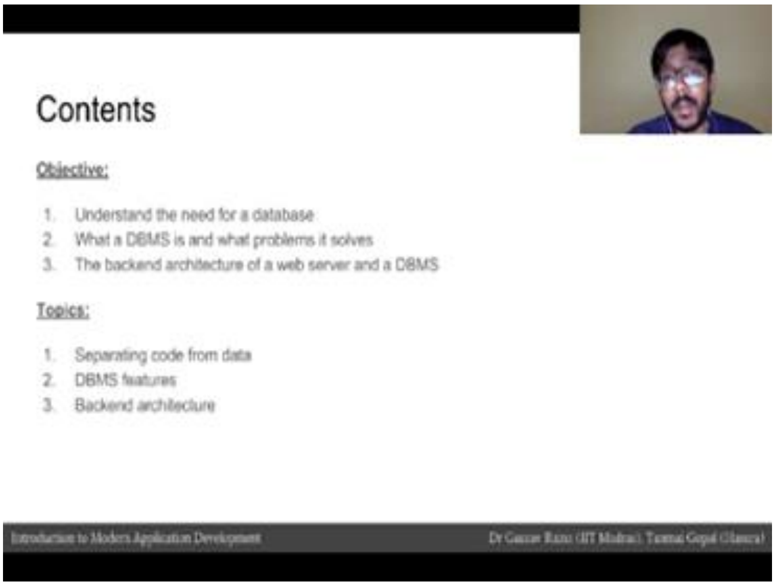
Introduction to Modern Application Development

Prof. Tanmai Gopal
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Module - 6
Lecture - 13
Introduction to Databases

Hi all. Welcome to Module-6 will be taking about Databases and getting a basic introduction to what databases are.

(Refer Slide Time: 00:06)



Contents

Objective:

1. Understand the need for a database
2. What a DBMS is and what problems it solves
3. The backend architecture of a web server and a DBMS

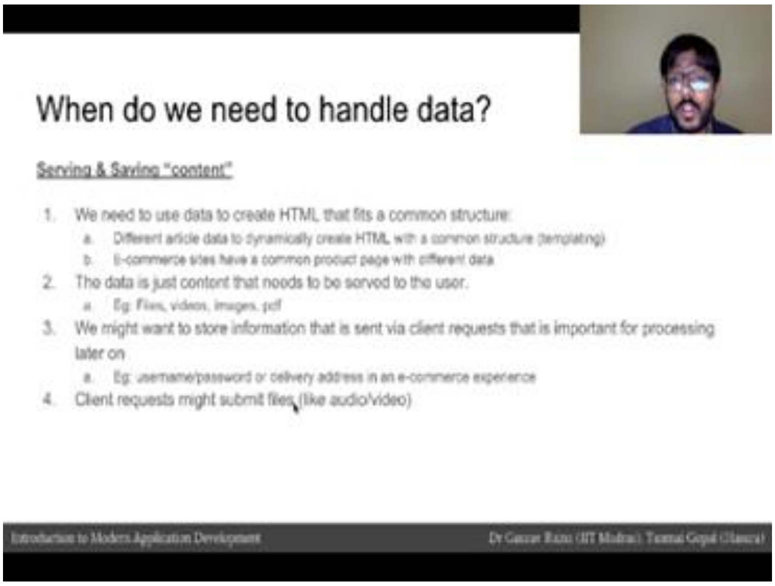
Topics:

1. Separating code from data
2. DBMS features
3. Backend architecture

Introduction to Modern Application Development Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

Our main objectives in this module is to understand the need for a data base and why of webapp even need data base once we understand that we look at popular DBMS system and what problems DBMS or a data base management system is suppose to solve. Then we look at what we architecture looks like if we have a web server and a DBMS.

(Refer Slide Time: 00:27)



When do we need to handle data?

Serving & Saving "content"

1. We need to use data to create HTML that fits a common structure:
   a. Different article data to dynamically create HTML with a common structure (templating)
   b. E-commerce sites have a common product page with different data
2. The data is just content that needs to be served to the user.
   a. Eg: Files, videos, images, pdf
3. We might want to store information that is sent via client requests that is important for processing later on
   a. Eg: username/password or delivery address in an e-commerce experience
4. Client requests might submit files (like audio/video)

Introduction to Modern Application Development

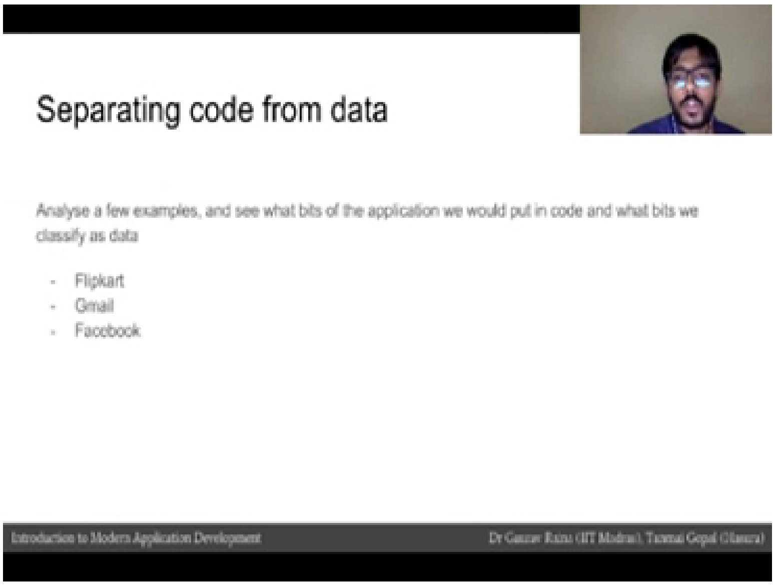
Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, to understand a data base, first thing to understand is when we need to handle data specifically to understand data bases the first very important to understand the use cases when we need to actually handle data, and what we mean by a data of context of application. Broadly and non-technically, whenever we serve or save content is typically when we need to handle data.

Let us look at few examples whenever you see pages that have a common structure, but different content on the page for example, product pages on e commerce site or the article page that we created last (Refer time: 01:00) week. Another example is, if you go to Facebook or flipkart.com or YouTube and you see a lot of contents that is serve to user video content a PDF content or images another example would be when you using an E-commerce site and receiving the delivery address or when you longing into a sight and saving username and password.

But in other place were user generated contents were needs to be handle explicitly user might also generate content of different types for example, image user audio files and this also needs to handle explicitly it is important understand that we need to separate code and data.

(Refer Slide Time: 01:33)



Separating code from data

Analyse a few examples, and see what bits of the application we would put in code and what bits we classify as data

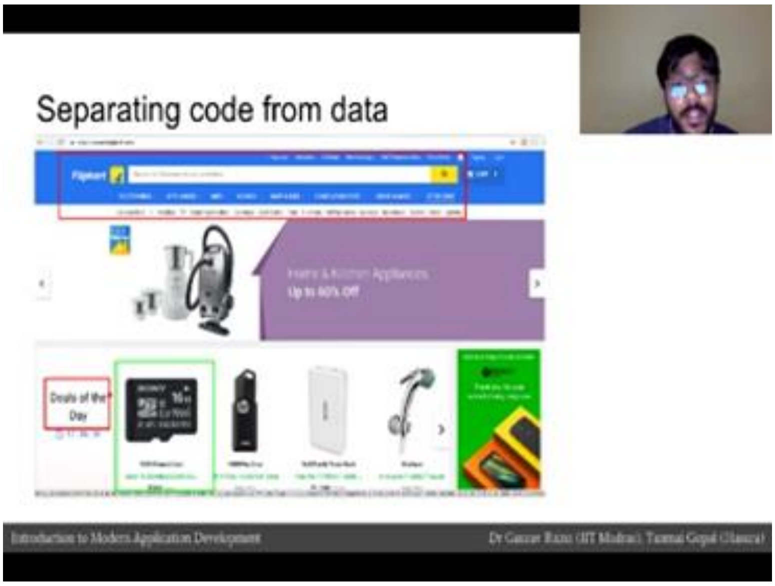
- Flipkart
- Gmail
- Facebook

Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

We need to separate the part of our application that sought of runs the application is the logic of the application and the part of the application that handles the data, and let us you could few examples to make this separation of code and data more clear.

(Refer Slide Time: 01:53)



Separating code from data

Introduction to Modern Application Development

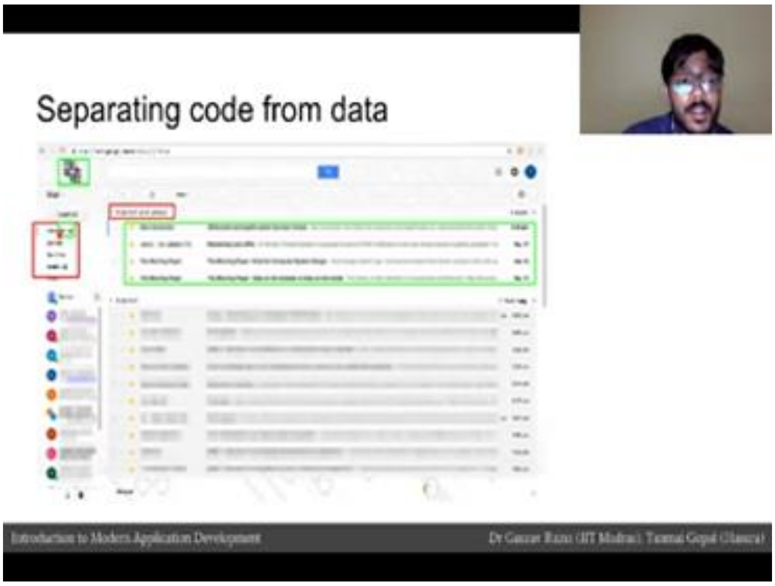
Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, let us let us look at a Flipkart and if you look at the Flipkart home page which is an

Ecommerce site. This topic of the flipkart site which is (Refer time: 02:00) a bit at red rectangular box. This part is what I would say is generated using the code bit of the flipkart were right, because these kinds of this are sought of hard coded or necessarily a part of the flipkart website. But if you look at this portion which is in the green box this part is a definitely a data because this is the part of the webapp that will change very frequently, and that has a very different a very characteristics in the way it is managed or in the way it is used on the webapp from this part of the content.

For example, if you just look at the ban are called deal of the day, this portion of the webapp that display a section that shows the heading called deals of the day, is a part of the code the content of what this banner contains is what i would say part of the data of the website.

(Refer Slide Time: 02:51)



Separating code from data

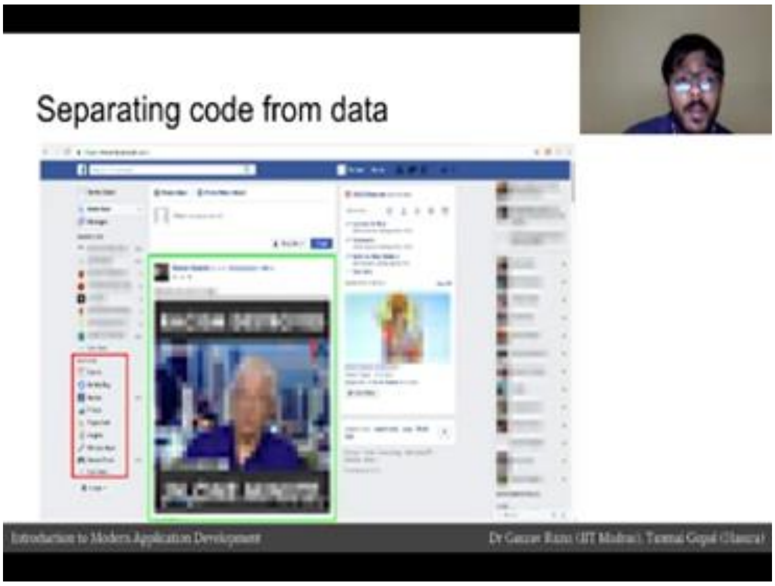
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras) Tanmai Gopal (Hasura)

If you look at Gmail and this is my screen shot of my inbox the bits of the app are responsible for generating this menu structure on the side that (Refer time: 03:00) are responsible for structuring this inbox into 3 sections. See, important; unread important and unread these sections are apart the code of the webapp and the email content, which is content at changes quite frequently that depends on other people sending the emails.

This is a part of the data of the g mail webapp in in this portion. If we look at the word inbox or start or sent mail these are the code portion of the webapp and this is the data potion of the webapp.

(Refer Slide Time: 03:33)



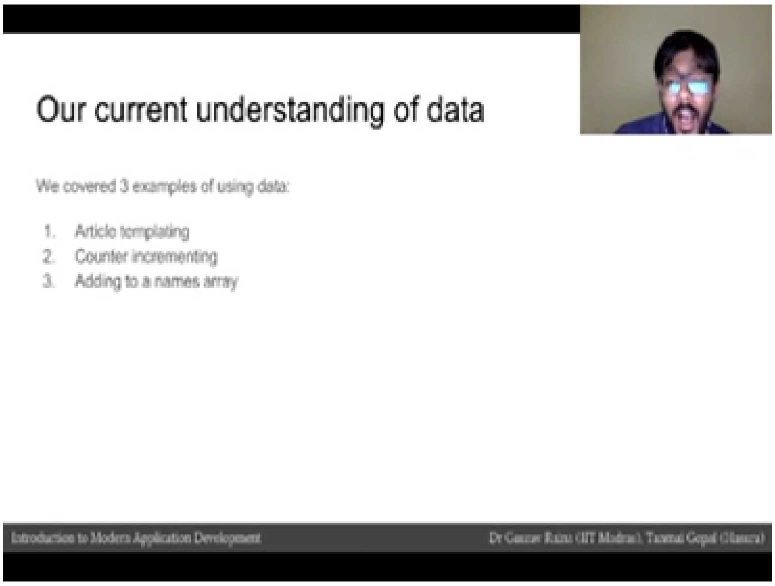
Separating code from data

Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

If the last examples, if we have look at facebook almost everything is on facebook is the data portion of the webapp apart from the structure from the overall side a. Let us look at the section here which is very definite and which does not change, but bits inside the green box which is a post this is definitely the data portion of the application.

(Refer Slide Time: 03:54)



Our current understanding of data

We covered 3 examples of using data:

1. Articlo templating
2. Counter incrementing
3. Adding to a names array

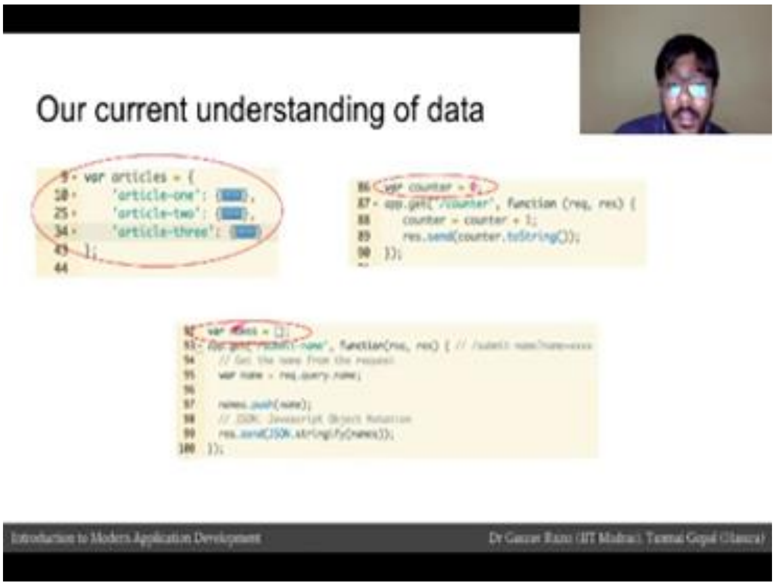
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, what is a current understanding of how we handle data and let us build upon the examples that we use (Refer time: 04:00) last week we templated article contents. So, we had 3 articles and we were displaying 3 articles in the same way with the same structure on article pages. We look at the examples of implementing a counter and that increases the counter if you can view click a p i and the a p i that handles an a p i that takes in a input in a name and outputs a list of names that it mean submitted, so for to the web.

These will the 3 places in your applications were, we are handling the data and let see how we are handle them so far.

(Refer Slide Time: 04:32)



Our current understanding of data

```
var articles = {
    'article-one': {...},
    'article-two': {...},
    'article-three': {...}
};
```

```
var counter = 0;
app.get('/counter', function (req, res) {
    counter = counter + 1;
    res.send(counter.toString());
});
```

```
var names = [];
app.get('/submit-name', function(req, res) { // /submit-name?name=xxxx
  // Get the name from the request
  var name = req.query.name;

  names.push(name);
  // JSON: Javascript Object Notation
  res.send(JSON.stringify(names));
});
```

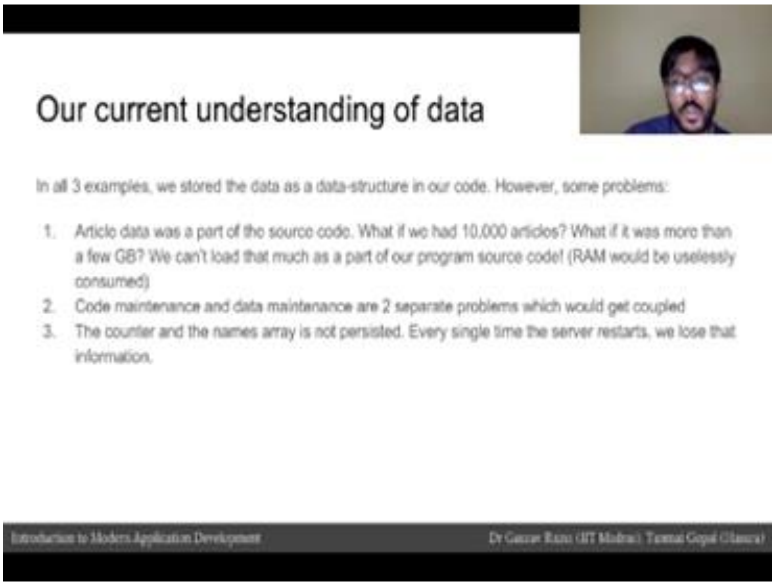
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

This is what our source code will look like for the webapp. We had an article object, this is a java script data structure and this data structure contains the article content. So, this is where a data was stored; if we look at the counter example, were the initial value of counter is zero and then every time the slash counter your always called the counter was the incremented we were handling our data which is just one integer as a variable inside a source code right.

And similarly when we had the (Refer time: 05:00) names end point where if you content the slash submit name and end point and a name was given to the URL end point it was been stored in an array. So, the data portion was the application was a kind of inside a variable called names which is m array.

(Refer Slide Time: 05:18)



Our current understanding of data

In all 3 examples, we stored the data as a data-structure in our code. However, some problems:

1. Article data was a part of the source code. What if we had 10,000 articles? What if it was more than a few GB? We can't load that much as a part of our program source code! (RAM would be uselessly consumed)
2. Code maintenance and data maintenance are 2 separate problems which would get coupled
3. The counter and the names array is not persisted. Every single time the server restarts, we lose that information.

Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

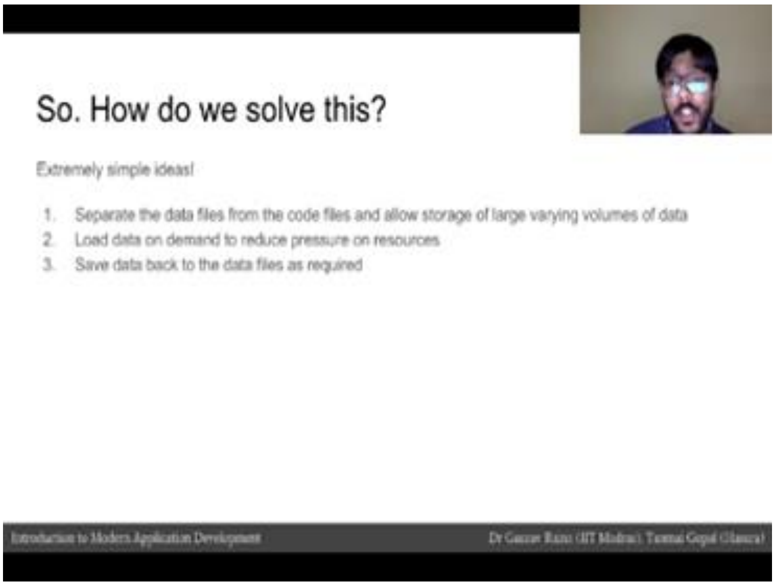
This was for very very simple example, but there are fairly big problems with doing thing this way. For example, the article data which was part of source code what if we had ten thousands articles what if we had few gb or a few 100 gb of articles would we store that inside this data structure, inside a source code; would be have a few million lines a inside a source code, which just handle the data in fact apart from the fact that this does not seem natural. We cannot even have that much either part of a program source code because very time the program is noted that much amount of a RAM would be consumed load the program and if the data was a few (Refer time: 06:00) 100 gb then it is not even a possible for a program source code to load.

The other kind of problem that one can have is that even if one does not have a few 100 articles for any webapp, the way you maintained the code and the way you maintained the data are two very separate activities. Just to take a very simple example developers maintain the source code, but people who maintain the data for example, a block side might not be a developers, they might be a people who manage a content and the life cycle in the tools that we use to manipulate code and manipulate the data at this case, could be very different.

The third problem that we have in our way doing things, in a very way of doing things that we use so far is that, the counter and the names array which is the data structure that modified as the app is used, this data was not processed that mean every time the web server restarted that information was lost. So, every time we reach the web servers are count has started from zero. Every time we restarted (Refer time: 07:00) a web server our names are restarted of an empty array.

So, if we ever have a restart or if a server crashes or we take it down maintenance of we update code, we would lose all the information that we gotten so far. So, these are the problems this is a problems in doing things by storing data as a part of the source code. So, how is this problem seems to be a very simple, right.

(Refer Slide Time: 07:22)



So. How do we solve this?

Extremely simple ideas!

1. Separate the data files from the code files and allow storage of large varying volumes of data
2. Load data on demand to reduce pressure on resources
3. Save data back to the data files as required

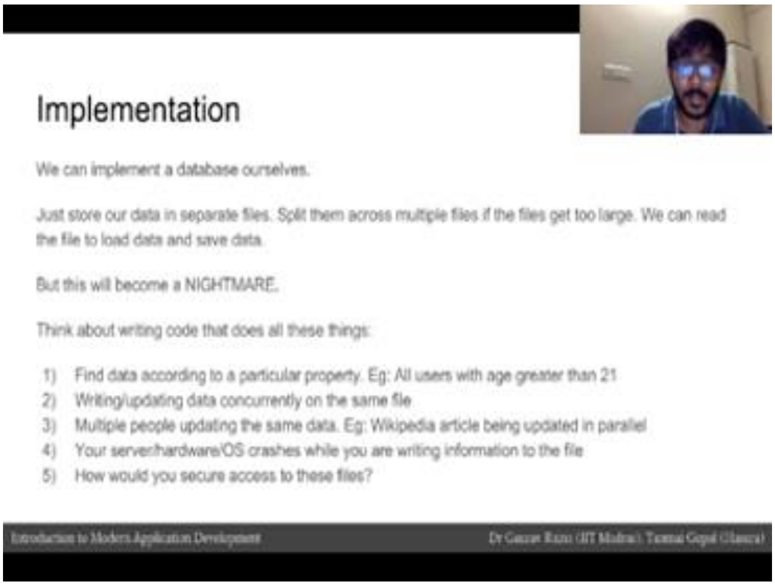
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

What would be the most things to do would be to separate the data files from the code files. And this would allow us to store a very large and sought of variable amounts of data because we just have separate files to store data.

Whenever we need the data we just loaded on demand so that we do not have to vary about the volume of a data we can just load the sub set data there is required from those file and whenever we need to save data. For example, in a counted example or a name the example we would just save the data back to these files; this on the surface of the sounds very simple and this in fact data driven applications work.

(Refer Slide Time: 07:56)



Implementation

We can implement a database ourselves.

Just store our data in separate files. Split them across multiple files if the files get too large. We can read the file to load data and save data.

But this will become a NIGHTMARE.

Think about writing code that does all these things:

1) Find data according to a particular property. Eg: All users with age greater than 21
2) Writing/updating data concurrently on the same file
3) Multiple people updating the same data. Eg: Wikipedia article being updated in parallel
4) Your server/hardware/OS crashes while you are writing information to the file
5) How would you secure access to these files?

Introduction to Modern Application Development   Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, even the making the application and using a data base sounds very simple we cannot (Refer time: 08:00) actually do that very easily for example, if we thing that we can just store a data in a separate files and split them across a multiple just the files get to large and then the read from the files a or ave. To the file we need to just doing these things will soon become an absolute nightmare a think just thing about these kinds of things.

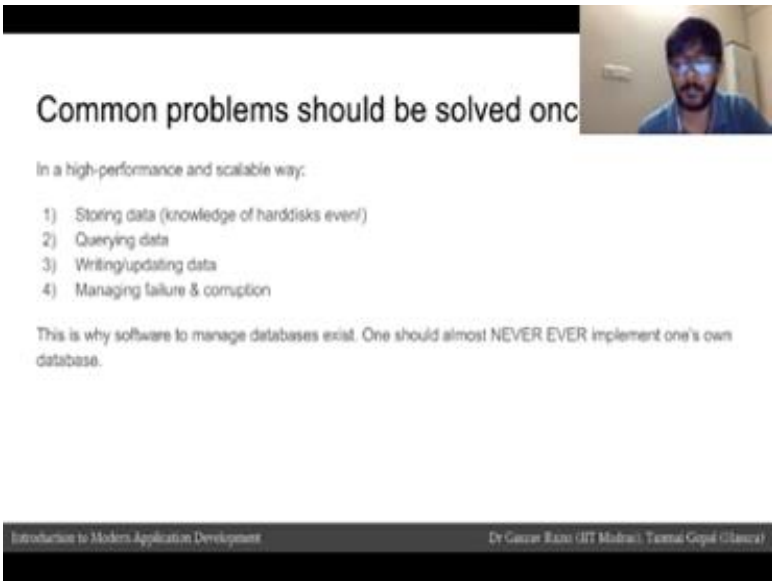
Just think about these kind of things that you have to implement not only will have to read them a file, but you also have to find that exact subset of data from the file which matches the quarry for example, you might want to find the all the users in your database. So, having age greater than 21 right and if you try to do that you will have to check cross all your files and by that particular quarry which you can imagine in a non task.

Another task that would quite hard is writing a updating the data congruently on the same file for example, if multiple people are writing to the file and see writing with the file is taking a long time or 3 or 4 people are writing to the file at the same time. Then you would have to handle the file writing very carefully to ensure that multiple people do not end up saving (Refer time: 09:00) in consistence version that the file.

This problem is even worse if multiple people are trying update the same data imagine a Wikipedia article being updated in parallel by several users' right, how would you know which the correct version of the article to save is and which one is incorrect and these are just problems then things are going well. What happens when things go wrong? How would you ensure that your data is does not get become corrupt when your server or your hardware or even your operating system crashes? How will you insure that you can recover from your a data files that you have? How you secure access to these files to ensure that people who are not offer rise to access your data base, do not access your database.

For example in larger application like Gmail not all developers of Gmail will be able to access the database of Gmail, which contains all the images all the uses on Gmail. So, how would you secure access to database and the worst part is we would have to solve these problems over and over again for each app that we built right; these are common problems. And of course, these are been common problem for a while (Refer time: 10:00) database management solution that exist which kind solve all these problems for us we maintain a database.

(Refer Slide Time: 10:06)



Common problems should be solved onc

In a high-performance and scalable way:

1) Storing data (knowledge of harddisks even!)
2) Querying data
3) Writing/updating data
4) Managing failure & corruption

This is why software to manage databases exist. One should almost NEVER EVER implement one's own database.

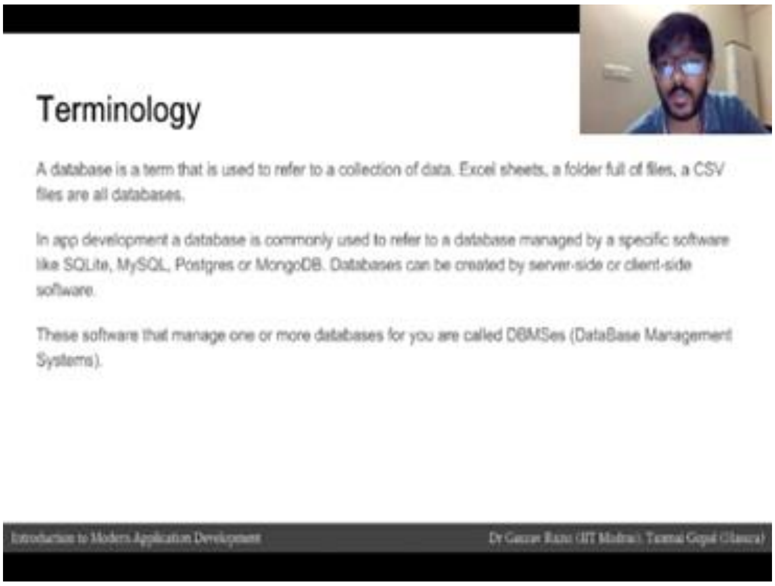
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, in a very high performance and in a scale able way this is what a database, management systems do for us they know, how to store the data and the best way they even exploit the knowledge of the hardest to ensure that the way that they manage the data on the disk, which has efficient and hard to corrupted is possible.

They provide easy ways and efficient ways of quarrying of data of writing and updating the data and of managing failure and data corruption and recovery methods of how do we cover data in case things goes wrong. We are almost in a position today were we should never ever implement our own database. We should always use a database a management solution of software whether exist open source of priority and use them in our application.

(Refer Slide Time: 10:50)



Terminology

A database is a term that is used to refer to a collection of data. Excel sheets, a folder full of files, a CSV files are all databases.

In app development a database is commonly used to refer to a database managed by a specific software like SQLite, MySQL, Postgres or MongoDB. Databases can be created by server-side or client-side software.

These software that manage one or more databases for you are called DBMSes (DataBase Management Systems).

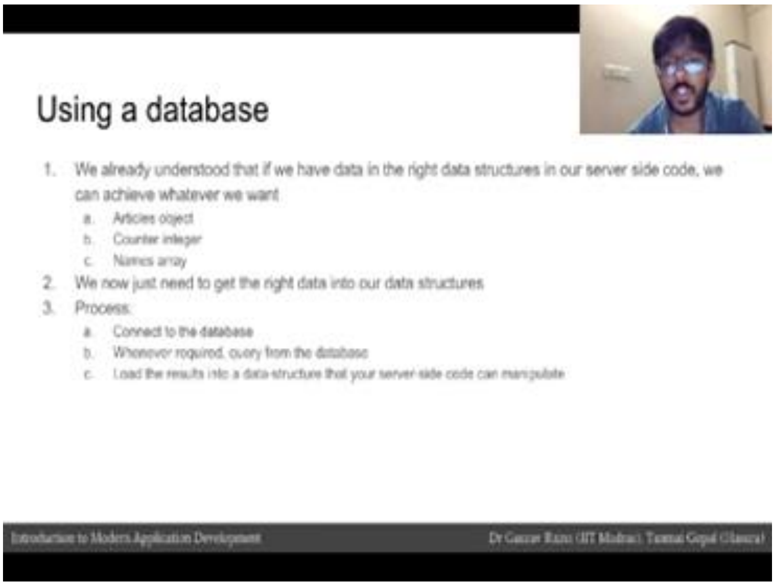
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, just to make the terminology clear when use the word the database; a database actually refers to a collection of a data. So, an excel sheet of folder full of files c s e these are (Refer time: 11:00) all databases right. But each of these databases are managed differently for example, excel sheets and a folder full of files are managed manually, they are not managed by software automatically and so software that manage these collection of data and manage these databases are called DBMS. So, database management system.

So, in web applications or in application development frequently when we use the word database we actually mean DBMS, but will be using them - inter changeably. Some common database management systems that you would have heard of are my SQL Postgers SQL MongoDB and in fact database can be used by our application both on the server side and client side. In these next few modules we will be looking at databases that we use on the server side right because that kind of what being discussed so far.

(Refer Slide Time: 11:46)



Using a database

1. We already understood that if we have data in the right data structures in our server side code, we can achieve whatever we want
   a. Articles object
   b. Counter integer
   c. Names array
2. We now just need to get the right data into our data structures
3. Process:
   a. Connect to the database
   b. Whenever required, query from the database
   c. Load the results into a data-structure that your server-side code can manipulate

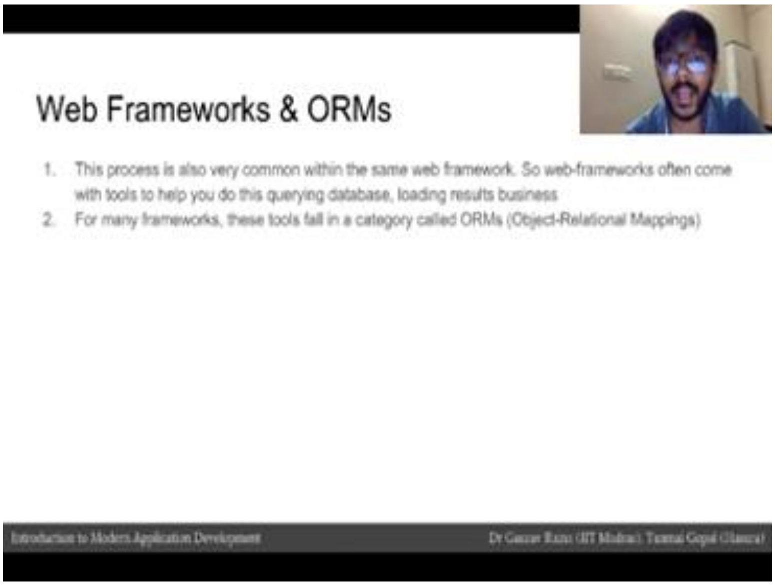
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

We already understood that if we have data and the right data structures in our sever set code we can sought of achieve what we right. So, we had articles objects we had an array of names we had a counter integers. So, we are able to get a work done once we have the data (Refer time: 12:00) structure in our code right. So, what we kind of need to do when we use a database is essentially get the right data from the database into structures and once, inside a data structures we can do what we want. So, the heart beat is contacting the database and getting the right data and putting that inside the data structure that we can use easily.

And this process is reasonably simple what we need to do connect the database from our application and when ever required we quarry the database and we look the results from this quarry into a data structure that our court can manipulate.

(Refer Slide Time: 12:33)



Web Frameworks & ORMs

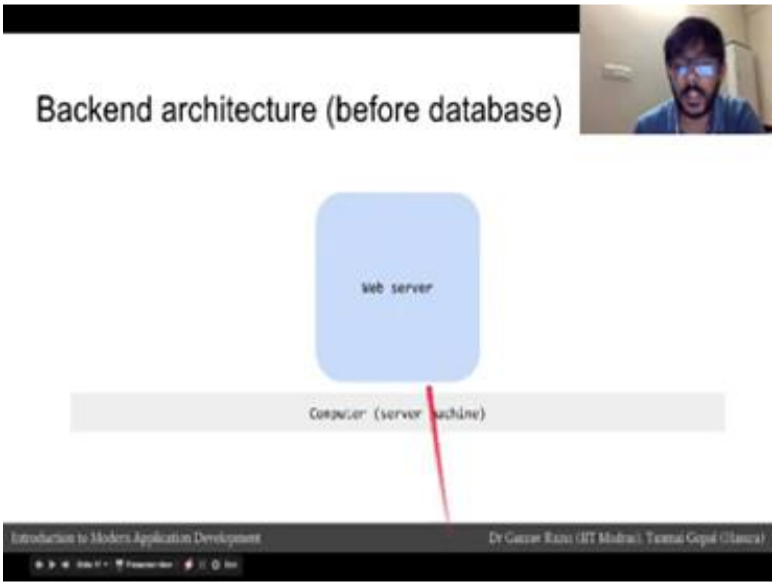
1. This process is also very common within the same web framework. So web-frameworks often come with tools to help you do this querying database, loading results business
2. For many frameworks, these tools fall in a category called ORMs (Object-Relational Mappings)

Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

This this process is also quite common the process of connecting to the database making a query and loading results into data structure into a programming language environment is also fairly common process; almost all major web frame box come with certain type of tool that come with certain packages and libraries make this easy for us and there are several tools that exist outside this frame box also that make it very easy for us to connect the database from (Refer time:13:00) a programming language environment, make a query and get results. And these tools or libraries fall into a category of software called ORMs.

(Refer Slide Time: 13:08)



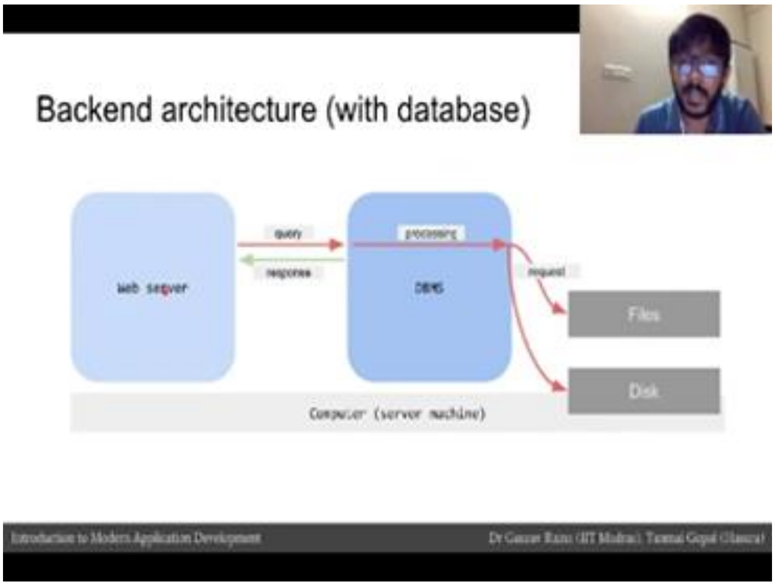
Backend architecture (before database)

Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

Before we have a database and this is kind of server site set up look like, there was a computer which is a server machine and top of that was a software which is a web server right once we have a database our web server is should be able to make the request to database load a data and then use that to process request and give out responses.

(Refer Slide Time: 13:29)



Backend architecture (with database)

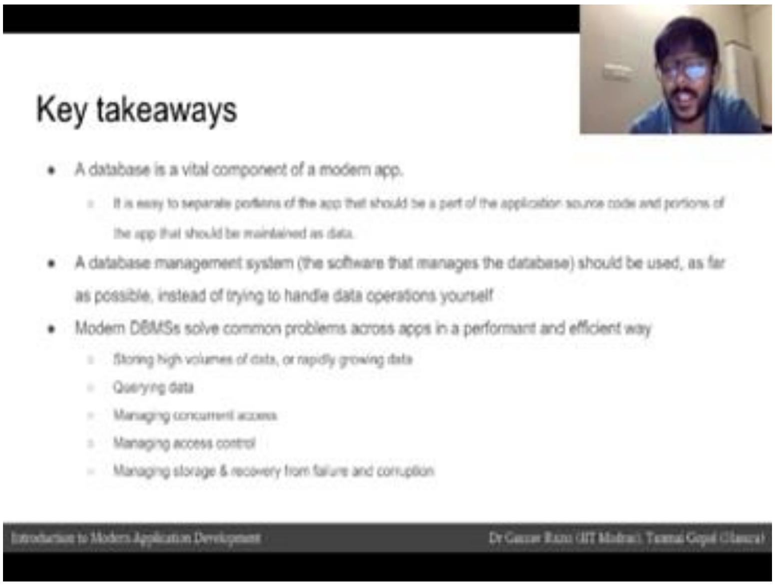
Introduction to Modern Application Development

Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, this is what that setup looks like there is software called DBMS which is here my s q l among d b and this is your web server. So, web server make to the quarry to the database the database process that quarry contacts various files contacts to disk to obtain the results of that quarry and gives that response to the web server and then the web server process is that response loads into data structures, process is it and uses the potentially to give a response to the client.

This architecture of making quarry and response is (Refer time: 14:00) also the client server of architecture and the database management system is also in fact a server it is not a web server, but it is also a server. So, the web server contact the database server to make the quarry and get response in fact because the database management system is a server, it does not actually need to leave on this and this on the same machine the DBMS can be on a different machine and the web server can contact the database server on a different machine and give the response.

(Refer Slide Time: 14:28)



Key takeaways

- A database is a vital component of a modern app.
  - It is easy to separate portions of the app that should be a part of the application source code and portions of the app that should be maintained as data.
- A database management system (the software that manages the database) should be used, as far as possible, instead of trying to handle data operations yourself
- Modern DBMSs solve common problems across apps in a performant and efficient way
  - Storing high volumes of data, or rapidly growing data
  - Querying data
  - Managing concurrent access
  - Managing access control
  - Managing storage & recovery from failure and corruption

Introduction to Modern Application Development Dr Gaurav Raina (IIT Madras), Tanmai Gopal (Hasura)

So, that is also completely possible in this quick introduction to databases these are the key points to take way a database is a vital component of modern application and it should be easy for us to understand to how to separate portion of the app that should be part of the application source code and part of the application and parts of the applications that should be maintained is data.

We also understood what database management system is, and that in database management systems should be used as far as possible instead of trying to data application our self because it can get ready hard we also understood that modern (Refer time:15:00) one DBMS are common problems across the requirement for building an application and this sought of common problems they solved are storing higher volume of data, of quarrying data managing concurrent access managing access control and security and managing storing and recovery from failure in corruption and of few cases.