, remove the data redundancy, use the complex queries and filter the result, display the result or data what use wants to see.

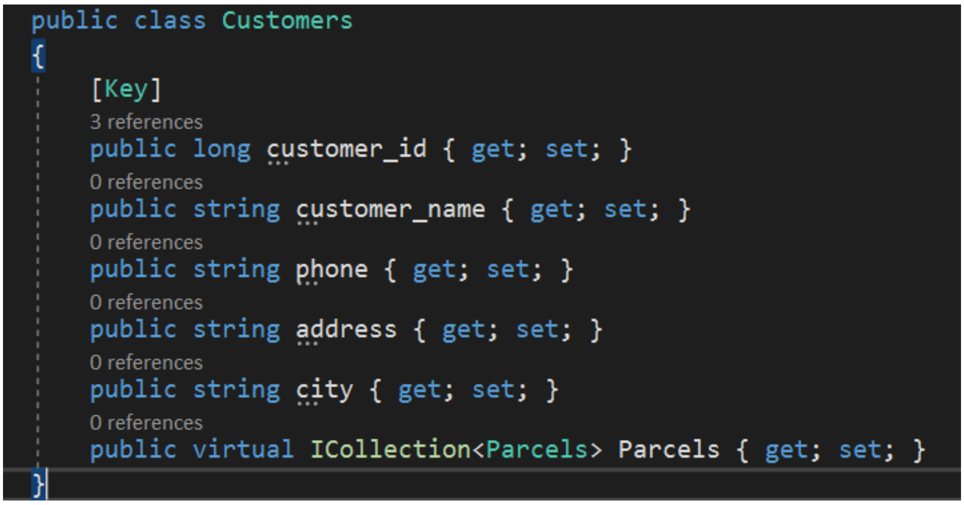
Tools

C#, Asp.Net Core, Entity Framework Core, SQL Server 2019.

Table Models

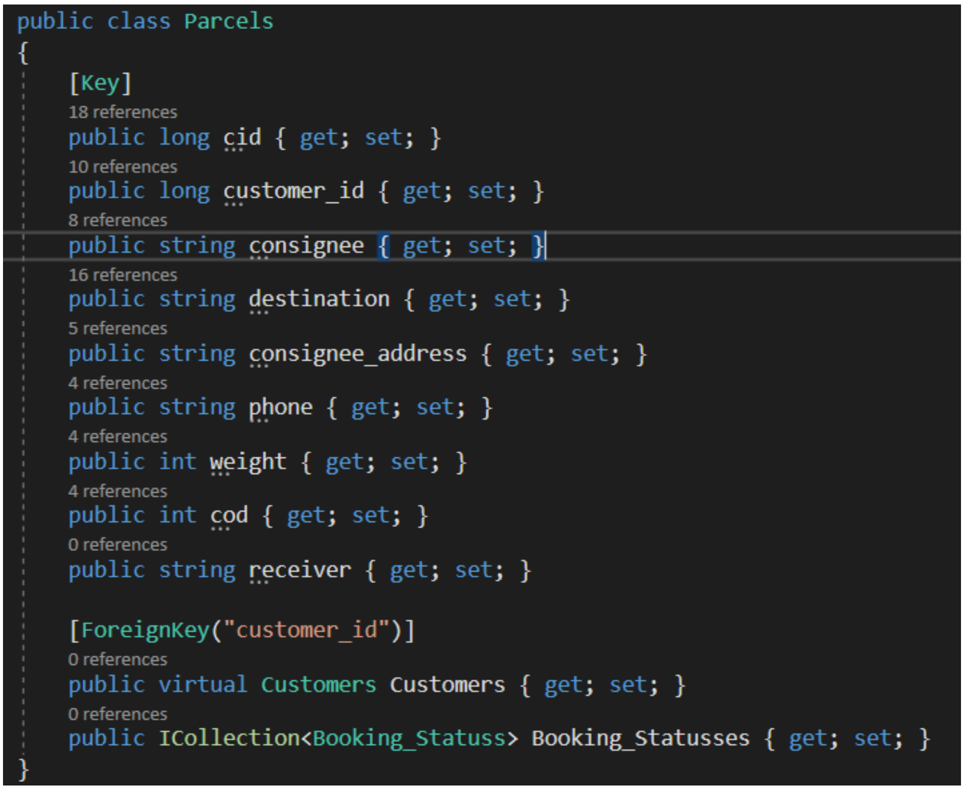
As discussed above the database diagrams, we will develop the Models for application.

1. Customer Model Class



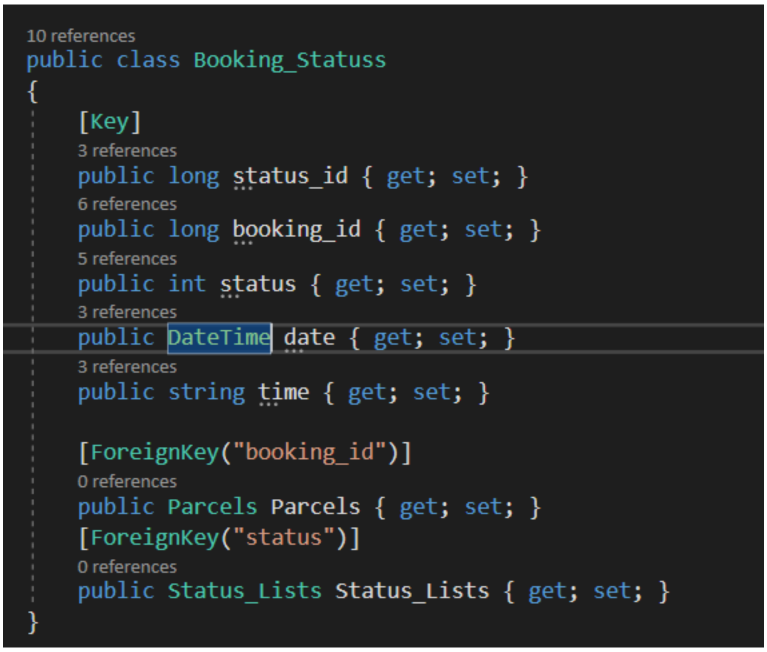
```
public class Customers
{
    [Key]
    3 references
    public long customer_id { get; set; }
    0 references
    public string customer_name { get; set; }
    0 references
    public string phone { get; set; }
    0 references
    public string address { get; set; }
    0 references
    public string city { get; set; }
    0 references
    public virtual ICollection<Parcels> Parcels { get; set; }
}
```

2. Parcel Model Class



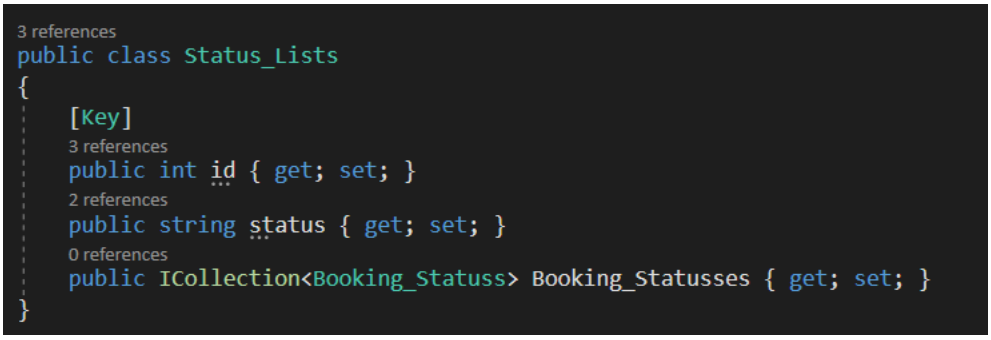
```
public class Parcels
{
    [Key]
    18 references
    public long cid { get; set; }
    10 references
    public long customer_id { get; set; }
    8 references
    public string consignee { get; set; }
    16 references
    public string destination { get; set; }
    5 references
    public string consignee_address { get; set; }
    4 references
    public string phone { get; set; }
    4 references
    public int weight { get; set; }
    4 references
    public int cod { get; set; }
    0 references
    public string receiver { get; set; }

    [ForeignKey("customer_id")]
    0 references
    public virtual Customers Customers { get; set; }
    0 references
    public ICollection<Booking_Statuss> Booking_Statusses { get; set; }
}
```

3. Parcel/Booking Status Model Class



```
10 references
public class Booking_Statuss
{
    [Key]
    3 references
    public long status_id { get; set; }
    6 references
    public long booking_id { get; set; }
    5 references
    public int status { get; set; }
    3 references
    public DateTime date { get; set; }
    3 references
    public string time { get; set; }

    [ForeignKey("booking_id")]
    0 references
    public Parcels Parcels { get; set; }
    [ForeignKey("status")]
    0 references
    public Status_Lists Status_Lists { get; set; }
}
```

4. Status List Model Class



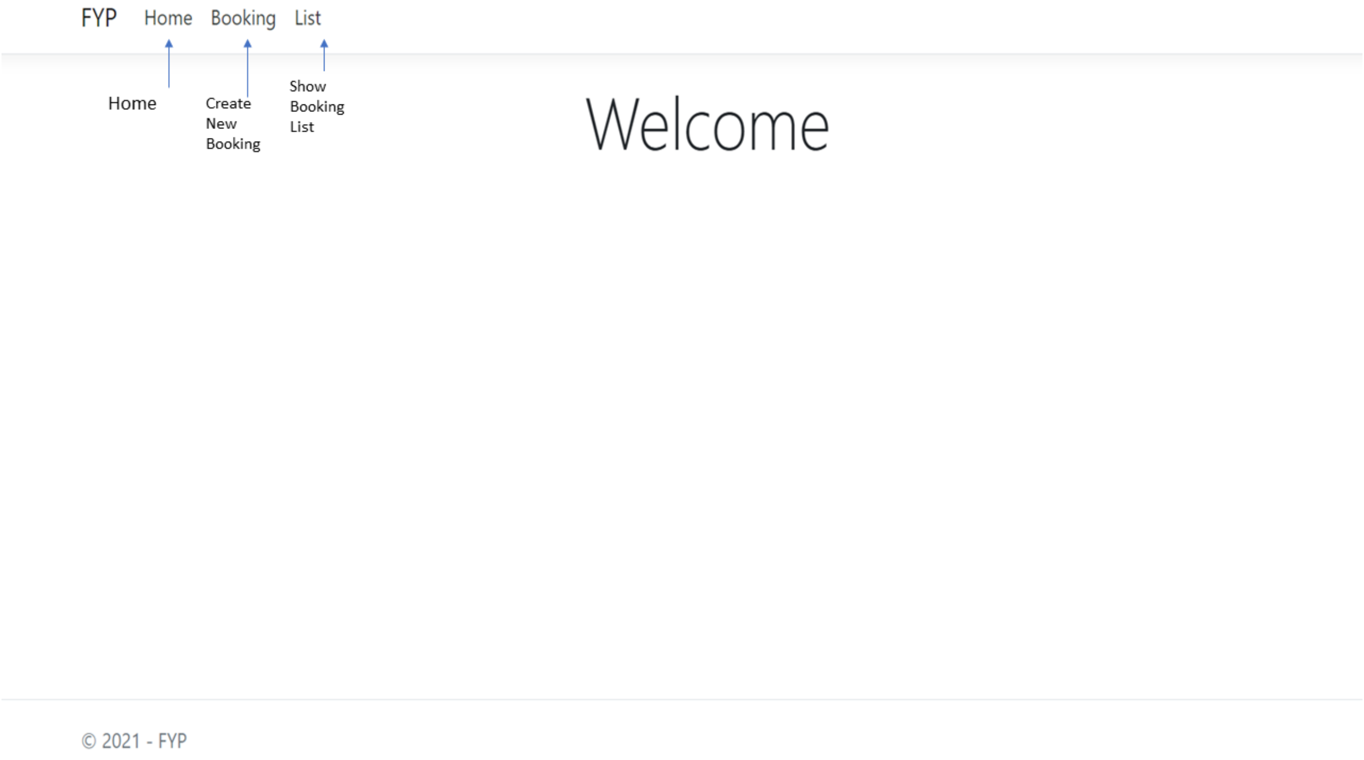
```
3 references
public class Status_Lists
{
    [Key]
    3 references
    public int id { get; set; }
    2 references
    public string status { get; set; }
    0 references
    public ICollection<Booking_Statuss> Booking_Statusses { get; set; }
}
```

User Interface with Features

One of the main point about web mining is that how we display content and features on web. Web content and style should be user friendly. Should be PWA.

1. Home Page

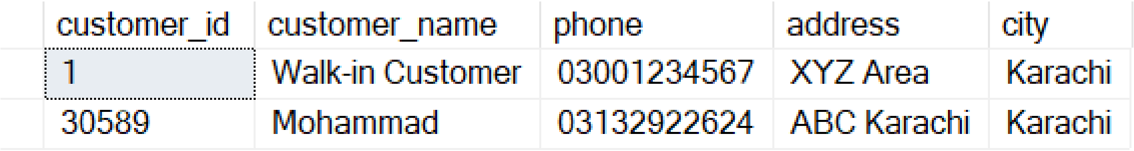
Main Home is shown as below



Labels show navigation bar with relevant links.
a. To create New Booking, click on Booking.
b. To see booking list, click on List.

2. Customers Table

Customer must be registered before creating a new parcel booking. Following is the details of customers.

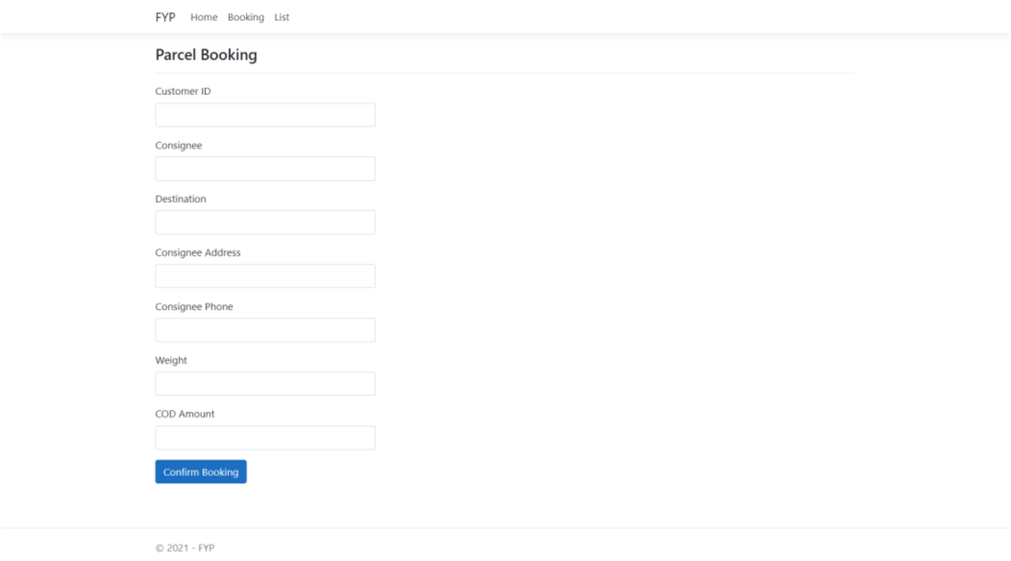


customer_id	customer_name	phone	address	city
1	Walk-in Customer	03001234567	XYZ Area	Karachi
30589	Mohammad	03132922624	ABC Karachi	Karachi

a. We have created a default customer for walk-in-customer type with Id: 1.
b. In case of other customers rather that a walk-in-customer, details of the Id: 30589 are shown above in row#02.

3. Parcel Booking Form

Keeping the points of web mining and knowledge discovery, only meaningful data should be inserted into the database. Booking form on front end must be relevant to database. Here we want to check that customer is registered or not before creating new booking for parcel. Backend techniques must be applied to check user registration and allow meaningful insertion of data.
Below is the form for creating a booking.



Parcel Booking

Customer ID

Consignee

Destination

Consignee Address

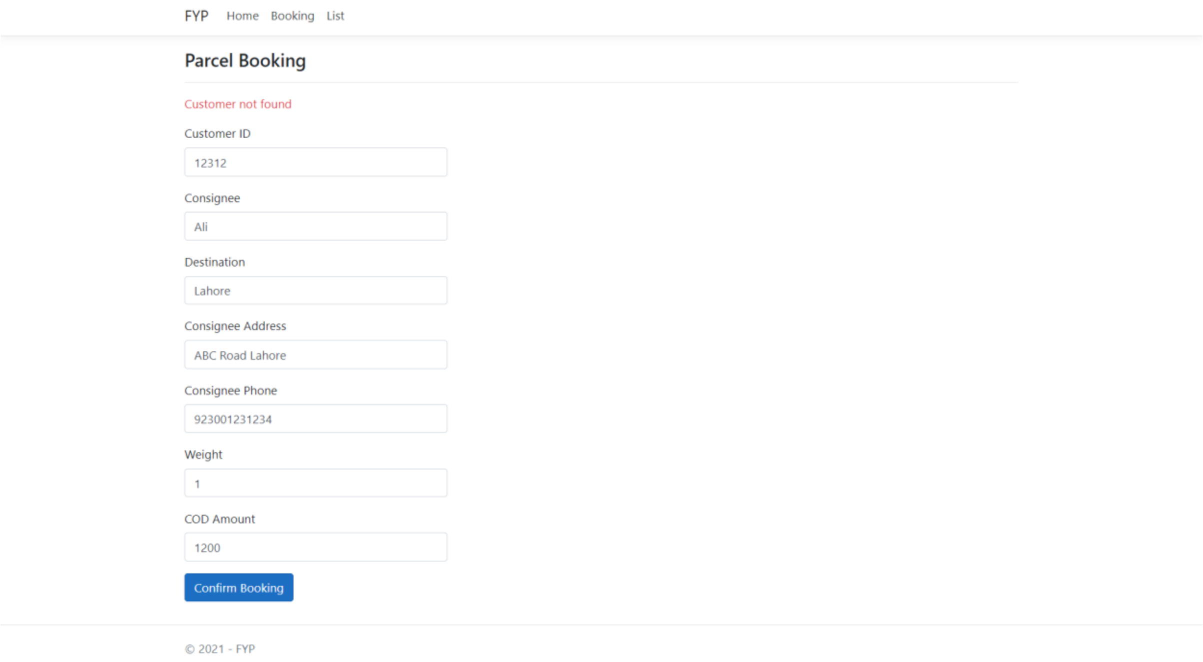
Consignee Phone

Weight

COD Amount

© 2021 - FYP

In case of wrong/not registered customer ID, no customer found message will be displayed.



Parcel Booking

Customer not found

Customer ID: 12312

Consignee: Ali

Destination: Lahore

Consignee Address: ABC Road Lahore

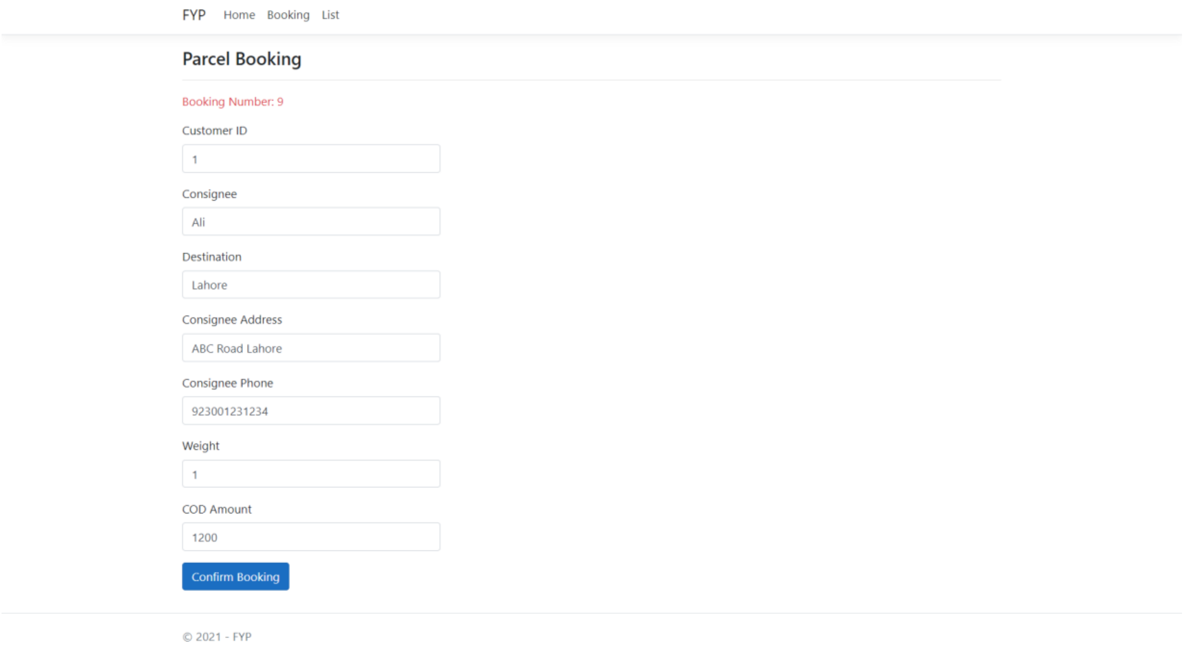
Consignee Phone: 923001231234

Weight: 1

COD Amount: 1200

© 2021 - FYP

In case of valid and registered customer, user will be registered. Not only parcel information will be inserted into parcel table, but also parcel status **Booking** will be inserted into **booking_ status** table automatically. These are necessary for web mining and knowledge discovery to automate system i.e. automatic update status but reduce data redundancy, insert valid data, avoid duplication and follow RDBMS structure.
Also return Booking/Order Number to track that order in future. It is necessary in web mining and KDD to return necessary details on web.



Parcel Booking

Booking Number: 9

Customer ID: 1

Consignee: Ali

Destination: Lahore

Consignee Address: ABC Road Lahore

Consignee Phone: 923001231234

Weight: 1

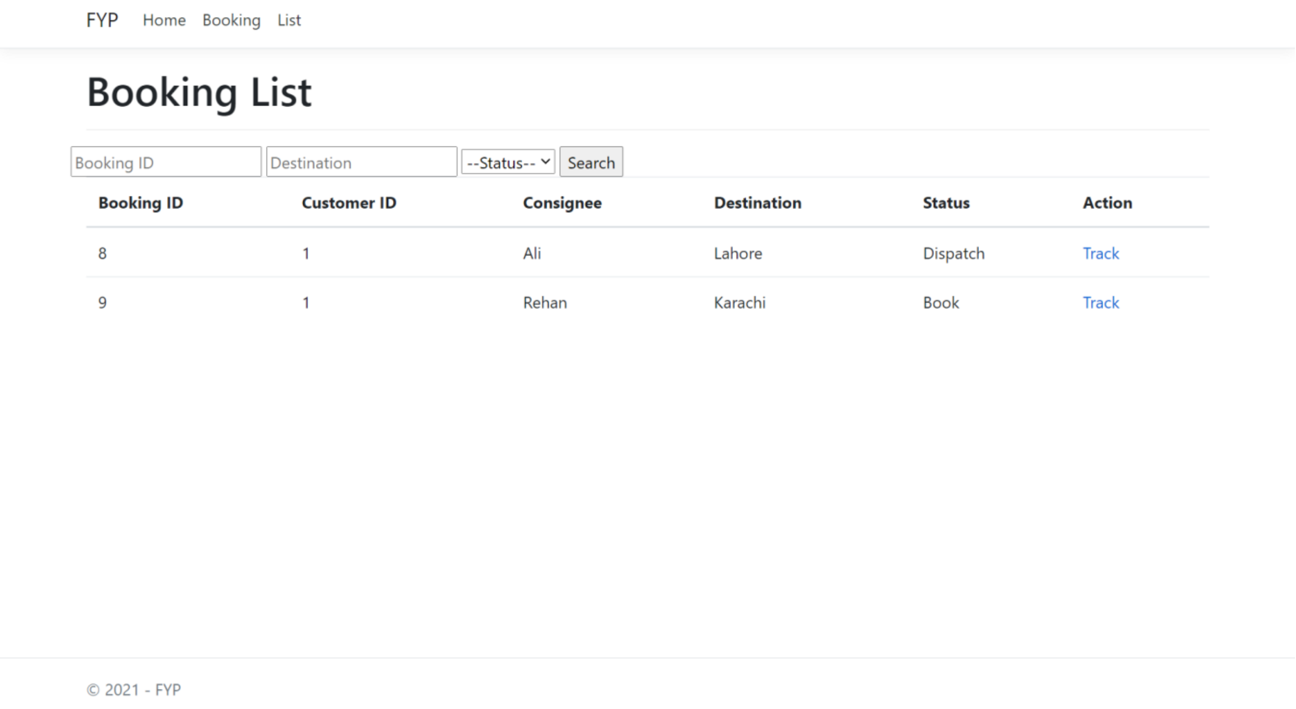
COD Amount: 1200

© 2021 - FYP

4. Get Booking List

It is necessary to show details and list of all orders/bookings on web in a such a way that user wants. Whether it’s a software for BI team or normal users, visibility of data in list or table is necessary. It is necessary in web mining and knowledge discovery to show relevant and usable data by using complex queries on backend. With these complex queries, get data from database, prepare view and display data in such a way that user wants.

Also, fields for applying filters are necessary, to select particular data. Below is table for displaying data with filter fields. All data without applying filters.

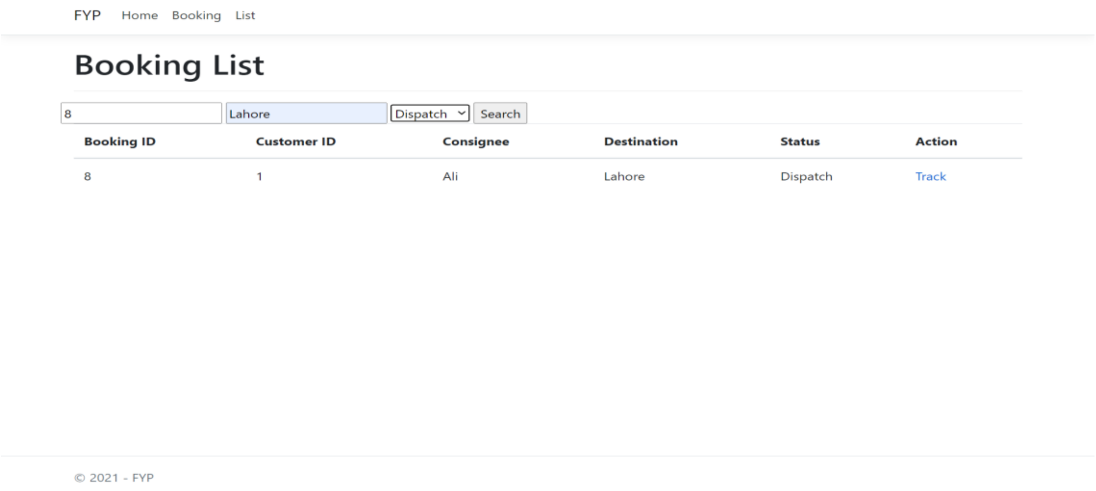


Booking List

Booking ID	Customer ID	Consignee	Destination	Status	Action
8	1	Ali	Lahore	Dispatch	Track
9	1	Rehan	Karachi	Book	Track

© 2021 - FYP

Now, apply filters and show particular data.



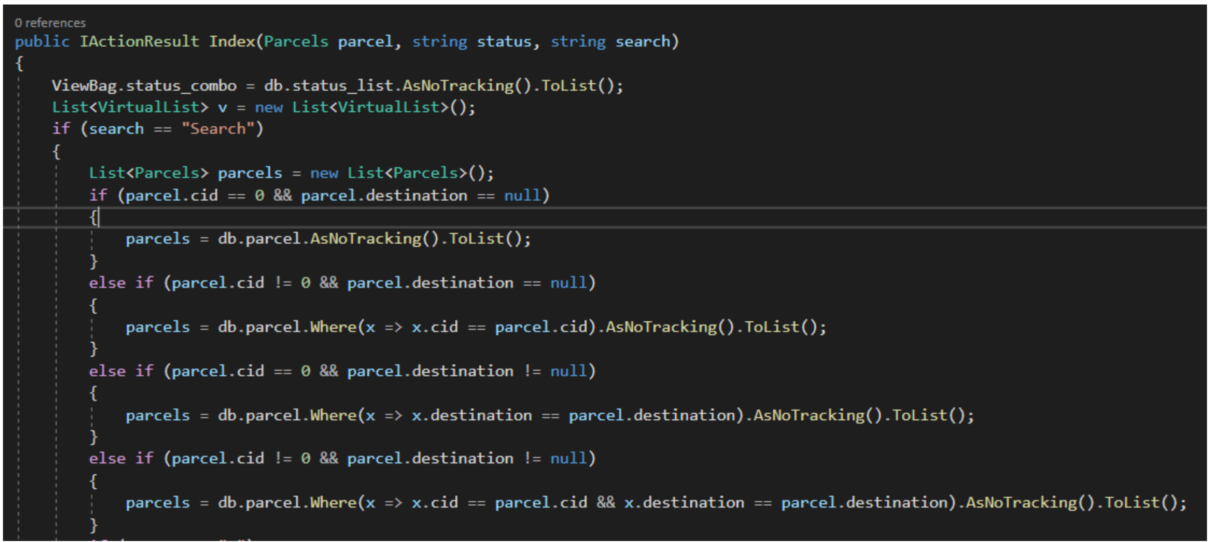
Booking List

8 Lahore Dispatch

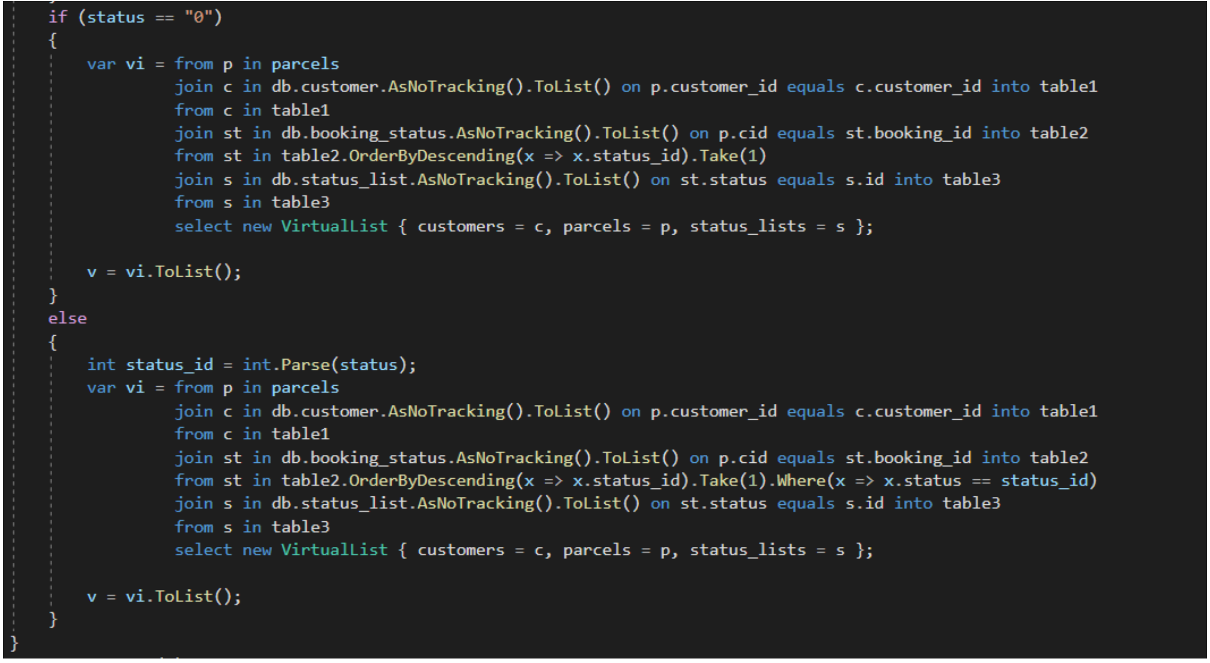
Booking ID	Customer ID	Consignee	Destination	Status	Action
8	1	Ali	Lahore	Dispatch	Track

© 2021 - FYP

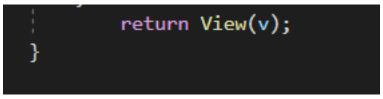
Following is the code for complex queries



```
0 references
public IActionResult Index(Parcels parcel, string status, string search)
{
    ViewBag.status_combo = db.status_list.AsNoTracking().ToList();
    List<VirtualList> v = new List<VirtualList>();
    if (search == "Search")
    {
        List<Parcels> parcels = new List<Parcels>();
        if (parcel.cid == 0 && parcel.destination == null)
        {
            parcels = db.parcel.AsNoTracking().ToList();
        }
        else if (parcel.cid != 0 && parcel.destination == null)
        {
            parcels = db.parcel.Where(x => x.cid == parcel.cid).AsNoTracking().ToList();
        }
        else if (parcel.cid == 0 && parcel.destination != null)
        {
            parcels = db.parcel.Where(x => x.destination == parcel.destination).AsNoTracking().ToList();
        }
        else if (parcel.cid != 0 && parcel.destination != null)
        {
            parcels = db.parcel.Where(x => x.cid == parcel.cid && x.destination == parcel.destination).AsNoTracking().ToList();
        }
```



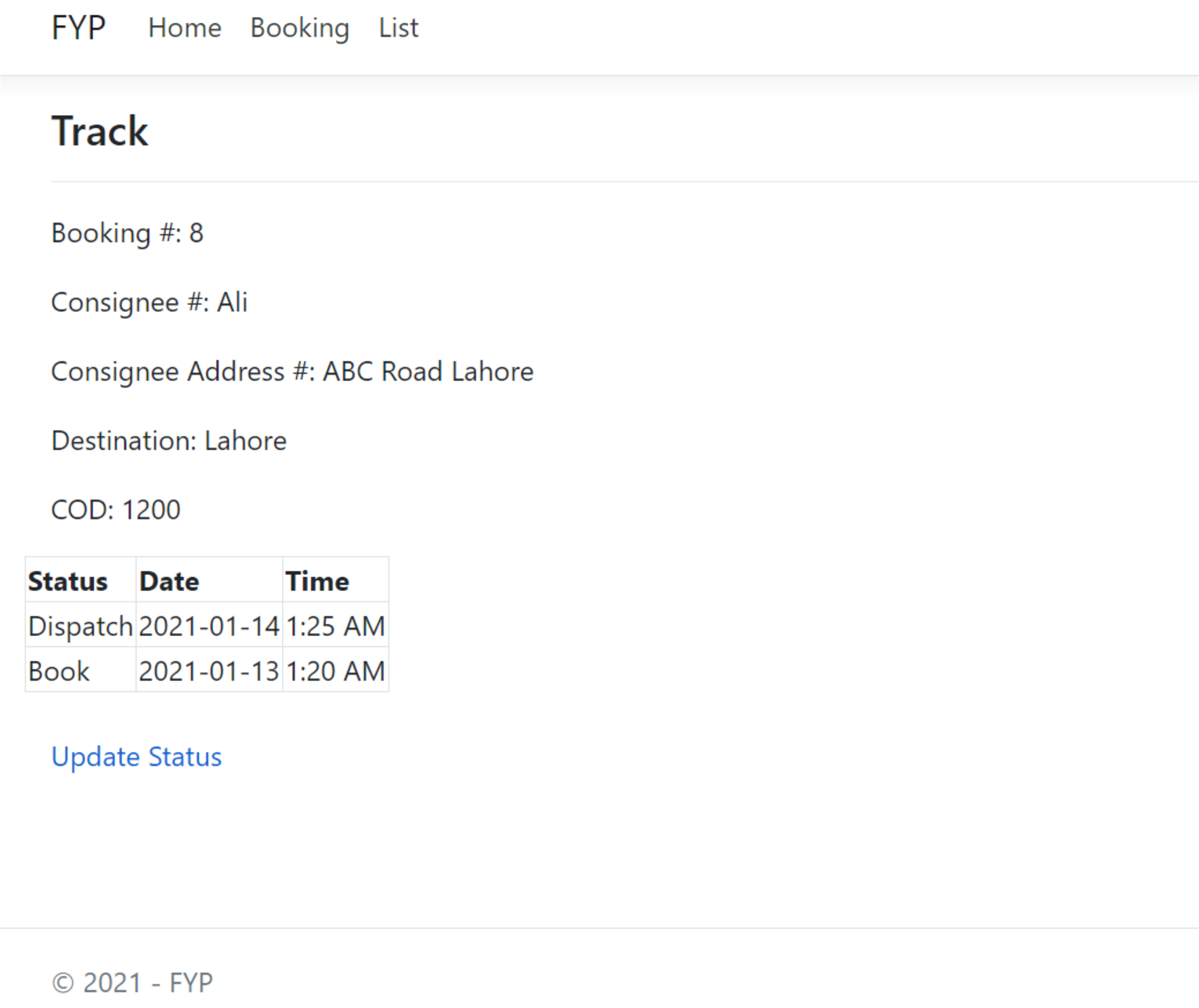
```
    if (status == "0")
    {
        var vi = from p in parcels
                 join c in db.customer.AsNoTracking().ToList() on p.customer_id equals c.customer_id into table1
                 from c in table1
                 join st in db.booking_status.AsNoTracking().ToList() on p.cid equals st.booking_id into table2
                 from st in table2.OrderByDescending(x => x.status_id).Take(1)
                 join s in db.status_list.AsNoTracking().ToList() on st.status equals s.id into table3
                 from s in table3
                 select new VirtualList { customers = c, parcels = p, status_lists = s };

        v = vi.ToList();
    }
    else
    {
        int status_id = int.Parse(status);
        var vi = from p in parcels
                 join c in db.customer.AsNoTracking().ToList() on p.customer_id equals c.customer_id into table1
                 from c in table1
                 join st in db.booking_status.AsNoTracking().ToList() on p.cid equals st.booking_id into table2
                 from st in table2.OrderByDescending(x => x.status_id).Take(1).Where(x => x.status == status_id)
                 join s in db.status_list.AsNoTracking().ToList() on st.status equals s.id into table3
                 from s in table3
                 select new VirtualList { customers = c, parcels = p, status_lists = s };

        v = vi.ToList();
    }
}
```



```
        return View(v);
}
```

5. Track Record Status and Details

It not possible show all details and complete status history in a single table. There option available on link **Track.** Click on **Track** and see details and booking history.



Track

Booking #: 8

Consignee #: Ali

Consignee Address #: ABC Road Lahore

Destination: Lahore

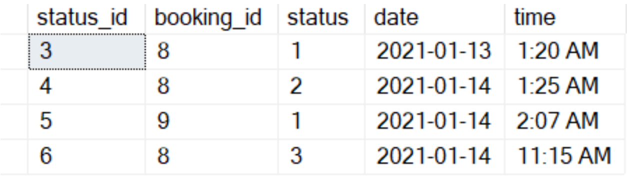
COD: 1200

Status	Date	Time
Dispatch	2021-01-14	1:25 AM
Book	2021-01-13	1:20 AM

© 2021 - FYP

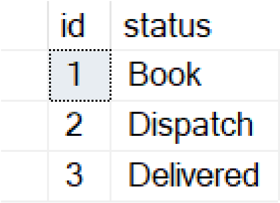
Here different joins are used and we show offline tracking in a single view. Below is the separate view for both booking_status and status_list

booking_status table:

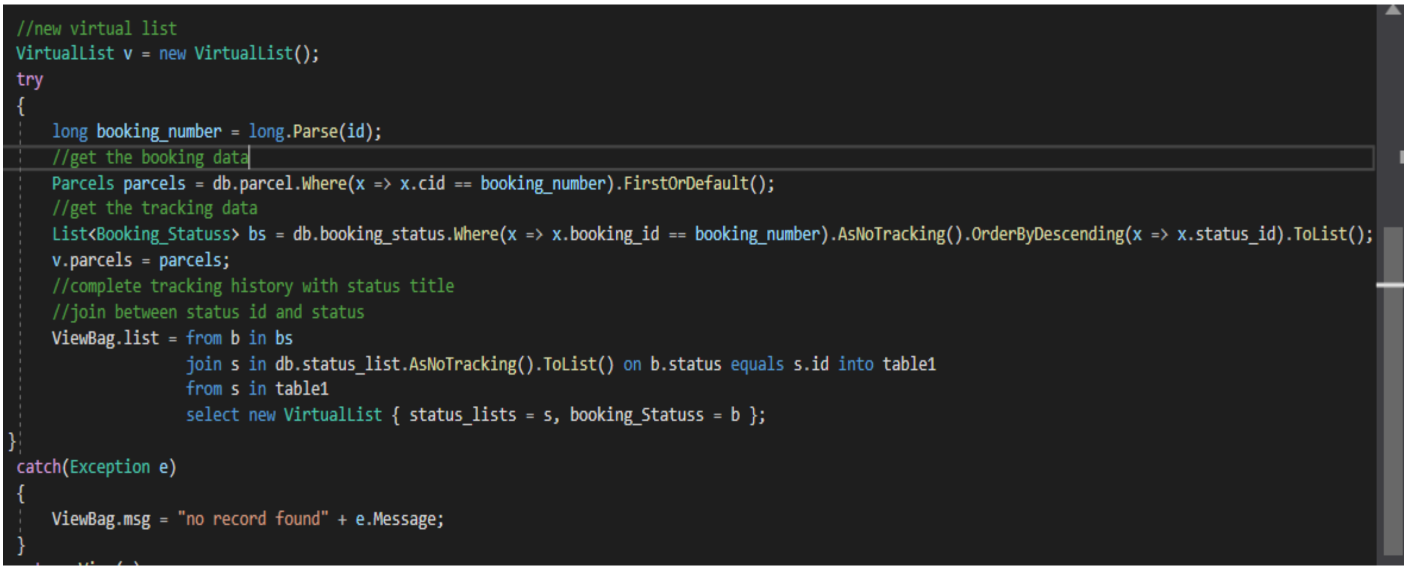


status_id	booking_id	status	date	time
3	8	1	2021-01-13	1:20 AM
4	8	2	2021-01-14	1:25 AM
5	9	1	2021-01-14	2:07 AM
6	8	3	2021-01-14	11:15 AM

status_list table:



id	status
1	Book
2	Dispatch
3	Delivered

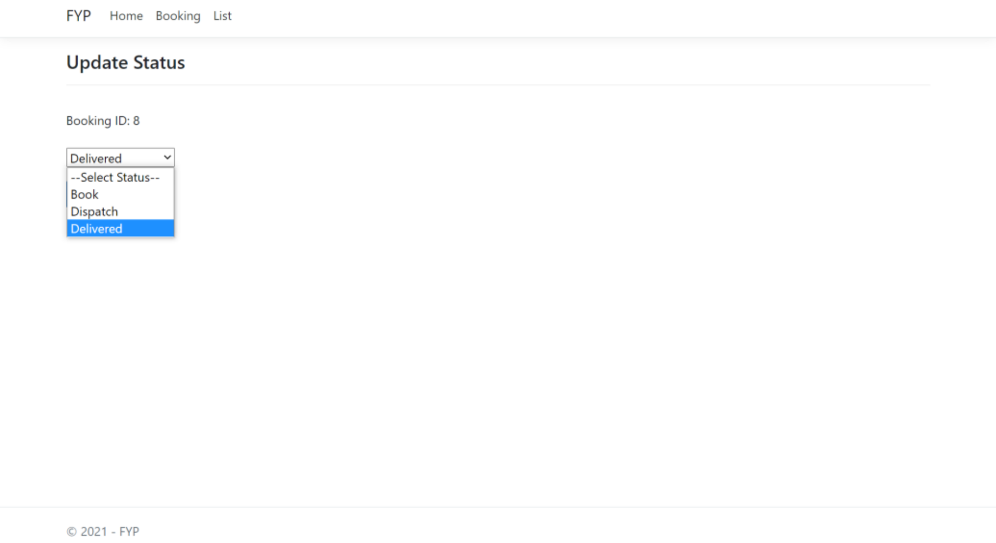


```
//new virtual list
VirtualList v = new VirtualList();
try
{
    long booking_number = long.Parse(id);
    //get the booking data
    Parcels parcels = db.parcel.Where(x => x.cid == booking_number).FirstOrDefault();
    //get the tracking data
    List<Booking_Statuss> bs = db.booking_status.Where(x => x.booking_id == booking_number).AsNoTracking().OrderByDescending(x => x.status_id).ToList();
    v.parcels = parcels;
    //complete tracking history with status title
    //join between status id and status
    ViewBag.list = from b in bs
                   join s in db.status_list.AsNoTracking().ToList() on b.status equals s.id into table1
                   from s in table1
                   select new VirtualList { status_lists = s, booking_Statuss = b };
}
catch(Exception e)
{
    ViewBag.msg = "no record found" + e.Message;
}
```

The above tracking shown in a single view by using joins. Following is the backend code for complex queries to apply web mining and knowledge discovery concept.

6. Update Status

To update status, Click on update status. To avoid manual binding of dropdown, same web mining and KDD concept, bind dropdown with database. Form for updating status.



Update Status

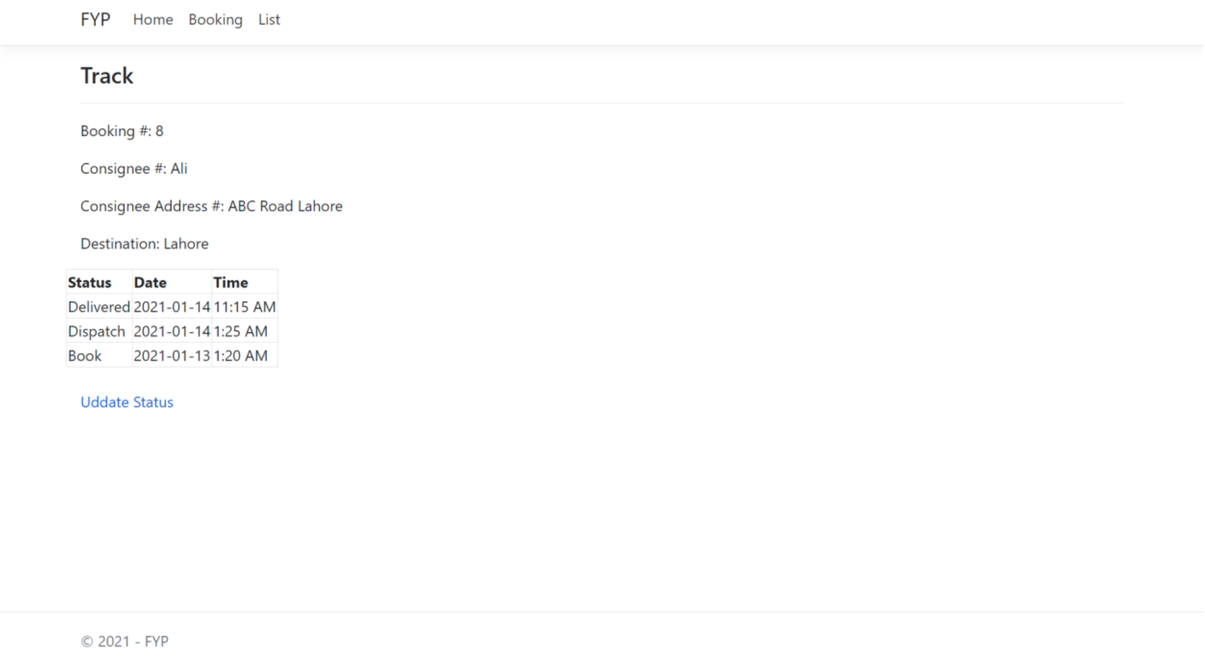
Booking ID: 8

Delivered

--Select Status--
Book
Dispatch
Delivered

© 2021 - FYP

After updating status, now see the offline tracking



Track

Booking #: 8

Consignee #: Ali

Consignee Address #: ABC Road Lahore

Destination: Lahore

Status	Date	Time
Delivered	2021-01-14	11:15 AM
Dispatch	2021-01-14	1:25 AM
Book	2021-01-13	1:20 AM

Uddate Status

© 2021 - FYP