**Decision Tree**

A decision tree is a diagram representation of **possible solutions** to a decision. It shows different outcomes from a set of decisions. The diagram is a widely used **decision-making tool** for analysis and planning.

The diagram starts with a box (or root), which branches off into several solutions. That’s way, it is called decision tree.

Decision trees are helpful for a variety of reasons. Not only they are easy-to-understand diagrams that support you ‘see’ your thoughts, but also because they provide a framework for estimating all possible alternatives.

In addition, decision trees help you manage the brainstorming process so you are able to consider the potential outcomes of a given choice.

**Example 1:** The Structure of Decision Tree

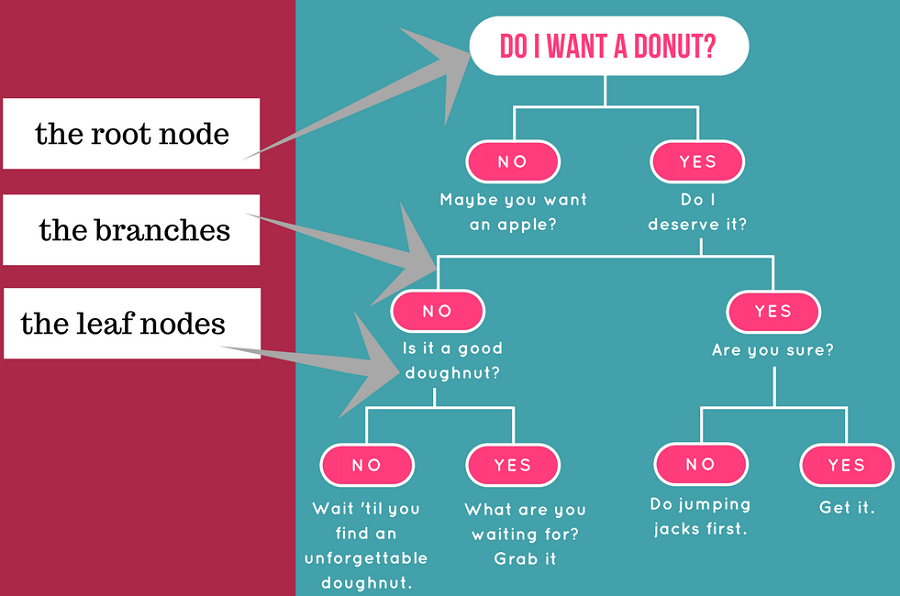
Let’s explain the decision tree structure with a simple example.

Each decision tree **has 3 key parts:**

* a root node
* leaf nodes, and
* branches.

No matter what type is the decision tree, it starts with a specific decision. This decision is depicted with a box – the root node.

Root and leaf nodes hold questions or some criteria you have to answer. Commonly, nodes appear as a squares or circles. Squares depict decisions, while circles represent uncertain outcomes.



Let's assume we want to play badminton on a particular day — say Saturday — how will you decide whether to play or not. Let's say you go out and check if it's hot or cold, check the speed of the wind and humidity, how the weather is, i.e. is it sunny, cloudy, or rainy. You take all these factors into account to decide if you want to play or not.

So, you calculate all these factors for the last ten days and form a lookup table like the one below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Weather** | **Temperature** | **Humidity** | **Wind** | **Play?** |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Cloudy | Hot | High | Weak | Yes |
| 3 | Sunny | Mild | Normal | Strong | Yes |
| 4 | Cloudy | Mild | High | Strong | Yes |
| 5 | Rainy | Mild | High | Strong | No |
| 6 | Rainy | Cool | Normal | Strong | No |
| 7 | Rainy | Mild | High | Weak | Yes |
| 8 | Sunny | Hot | High | Strong | No |
| 9 | Cloudy | Hot | Normal | Weak | Yes |
| 10 | Rainy | Mild | High | Strong | No |

Table 1. Obeservations of the last ten days.

Now, you may use this table to decide whether to play or not. But, what if the weather pattern on Saturday does not match with any of rows in the table? This may be a problem. A decision tree would be a great way to represent data like this because it takes into account all the possible paths that can lead to the final decision by following a tree-like structure.

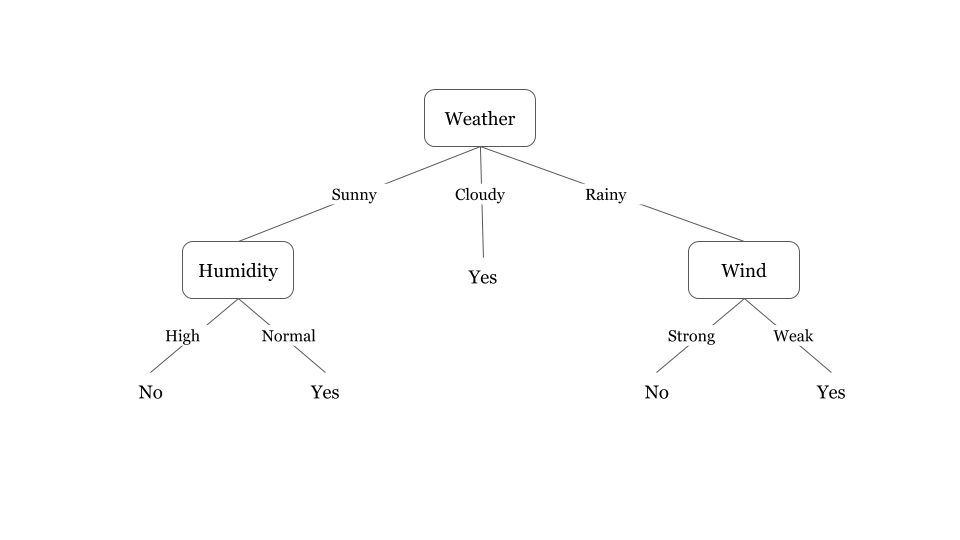
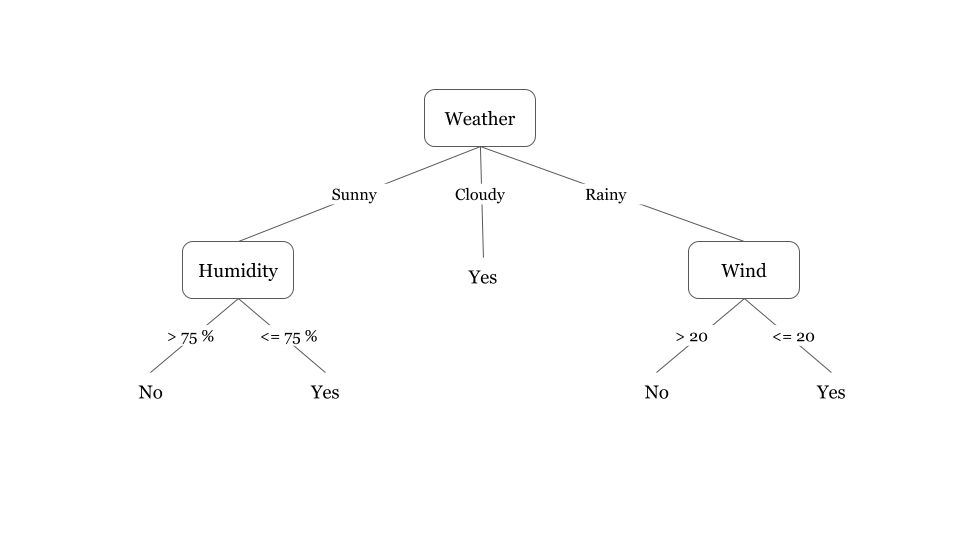


Fig 1. A decision tree for the concept Play Badminton

Fig 1. illustrates a learned decision tree. We can see that each node represents an attribute or feature and the branch from each node represents the outcome of that node. Finally, its the leaves of the tree where the final decision is made. If features are continuous, internal nodes can test the value of a feature against a threshold (see Fig. 2).

Fig 2. A decision tree for the concept Play Badminton (when attributes are continuous)

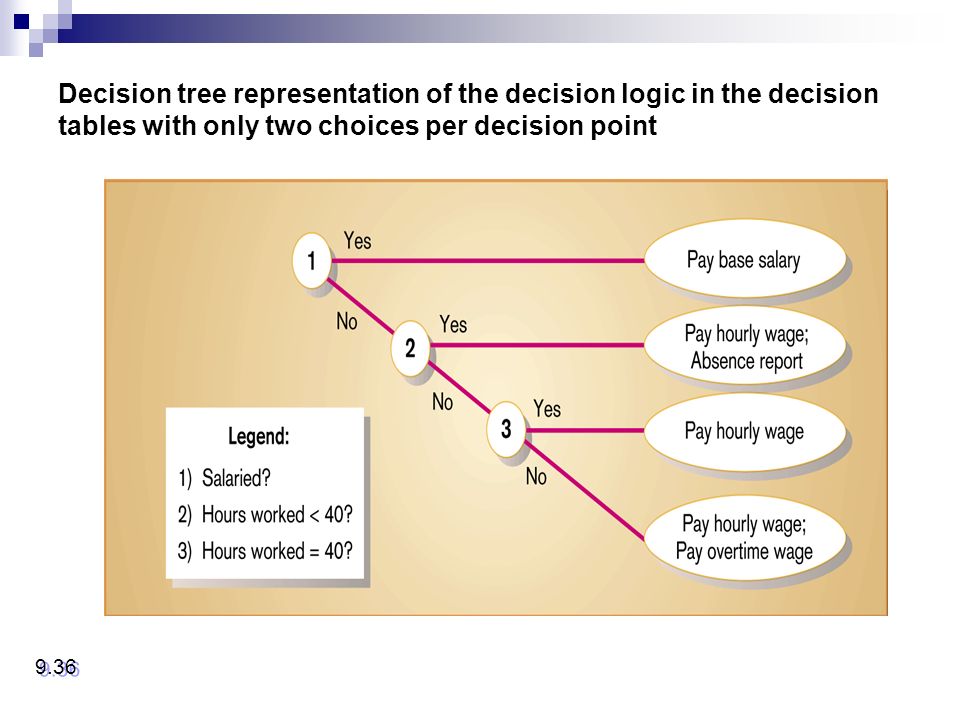
A general algorithm for a decision tree can be described as follows:

1. Pick the best attribute/feature. The best attribute is one which best splits or separates the data.
2. Ask the relevant question.
3. Follow the answer path.
4. Go to step 1 until you arrive to the answer.

The best split is one which separates two different labels into two sets.

**Questions**

1. Create a decision tree to represent the logic of payroll system described in the following narrative. There are two types of employees: salarized and hourly. All salarized employees get basic salary. Hourly wage is calculated for hourly worker. For hourly worker, if hours worked is less than 40 absence report is also produced and if it is greater than 40 overtime is also calculated.



1. Construct a decision tree and a decision table that represents a salesperson’s commission. The rules are as follows.

**a)      If fewer than 400 units are sold, than the salesperson’s commission is 2% of total sales.**

**b)      If between 400 and 499 units are sold, than the salesperson’s commission is 3% of total sales.**

**c)      If 500 or more units are sold, and the salesperson has been with the company more than one year, than the salesperson’s commission is 5% of total sales.**

**d)      If 500 or more units are sold, and the salesperson has been employed by the firm for one year or less, than his/her commission is 4% of total sales.**

Consider the following information on a decision process used within an organization to grant salesperson commissions:

1. If fewer than 400 units are sold, the salesperson’s commission is 1% of total sales.
2. If between 400 and 499 units are sold, the salesperson’s commission is 2% of total sales.
3. If 500 or more units are sold and the salesperson has been with the company for more than one year, the salesperson’s commission is 4% of total sales.
4. If 500 or more units are sold and the salesperson has been in the company for one year or less, his or her commission is 3% of total sales.

In addition to the above commission scheme, an additional commission of 0.5% of total sales is added if more than 80% of the sales have been collected by the salesperson. Draw an extended decision table representing the above information. Compress the table and convert it to a decision tree.

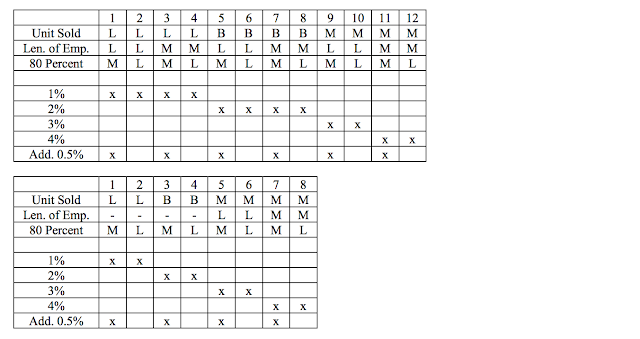
**Solution:**

For this particular use case, let us assume three decision variables:

1. Number of unit sold (Less then 400, Between 400 and 499, and More then 499),
2. Length of employment (Less than one, More than one year), and
3. Sales percentage (More than 80%, Less than 80%)

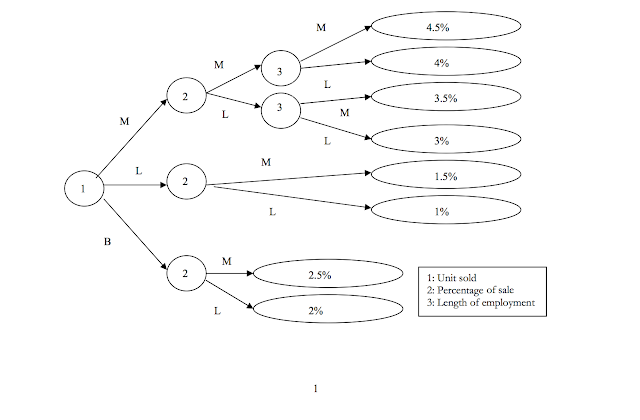
Based on these three decision variables and information provided in the question (points 1, 2, 3 and 4) we can construct decision table:

**Decision Table:**

[](https://3.bp.blogspot.com/-MrQ61R-WOUw/WFBmmbFPs6I/AAAAAAAAJA0/lv6TAvsDrtovmWRkm12HgXpFi5Kjw3cOACLcB/s1600/Screen+Shot+2016-12-13+at+22.18.41.png)

Based on above decision table, we can construct decision tree as follows:

**Decision Tree:**

[](https://1.bp.blogspot.com/-jmyPqfDAq-M/WFBm3sSBxPI/AAAAAAAAJA4/2wuKdpI9k7wY0TN5qzspSjtakSCgJnd3wCLcB/s1600/Screen+Shot+2016-12-13+at+22.22.58.png)

1. Develop a decision tree and a decision table for the following :

**         If the person is under three years of age, there is no admission fee.**

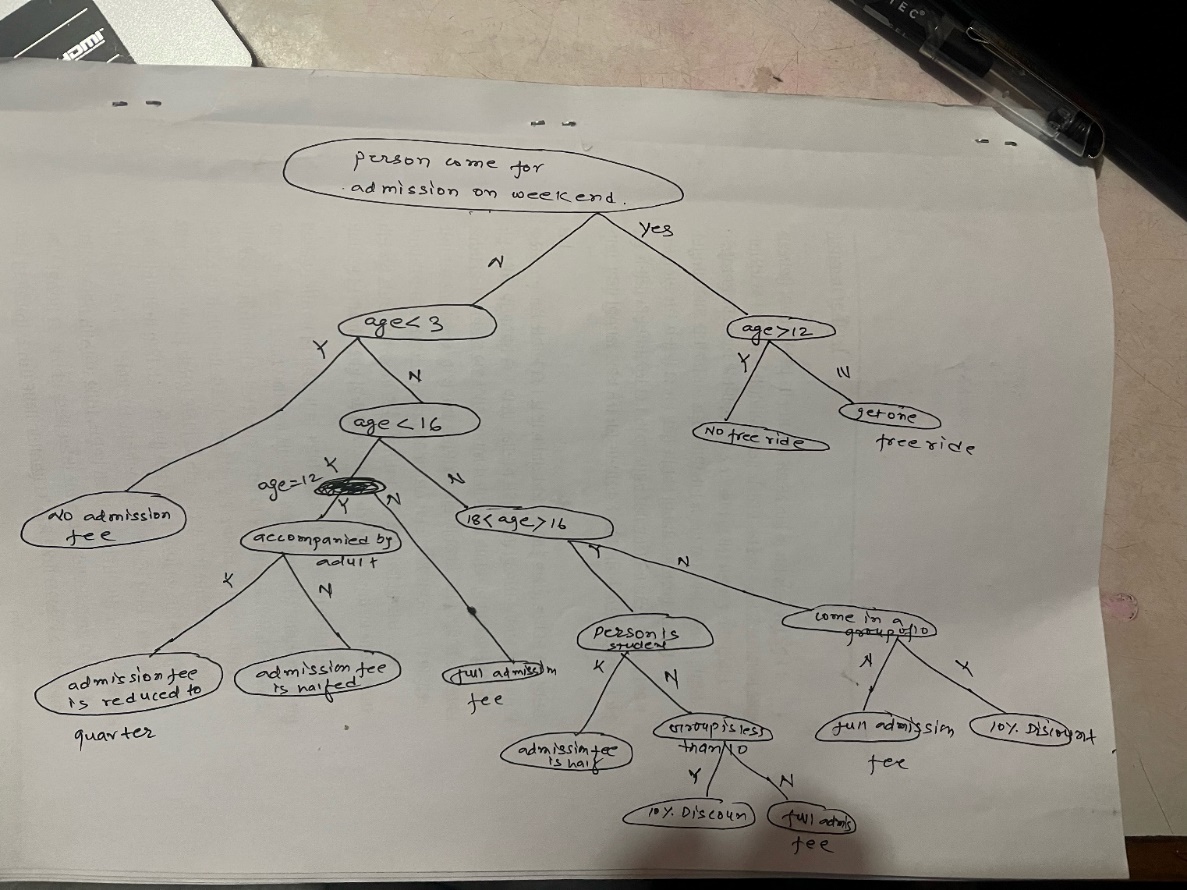
**         If a person is under 16, half the full admission is charged and this admission is reduced to a quarter of full admission if the person   is accompanied by an adult (the reduction applies only if the person is under 12).**

**         Between 16 and 18, half the full admission fee is charged if the person is a student; otherwise the full admission is charged.**

**         Over 18, the full admission fee is charged.**

**         A discount of 10 percent is allowed for a person over 16 if they are in a group of 10 or more.**

**         There are no student’s concessions during weekends. On weekdays under -12s get one free ride.**



# **Payback Period**

##### ****Example:****

Suppose ABC ltd is analyzing a project which requires an investment of $2,00,000 and it is expected to generate [cash flows](https://www.wallstreetmojo.com/cash-flow/) as follows

|  | **Year Annual cash inflows** |
| --- | --- |
| 1 | 80,000 |
| 2 | 60,000 |
| 3 | 60,000 |
| 4 | 20,000 |

In this cash payback period can be calculated as follows by calculating cumulative cashflows

|  |  |  |  |
| --- | --- | --- | --- |
| **Year** | **Annual cash inflows** | **Cumulative Annual cash inflows** | **Payback period** |
| 1 | 80,000 | 80,000 |  |
| 2 | 60,000 | 1,40,000(80,000+60,000) |  |
| 3 | 60,000 | 2,00,000(1,40,000+60,000) | **In this Year 3 we got initial investment of $ 2,00,000 so this is the pay back year** |
| 4 | 20,000 | 2,20,000(2,00,000+20,000) |  |

Suppose, in the above case, if the cash outlay is $2,05,000, then pa back period is

|  |  |  |  |
| --- | --- | --- | --- |
| **Year** | **Annual cash inflows** | **Cumulative Annual cash inflows** | **Payback period** |
| 1 | 80,000 | 80,000 |  |
| 2 | 60,000 | 1,40,000(80,000+60,000) |  |
| 3 | 60,000 | 2,00,000(1,40,000+60,000) |  |
| 4 | 20,000 | 2,20,000(2,00,000+20,000) | The payback period is between 3 and 4 years |

For up to three years, a sum of $2,00,000 is recovered, the balance amount of $ 5,000($2,05,000-$2,00,000) is recovered in a fraction of the year, which is as follows.

Forgetting $20,000 additional cash flows, the project is taking complete  12 months. So for getting additional of $ 5,000($2,05,000-$2,00,000) it will take (5,000/20,000) 1/4th Year. i.e., 3 months.

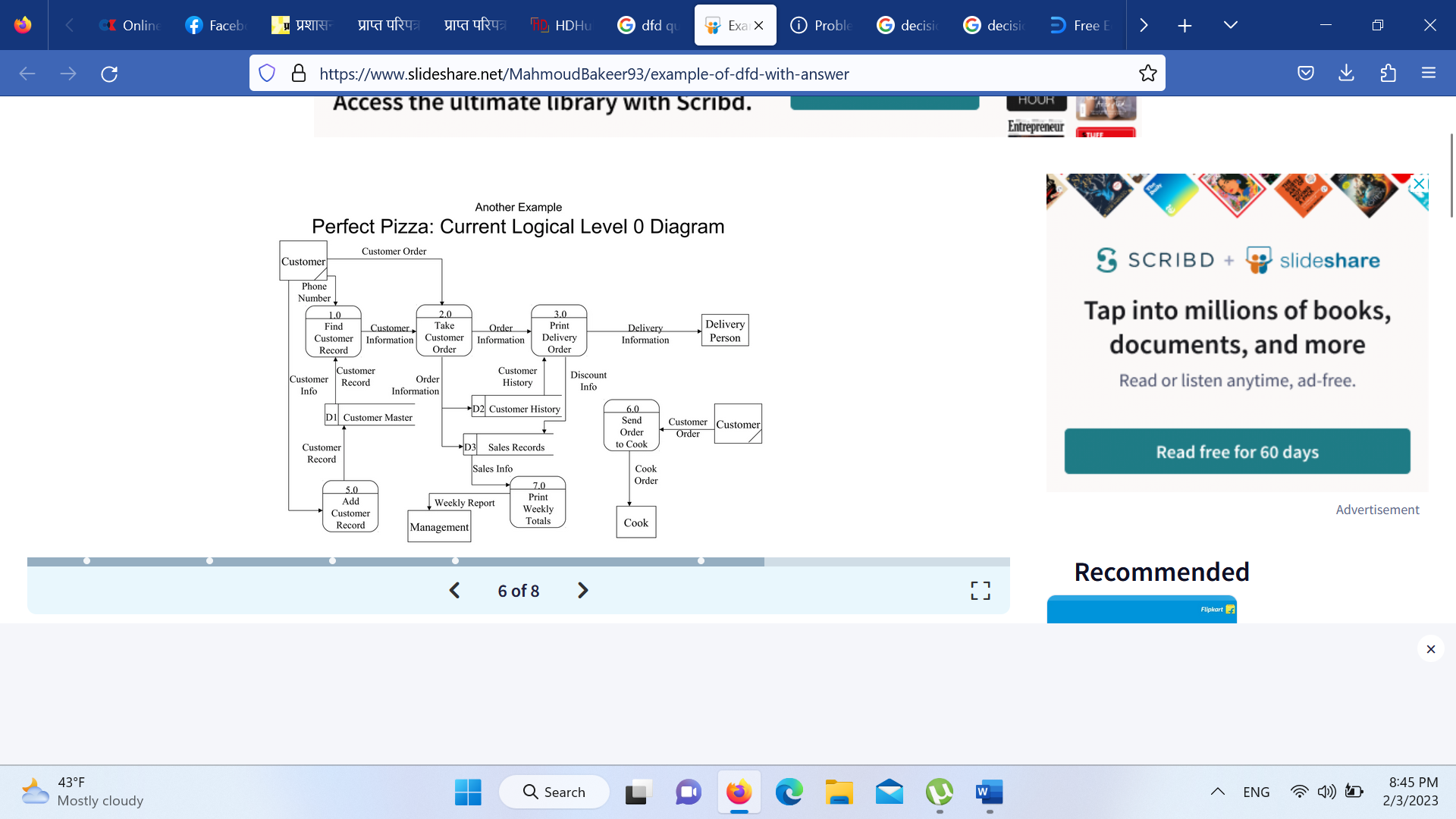
So, the project payback period is **3 years 3 months.**

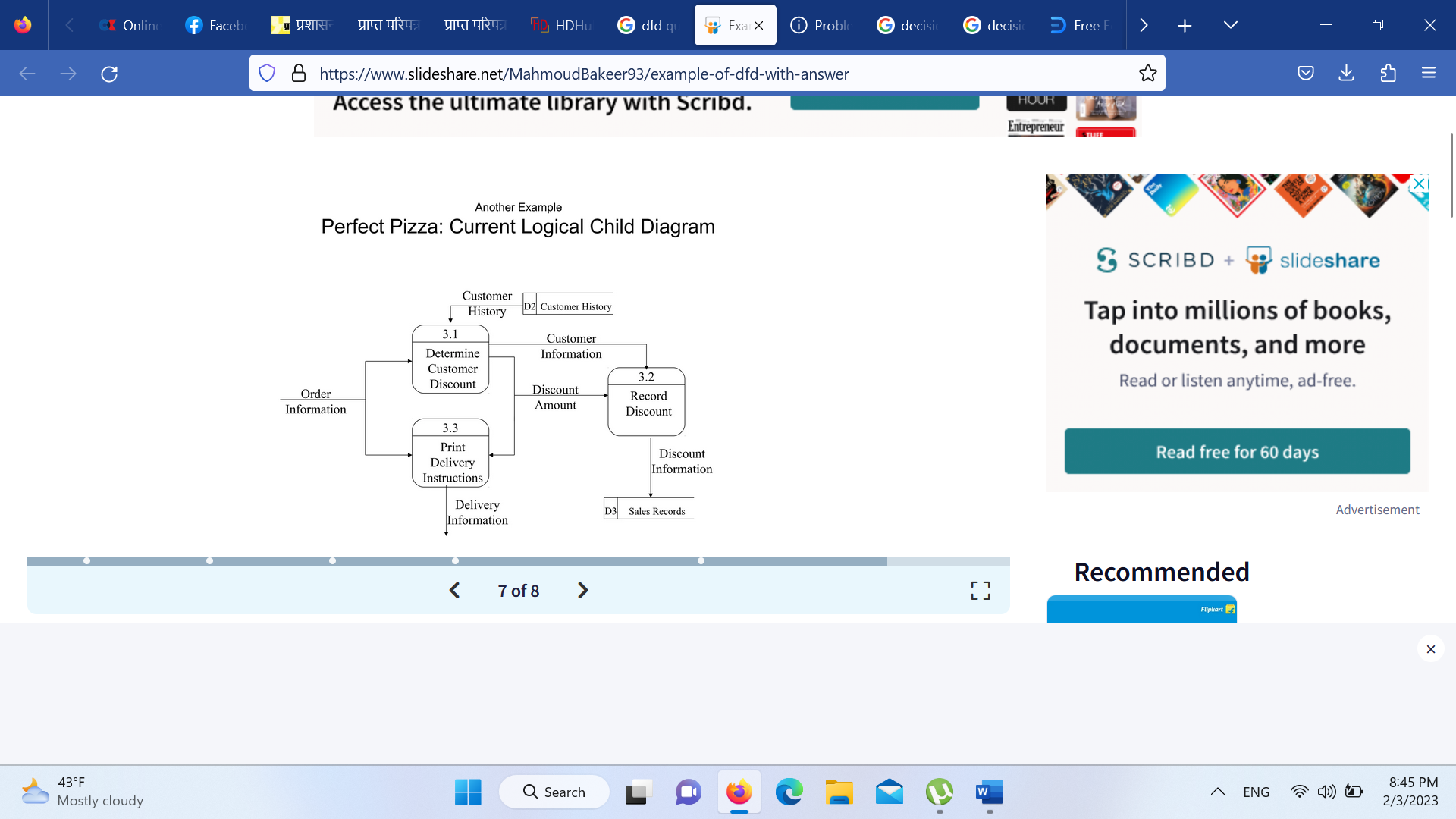
# **Old question**

1. A system costs Rs. 2,00,000 to install. The profit per year is Rs. 50,000. Assuming an interest rate is 6%, what is the payback period of the investment?
2. A system costs Rs. 1, 00, 000 to install and Rs. 8, 000 per month as recurring expenses. The benefit per year is Rs. 1, 50, 000. Assuming an interest rate is 12%, what is the payback period of the investment?
3. A system costs Rs. 1, 00, 000 to install and Rs. 8, 000 per month as recurring expenses. The benefit per year is Rs. 1, 50, 000. Assuming an interest rate is 14%, what is the payback period of the investment

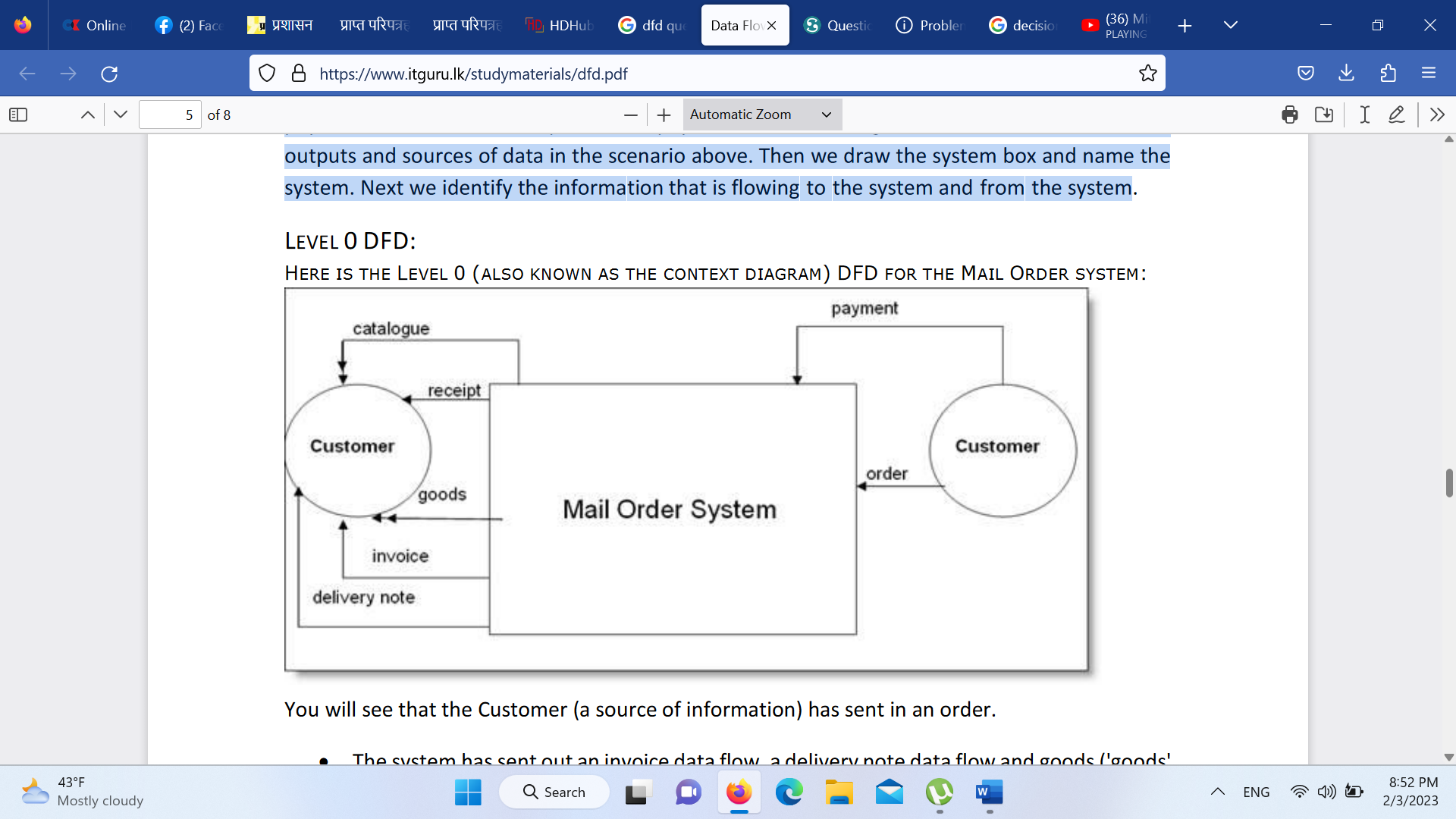
##### 

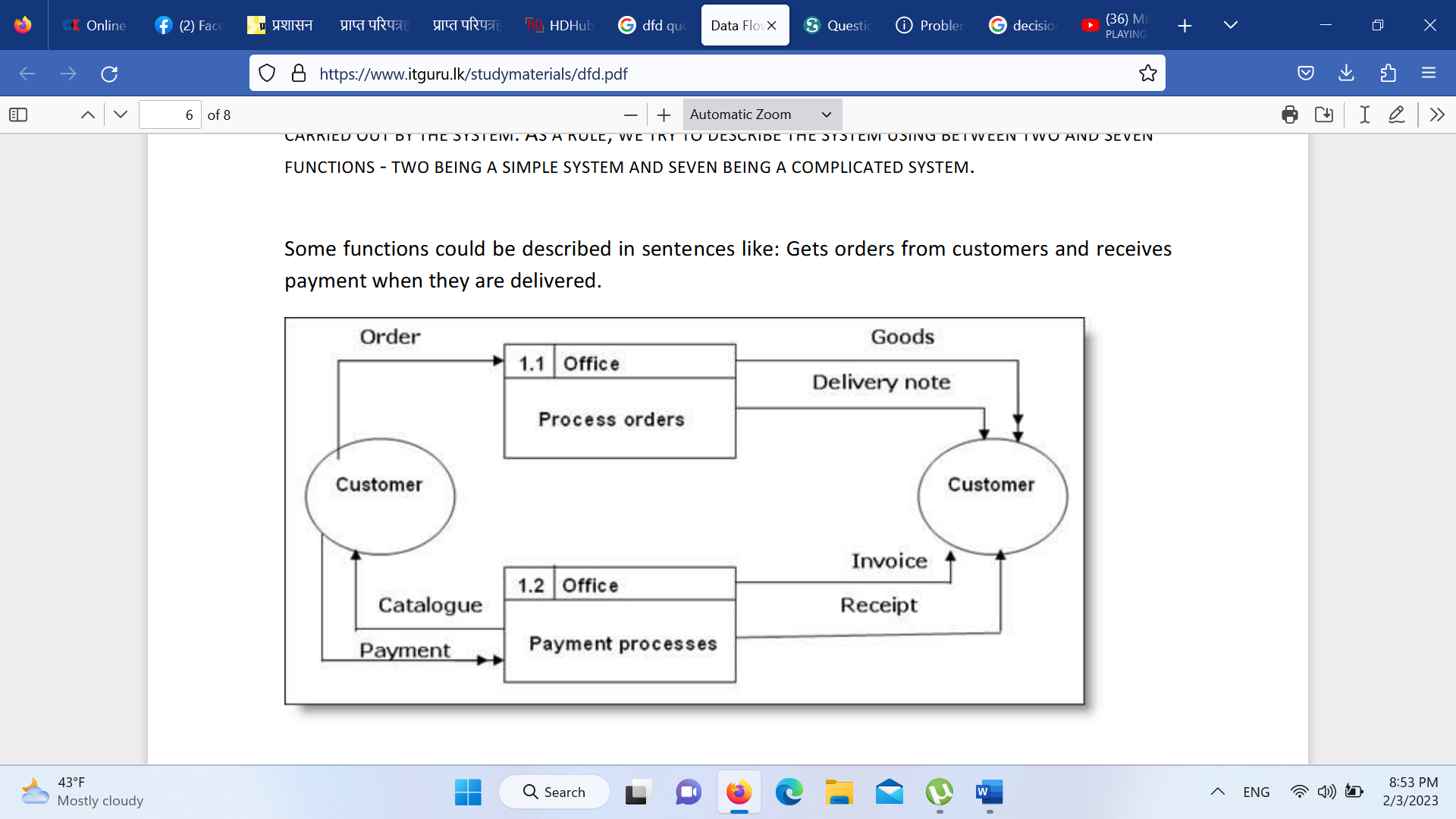
##### 





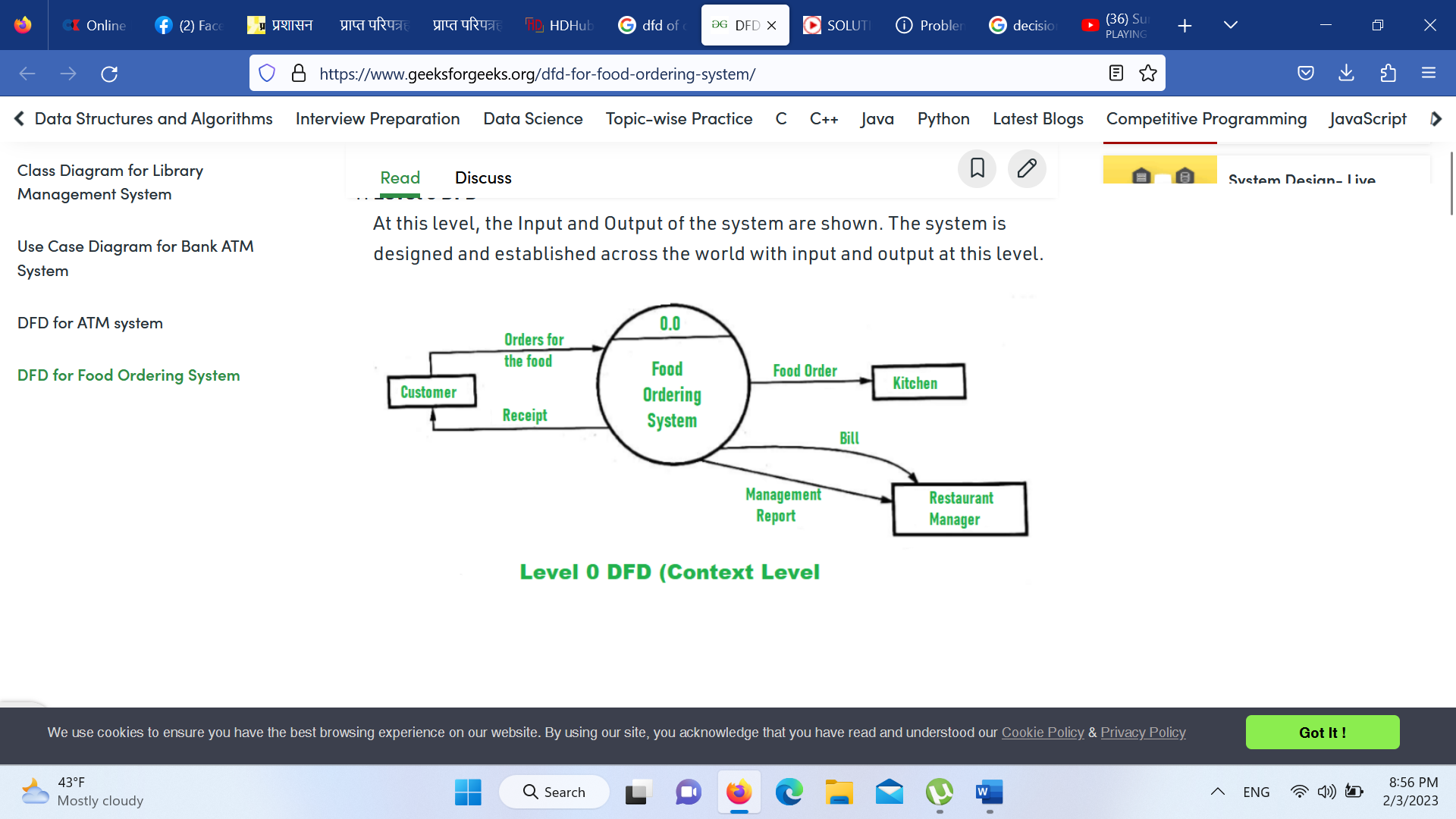
Suppose you are given the details of a small mail order catalogue system that allows people to  
shop from home. When a customer receives the catalogue and wants to buy something, they cantelephone, fax or email their order to the company. The company gets the order and sends the goods and an invoice. When the customer receives the goods with a delivery note, they send payment and receive a receipt for their payment. The first thing we must do is model the main outputs and sources of data in the scenario above. Then we draw the system box and name the system. Next we identify the information that is flowing to the system and from the system

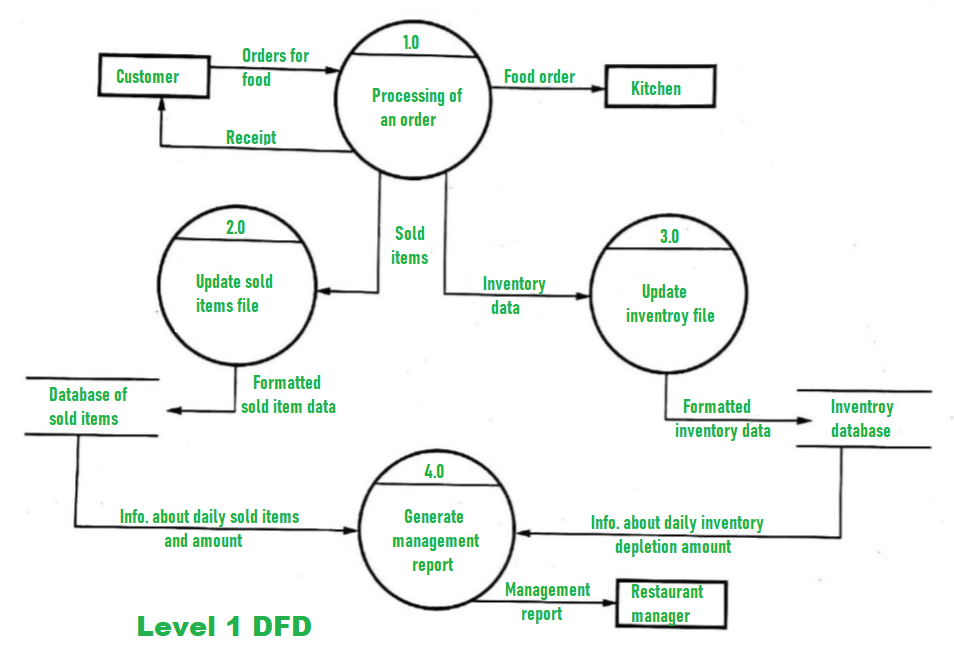


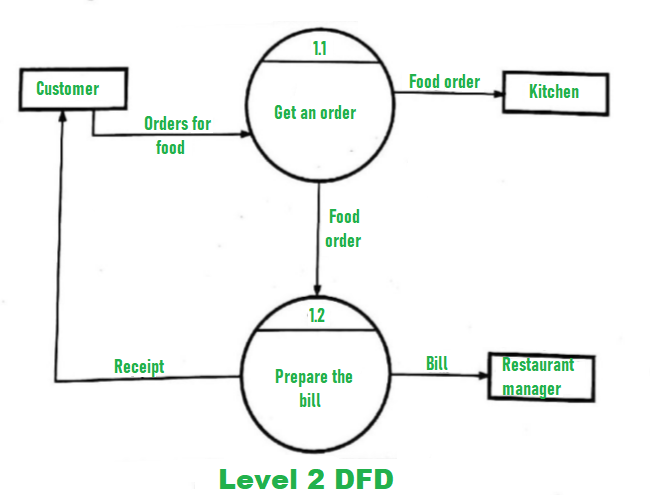




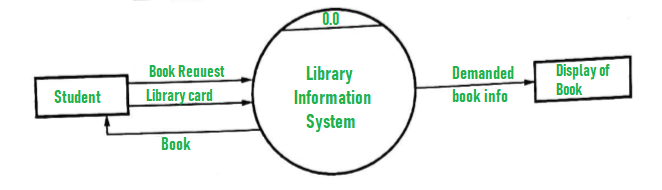
# **DFD for Food Ordering System**

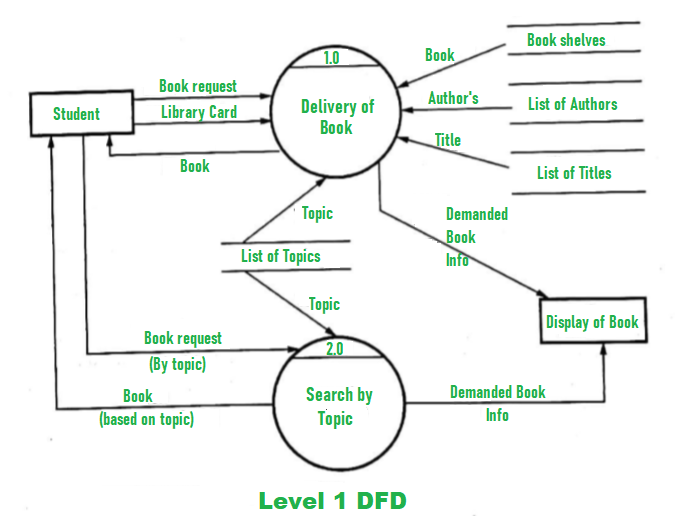


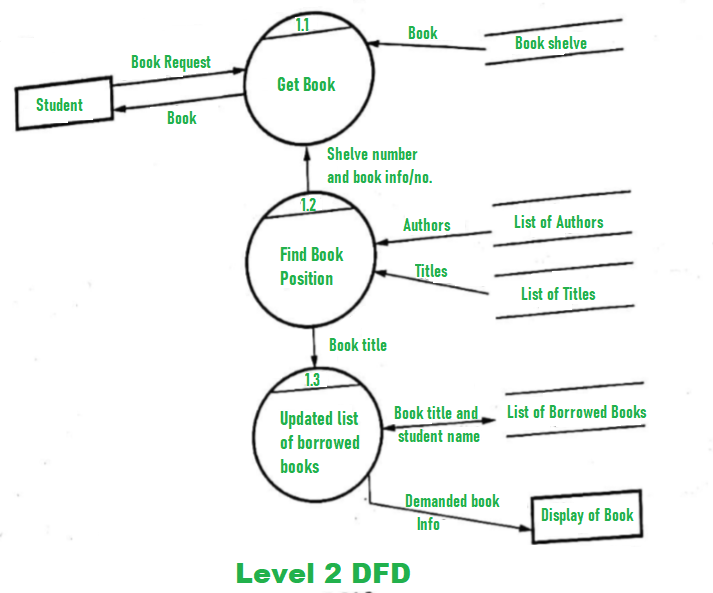




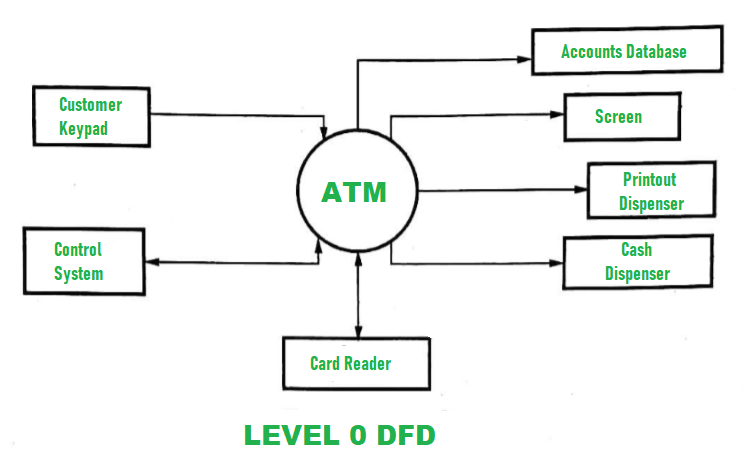
# DFD for Library Management System

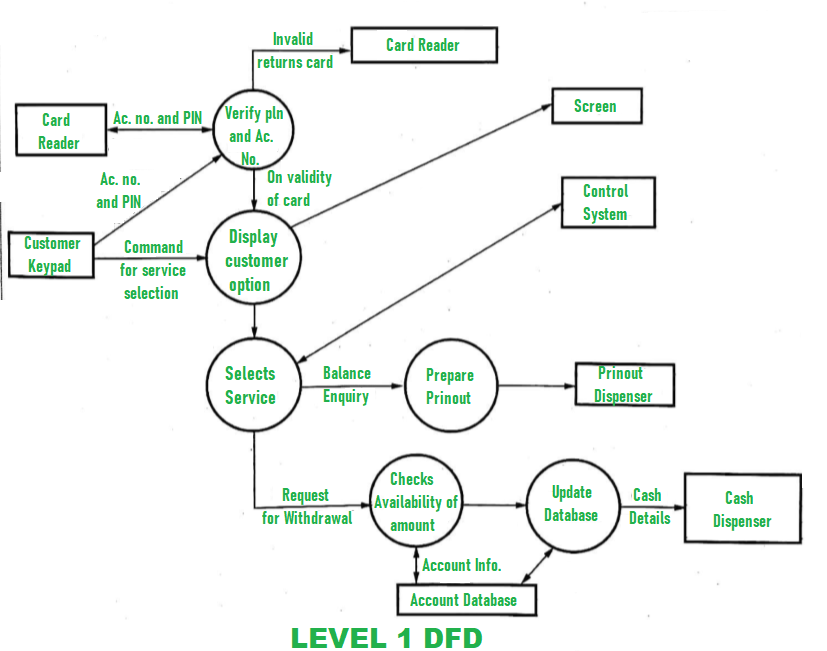




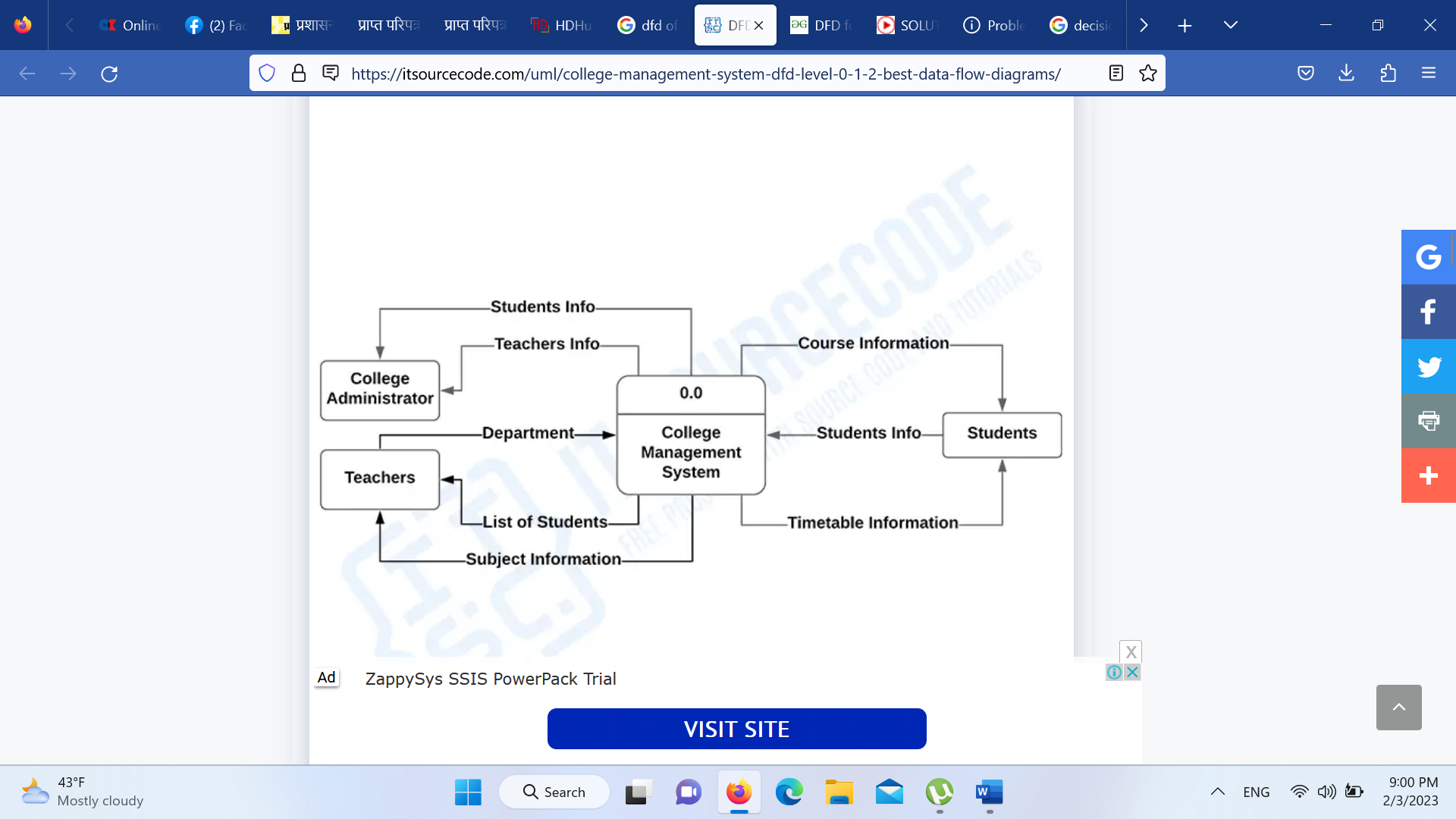


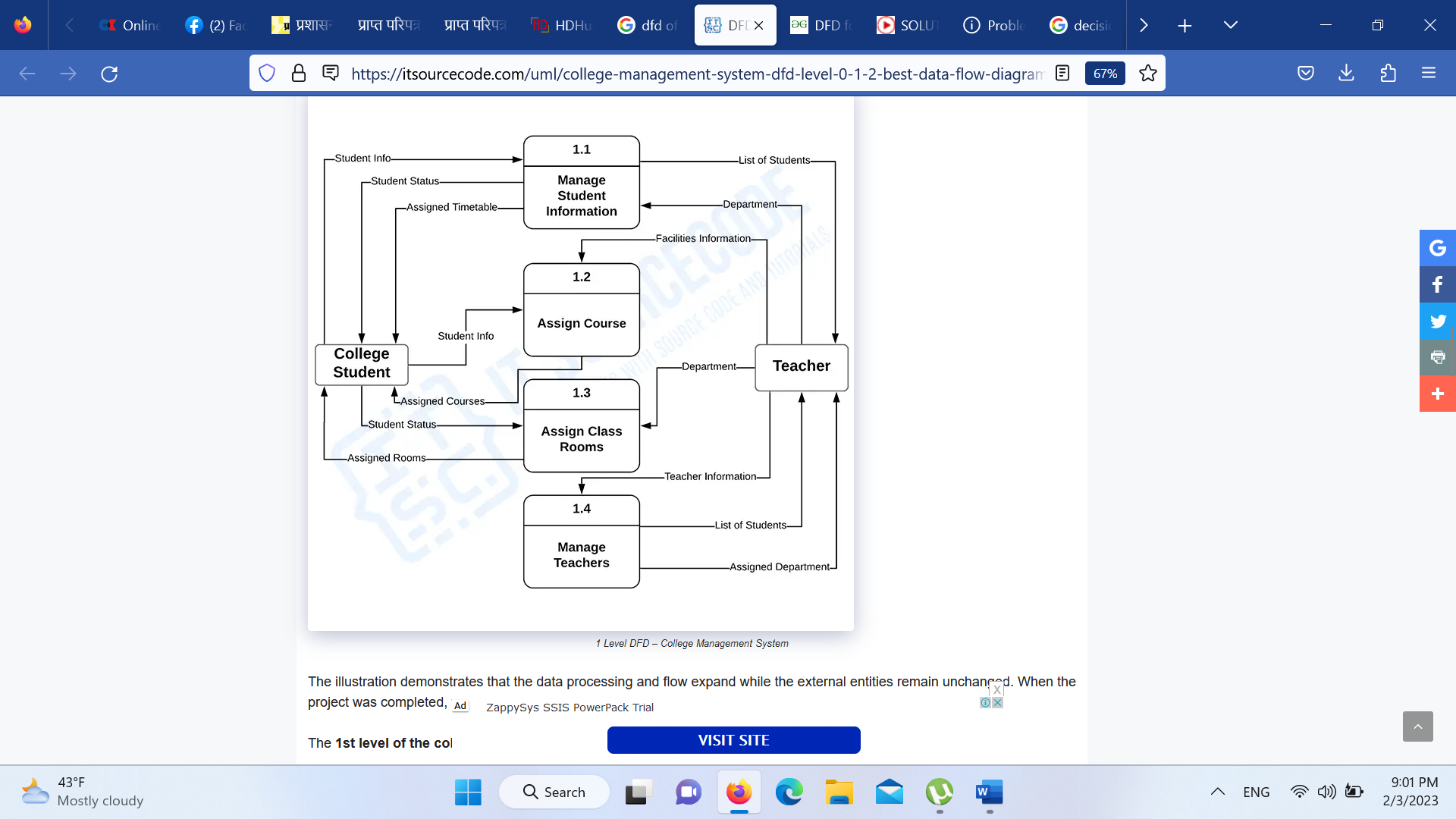
# **DFD for ATM system**





# **DFD for College Management System**







# **Entity relationship diagram**

1. **The Library Management**

The Library Management System database keeps track of readers with the following considerations –

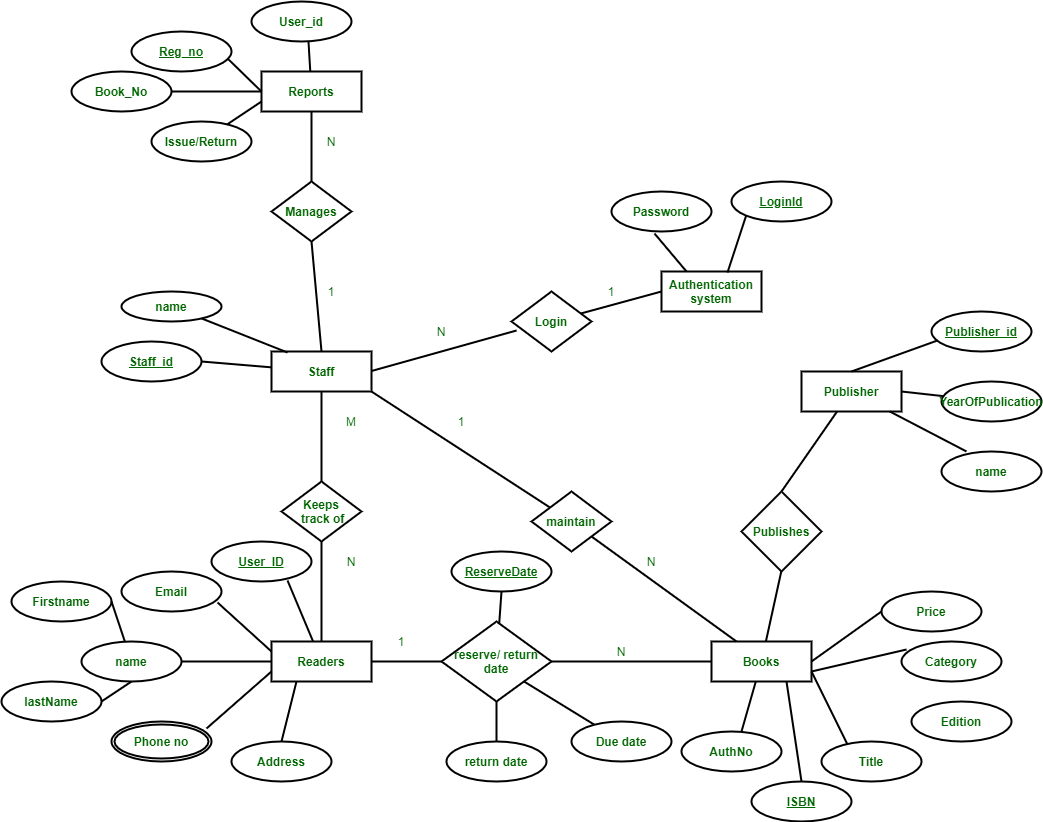
* The system keeps track of the staff with a single point authentication system comprising login Id and password.
* Staff maintains the book catalog with its ISBN, Book title, price(in INR), category(novel, general, story), edition, author Number and details.
* A publisher has publisher Id, Year when the book was published, and name of the book.
* Readers are registered with their user\_id, email, name (first name, last name), Phone no (multiple entries allowed), communication address. The staff keeps track of readers.
* Readers can return/reserve books that stamps with issue date and return date. If not returned within the prescribed time period, it may have a due date too.
* Staff also generate reports that has readers id, registration no of report, book no and return/issue info.

**Entities and their Attributes –**

* **Book Entity :** It has authno, isbn number, title, edition, category, price. ISBN is the Primary Key for Book Entity.
* **Reader Entity :** It has UserId, Email, address, phone no, name. Name is composite attribute of firstname and lastname. Phone no is multi valued attribute. UserId is the Primary Key for Readers entity.
* **Publisher Entity :** It has PublisherId, Year of publication, name. PublisherID is the Primary Key.
* **Authentication System Entity :** It has LoginId and password with LoginID as Primary Key.
* **Reports Entity :** It has UserId, Reg\_no, Book\_no, Issue/Return date. Reg\_no is the Primary Key of reports entity.
* **Staff Entity :** It has name and staff\_id with staff\_id as Primary Key.
* **Reserve/Return Relationship Set :** It has three attributes: Reserve date, Due date, Return date.

[**Relationships**](https://www.geeksforgeeks.org/attributes-to-relationships-in-er-model/) **between Entities –**

* A reader can reserve N books but one book can be reserved by only one reader. The relationship 1:N.
* A publisher can publish many books but a book is published by only one publisher. The relationship 1:N.
* Staff keeps track of readers. The relationship is M:N.
* Staff maintains multiple reports. The relationship 1:N.
* Staff maintains multiple Books. The relationship 1:N.
* Authentication system provides login to multiple staffs. The relation is 1:N.



1. **ER Diagram of a Company**

ER diagram of Company has the following description :

* Company has several departments.
* Each department may have several Location.
* Departments are identified by a name, D\_no, Location.
* A Manager control a particular department.
* Each department is associated with number of projects.
* Employees are identified by name, id, address, dob, dat e\_of\_joining.
* An employee works in only one department but can work on several project.
* We also keep track of number of hours worked by an employee on a single project.
* Each employee has dependent
* Dependent has D\_name, Gender and relationship.

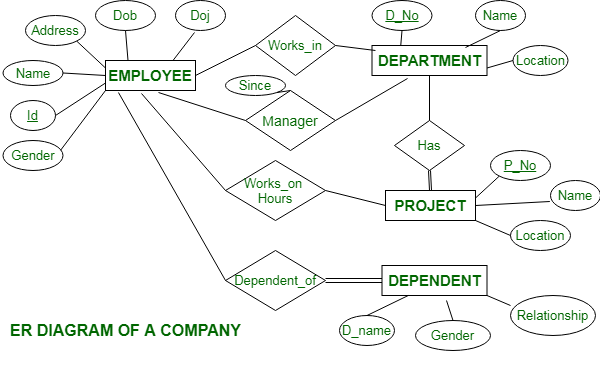
This Company ER diagram illustrates key information about Company, including entities such as employee, department, project and dependent. It allows to understand the relationships between entities.

**Entities** and their **Attributes** are

* **Employee Entity :** Attributes of Employee Entity are Name, Id, Address, Gender, Dob and Doj.  
  Id is Primary Key for Employee Entity.
* **Department Entity :** Attributes of Department Entity are D\_no, Name and Location.  
  D\_no is Primary Key for Department Entity.
* **Project Entity :** Attributes of Project Entity are P\_No, Name and Location.  
  P\_No is Primary Key for Project Entity.
* **Dependent Entity :** Attributes of Dependent Entity are D\_no, Gender and relationship.

**Relationships** are :

* **Employees works in Departments –**  
  Many employee works in one Department but one employee can not work in many departments.
* **Manager controls a Department –**  
  employee works under the manager of the Department and the manager records the date of joining of employee in the department.
* **Department has many Projects –**  
  One department has many projects but one project can not come under many departments.
* **Employee works on project –**  
  One employee works on several projects and the number of hours worked by the employee on a single project is recorded.
* **Employee has dependents –**  
  Each Employee has dependents. Each dependent is dependent of only one employee.



1. **ER diagram of Bank Management System**

ER diagram of Bank has the following description : 

* Bank have Customer.
* Banks are identified by a name, code, address of main office.
* Banks have branches.
* Branches are identified by a branch\_no., branch\_name, address.
* Customers are identified by name, cust-id, phone number, address.
* Customer can have one or more accounts.
* Accounts are identified by account\_no., acc\_type, balance.
* Customer can avail loans.
* Loans are identified by loan\_id, loan\_type and amount.
* Account and loans are related to bank’s branch.

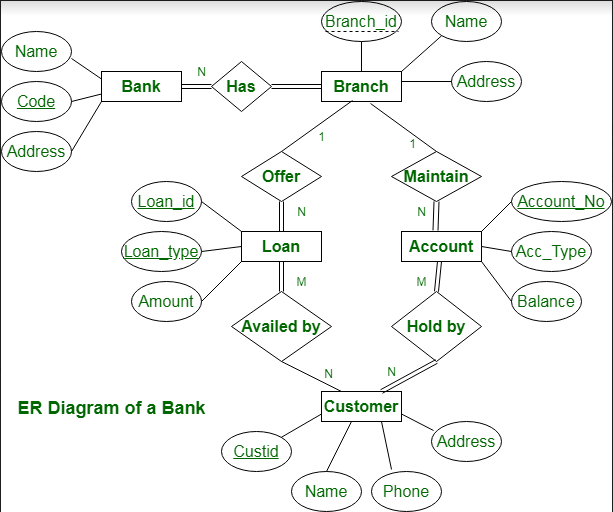
This bank ER diagram illustrates key information about bank, including entities such as branches, customers, accounts, and loans. It allows us to understand the relationships between entities.

**Entities** and their **Attributes** are :

* **Bank Entity :** Attributes of Bank Entity are Bank Name, Code and Address.   
  Code is Primary Key for Bank Entity.
* **Customer Entity :** Attributes of Customer Entity are Customer\_id, Name, Phone Number and Address.   
  Customer\_id is Primary Key for Customer Entity.
* **Branch Entity :** Attributes of Branch Entity are Branch\_id, Name and Address.   
  Branch\_id is Primary Key for Branch Entity.
* **Account Entity :** Attributes of Account Entity are Account\_number, Account\_Type and Balance.   
  Account\_number is Primary Key for Account Entity.
* **Loan Entity :** Attributes of Loan Entity are Loan\_id, Loan\_Type and Amount.   
  Loan\_id is Primary Key for Loan Entity.

[Relationships](https://www.geeksforgeeks.org/attributes-to-relationships-in-er-model/) are : 

* **Bank has Branches => 1 : N**  
  One Bank can have many Branches but one Branch can not belong to many Banks, so the relationship between Bank and Branch is one to many relationship.
* **Branch maintain Accounts => 1 : N**   
  One Branch can have many Accounts but one Account can not belong to many Branches, so the relationship between Branch and Account is one to many relationship.
* **Branch offer Loans => 1 : N**   
  One Branch can have many Loans but one Loan can not belong to many Branches, so the relationship between Branch and Loan is one to many relationship.
* **Account held by Customers => M : N**   
  One Customer can have more than one Accounts and also One Account can be held by one or more Customers, so the relationship between Account and Customers is many to many relationship.
* **Loan availed by Customer => M : N**   
  (Assume loan can be jointly held by many Customers).   
  One Customer can have more than one Loans and also One Loan can be availed by one or more Customers, so the relationship between Loan and Customers is many to many relationship.



1. **ER Diagram for Online Shopping System**

