INDIAN INSTITUTE OF TECHNOLOGY MADRAS • सिद्धिर्भवति कर्मजा •

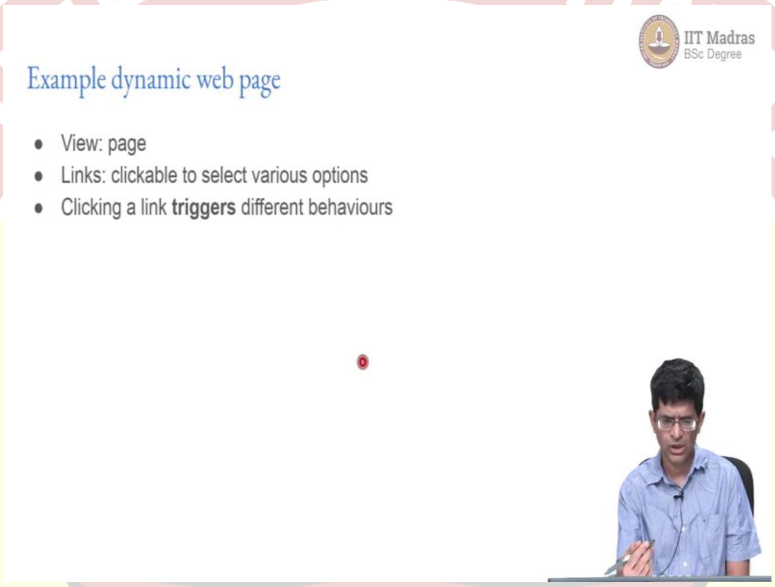
IIT Madras

ONLINE DEGREE

Modern Application Development 1
Professor. Nitin Chandrachoodan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Requests and Responses

Hello everyone and welcome to this course on Modern Application Development. So, to understand how MVC or models like that can be used in the context of a web application, we need to understand Requests and Responses.

(Refer Slide Time: 0:25)



Example dynamic web page

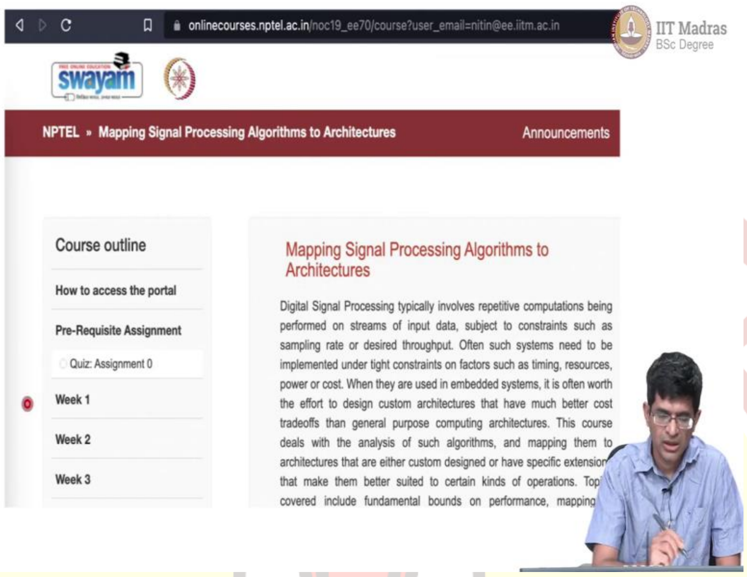
- View: page
- Links: clickable to select various options
- Clicking a link **triggers** different behaviours

And we can take the example of a dynamic web page, the view over here is essentially the page as presented to the user and what happens in a web page, what do you really have to interact with, there are clickable links and clicking on each link will take you typically to a new page, or it might just select certain kinds of options and modify the existing page in some ways, by the original definition of the web of course, clicking on a link just takes you to a new page, it creates a new request to the web server and most likely, what the server is going to do is go back load the page corresponding to that and send that back to you.

Because all the web server was doing was picking up HTML files and sending them back to the user. Now at present, things have changed a little bit, of course, servers tend to dynamically generate the messages that need to be sent back to the user. So, how does that work in practice?

In general, now you need to start thinking of clicking a link, as not just requesting a new page, but triggering certain kinds of behaviour, it can trigger changes on the server side, but at the end of the day, finally it will come back to the user as something that they see.

(Refer Slide Time: 1:37)



onlinecourses.nptel.ac.in/noc19_ee70/course?user_email=nitin@ee.iitm.ac.in

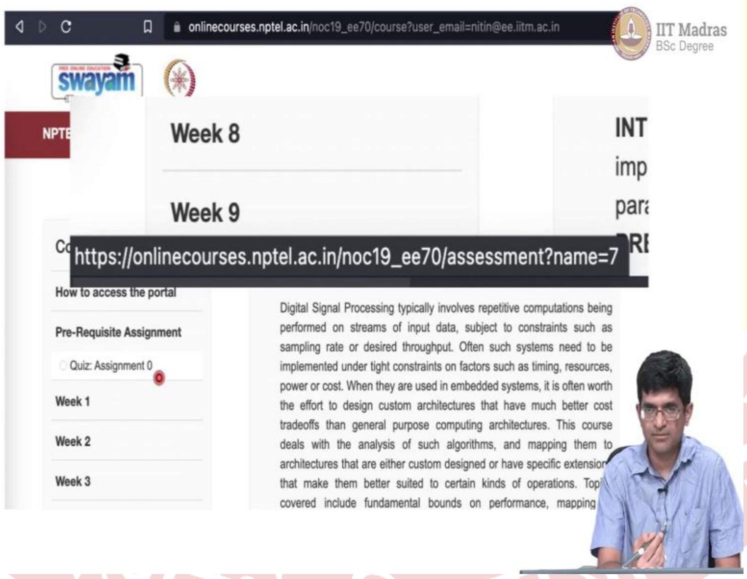
IIT Madras BSc Degree

NPTEL » Mapping Signal Processing Algorithms to Architectures — Announcements

Course outline
- How to access the portal
- Pre-Requisite Assignment
- Quiz: Assignment 0
- Week 1
- Week 2
- Week 3

Mapping Signal Processing Algorithms to Architectures

Digital Signal Processing typically involves repetitive computations being performed on streams of input data, subject to constraints such as sampling rate or desired throughput. Often such systems need to be implemented under tight constraints on factors such as timing, resources, power or cost. When they are used in embedded systems, it is often worth the effort to design custom architectures that have much better cost tradeoffs than general purpose computing architectures. This course deals with the analysis of such algorithms, and mapping them to architectures that are either custom designed or have specific extension that make them better suited to certain kinds of operations. Top covered include fundamental bounds on performance, mapping



onlinecourses.nptel.ac.in/noc19_ee70/course?user_email=nitin@ee.iitm.ac.in

IIT Madras BSc Degree

Week 8

Week 9

https://onlinecourses.nptel.ac.in/noc19_ee70/assessment?name=7

Course outline
- How to access the portal
- Pre-Requisite Assignment
- Quiz: Assignment 0
- Week 1
- Week 2
- Week 3

Digital Signal Processing typically involves repetitive computations being performed on streams of input data, subject to constraints such as sampling rate or desired throughput. Often such systems need to be implemented under tight constraints on factors such as timing, resources, power or cost. When they are used in embedded systems, it is often worth the effort to design custom architectures that have much better cost tradeoffs than general purpose computing architectures. This course deals with the analysis of such algorithms, and mapping them to architectures that are either custom designed or have specific extension that make them better suited to certain kinds of operations. Top covered include fundamental bounds on performance, mapping

So, I just want to sort of pinpoint a couple of specific examples over here. What I have is, I mean this was a course on NPTEL, that I had taught a couple of years ago and you will notice that you know, this is the view, this is the page that is presented once a user logs in, so there is clearly some content over here, there is heading, which basically gives the title of the course and also some kind of navigation information, there are certain announcements and some other tabs up there on top.

Out here is there is a course outline. So, on the left hand sidebar, there is a course outline, which helps us to navigate through the different weeks and over here, there is the overall body of the page, which sort of tells us okay, this is the content, this is what you expect to read.

Now, this is one way of laying out a page, you might find that, for example in certain cases, this could have got collapsed into the side somewhere else, depending on how you are trying to make this mobile friendly, either the title and navigation bar could have been made into, the prebar icon or something like that.

But broadly, this is a useful layout, when you are trying to convey something that, the bulk of the information occupies a large part of the page out here and the rest of it essentially acts as some kind of navigation, could this have been done better, quite possible, I mean, maybe you did not need so much space for the navigation, perhaps this could adapt differently, depending on the screen size.

What I have put out here is only a part of the total screen, by the way. So, it is not as the navigation actually takes up one fourth or one third of the screen space. So, there could have been some changes. But again, it is subjective at that point. Now, the interesting thing to note over here is right on top, you can see the URL. In this case, it says onlinecourses.nptel.ac.in, as you know, that essentially refers to the server itself.

So, that is the domain name of the server that hosts this application. Is it one single machine somewhere? Is it a whole bunch of machines somewhere in the Google Cloud? We do not know and we do not actually care. All that it matters is any requests sent in. So, the first thing that could happen is I would need to look up the IP address corresponding to this domain name, send a request to that IP address.

And somebody at the other end, is it like one server or you know, one out of n servers? We do not really know, we cannot know and we do not really care. One of those servers will respond. Now, what does a server need in order to respond? It needs further information about what kind of page I am looking for and that essentially comes once again, as part of the URL.

So, this /noc19_ee70 was the code given to the course. So, you can see over here that this part of it appears right here and under that it also has another part of the URL, it says, underscore, slash(/) course. So clearly, what it is telling you is that I want to go to this server. I want to ask it information about this course and what kind of information about the course well this last part out here, the user email is probably optional, it should not ideally be required in this, because after all, I am logged in when I am looking at this.

So, probably even without having that information, it should have been able to show me the same page. But when I go to this page, essentially it knows that it is supposed to show me the outline corresponding to the course and one of the links that I have over here, for example corresponding to assignment 0, if I look at it, and sort of zoom in further, I find that that link essentially corresponds to this, what I have shown here is just a zoomed in version of the URL, which sort of appears only at the bottom of the screen.

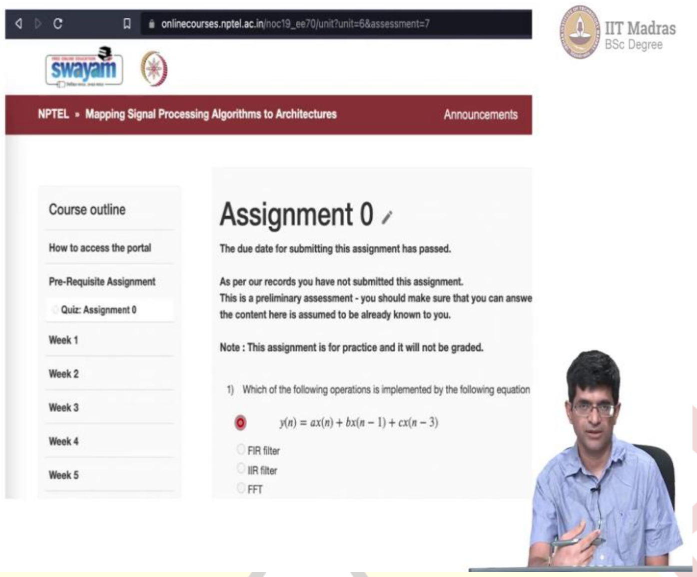
Now, you can see that that URL has a similar structure, it once again has the same domain name for the server. Once again, it has the noc19_ee70, which sort of tells the server that, this is what I am interested in. But now instead of course, it says this is assessment. So, a quiz or an assignment is essentially related to the assessment data type inside my system.

So in other words, as far as the user is concerned, they see something referred to as a quiz. But inside the system, it is referred to as an assessment. Why choose these names? That depends on the developer. At the end of the day, most users are not really looking at the URL, they are interested in this. As long as they understand what is written over here, there could be changes over here, which you are not particularly concerned about.

Having said that, it does make sense to give it a logical name, so that when you look at it and even if someone looks at the URL, they should be able to get an idea of what you are trying to do. So in other words, what you can see is this URL has encoded some information, which is then sent to the server and appropriately pulls it out.

So, just by having that information there, the request that is sent to the server, which will basically contain this entire path name out here, will have enough information that the server can respond appropriately.

(Refer Slide Time: 6:53)


onlinecourses.nptel.ac.in/noc19_ee70/unit?unit=6&assessment=7

NPTEL » Mapping Signal Processing Algorithms to Architectures

Announcements

Course outline

How to access the portal

Pre-Requisite Assignment

Quiz: Assignment 0

Week 1

Week 2

Week 3

Week 4

Week 5

Assignment 0

The due date for submitting this assignment has passed.

As per our records you have not submitted this assignment.
This is a preliminary assessment - you should make sure that you can answe
the content here is assumed to be already known to you.

Note : This assignment is for practice and it will not be graded.

1) Which of the following operations is implemented by the following equation

$y(n) = ax(n) + bx(n-1) + cx(n-3)$

FIR filter

IIR filter

FFT

Now, if I actually click on that link, what I will find is that, you know, it basically loads assignment 0 and of course, in this case, the due date has passed. So, it does not really allow me to do very much. But you can see that on this page now, there are a few other options. So, it has, for example a choice button, which allows me to click on something and in fact, if I go look at the URL out here, I see that now it is looking at something called unit.

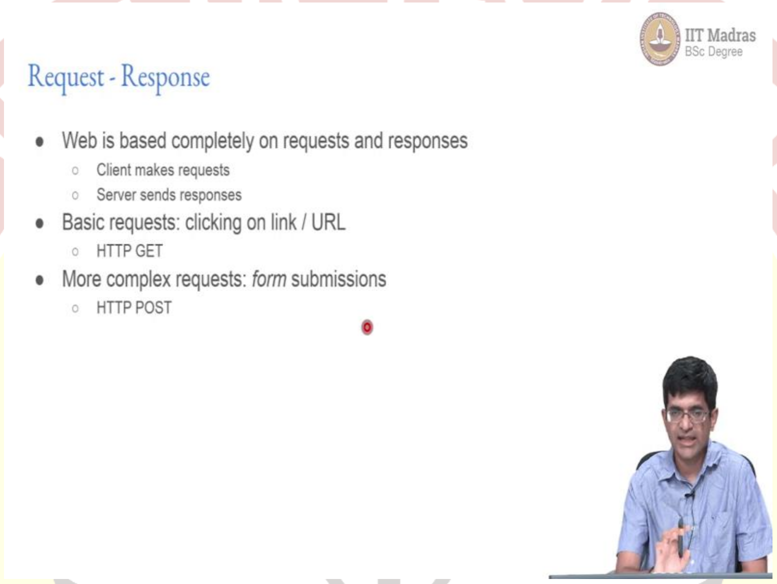
So under assessment, it then sort of goes under unit, which is basically one of the modules which is used inside this. So, if you start digging into the code of what is happening out here, you will find that ultimately, this is also building upon a similar kind of framework. There are models called units, some of those units have the type assessment, the assessments basically have certain kinds of views associated with them, which basically allows me to show it to you like this.

If it was on a mobile phone, it would probably look slightly different. So, the view could be adapted depending on what the user is using to look at it and finally, this part of it, which is in the URL, is actually each time I click on something, it is telling me what needs to be done. So, clicking on quiz, assignment 0, for example is a way of invoking a controller that tells the server that I now need to pull out data from the model corresponding to assignment 0 and send back that view.

So, the controller decides which model needs to be sort of poked to get data out of it and which view needs to be rendered to the user. So, it is tied tightly to both. But it is the controller that actually decides what shows up at the end. Similarly, if I go click on one of these buttons, there will be an associated controller. That could be, because of the way the web works, it might be that, it just allows me to select all the buttons and then click on submit or alternatively, it might be that every time I click on a button, it automatically invokes a controller, which does something.

Both of those are options. But you can see what is happening, there are requests being sent each time we want a change back to the server and the server's response is what finally shows up at our end.

(Refer Slide Time: 9:03)



Request - Response

- Web is based completely on requests and responses
  - Client makes requests
  - Server sends responses
- Basic requests: clicking on link / URL
  - HTTP GET
- More complex requests: *form* submissions
  - HTTP POST

So, the web because it is based completely on this idea of clients, making requests and servers sending back responses, means that if we want to implement MVC or any other kind of model like that, we need to do it using this pattern. So, the approach that has been taken is that, basic requests, such as just clicking on a link or a URL will essentially invoke a HTTP GET request. That is the most basic form of HTTP.

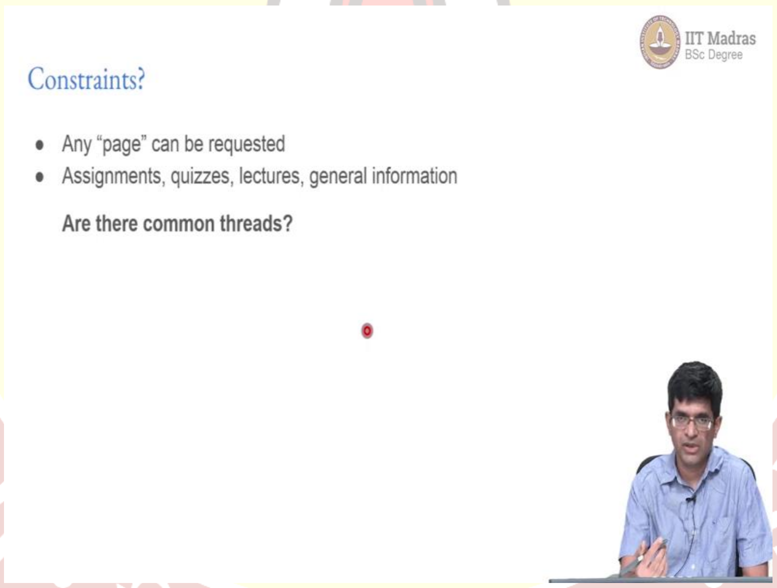
The most fundamental thing that you can do in the hypertext transfer protocol is to request a web page, request a URL. The call, we automatically tend to say request a web page, but in fact you are only requesting, sending a request for a URL. Is that actually a page or is it something that just does something at the back end? It may not even send you back any information, that is why I am saying that there is a difference between requesting a page and requesting a URL.

All that you can do is request a URL. In most cases, that will give you back a new view, which we will think of as a page. Now, that is the most basic form. But clearly more complex approaches can be taken as well and the sort of most common one that we know of is a form submission.

A form could be something which basically allows you to type in your name, some information about yourself, maybe a small text box that allows you to, fill out a little bit of detail and finally, click on submit and in most cases, when you have a form of that sort, you will find that the HTTP method that is used in order to communicate back to the server is not a GET, but instead a POST request and the reason for using post is that it allows multiple pieces of data to be encoded neatly into the message to be sent without sort of making a mess of a URL that you are sending across.

So, the URL looks clean. The posted data is like nicely encapsulated, encoded to prevent sort of mangling in different ways and posted into the server, so that the server can just pull that out and do what it needs to do with it.

(Refer Slide Time: 11:11)



Constraints?

- Any "page" can be requested
- Assignments, quizzes, lectures, general information

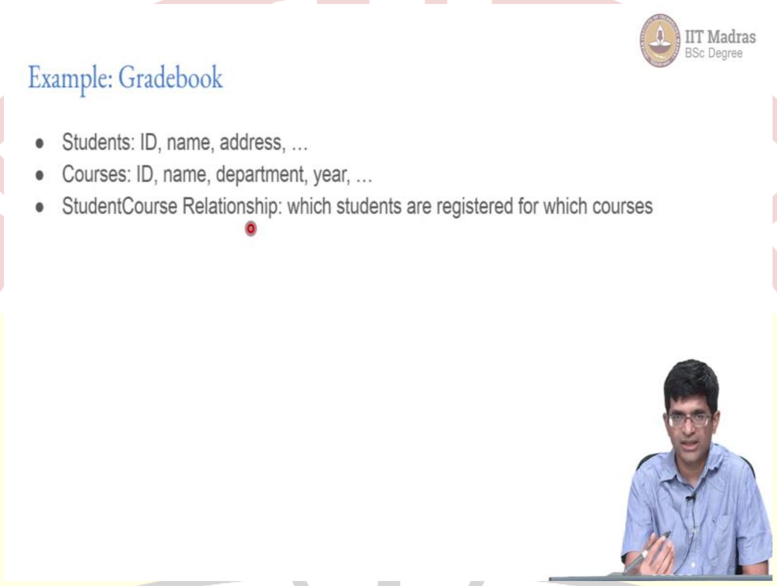
Are there common threads?

So, what this means is that any page or any URL can be requested and in the example that I gave, you could have assignments, you could have quizzes, you could have lectures, you could have general information about the course or about the website itself, all of those could finally be something that comes back as a response to a click.

So, one question that we can ask is, so fine, I build the SYAYAM platform. It has all this, quizzes and lectures and various other things. I then go and decide to build something else, maybe a shopping cart application, I decide I am going to recreate Amazon or Flipkart. Now, once again over there, instead of quizzes and lectures and assignments, I have objects that can be bought, I have shopping carts, I have users instead of students, but very similar kinds of ideas.

So, the question that comes up basically is, are there common threads that we can try and exploit or here to reduce sort of duplication of effort and once again, see what are we looking for, a design pattern.

(Refer Slide Time: 12:17)

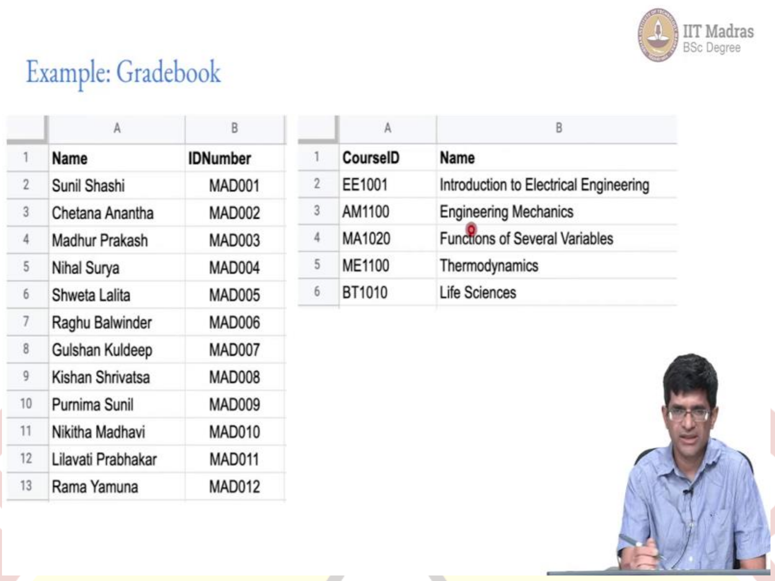


Example: Gradebook

- Students: ID, name, address, ...
- Courses: ID, name, department, year, ...
- StudentCourse Relationship: which students are registered for which courses

Now, going back to the example that we had, when we were talking about models, we were talking about a grade book, where we would have students, the student's ID, name, address, etc, will be stored in the student model, we would have courses which once again, would have a unique ID, but then also a name or department year offered and so on and then a student course relationship, which is not exactly a model, it is a table in the database, but usually not thought of as a separate model in itself. Which basically tells you which students are registered for which course, how many marks they got and so on.

(Refer Slide Time: 12:54)



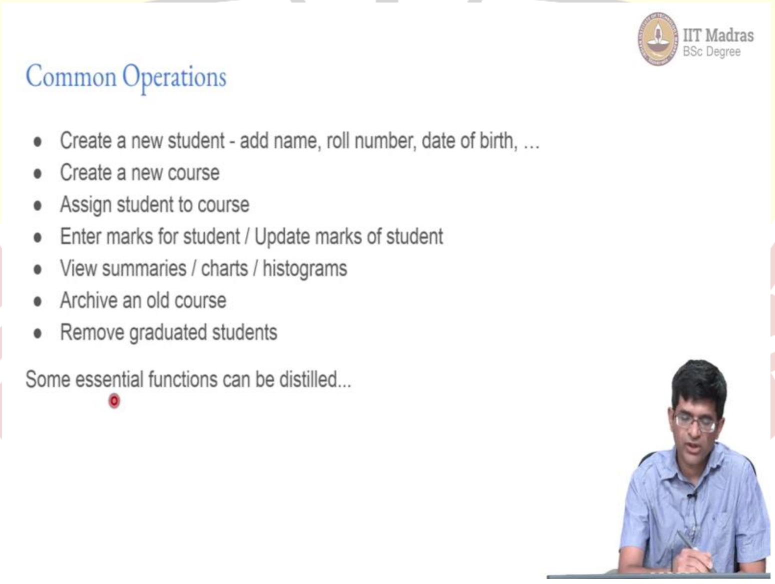
Example: Gradebook

	A	B
1	Name	IDNumber
2	Sunil Shashi	MAD001
3	Chetana Anantha	MAD002
4	Madhur Prakash	MAD003
5	Nihal Surya	MAD004
6	Shweta Lalita	MAD005
7	Raghu Balwinder	MAD006
8	Gulshan Kuldeep	MAD007
9	Kishan Shrivatsa	MAD008
10	Purnima Sunil	MAD009
11	Nikitha Madhavi	MAD010
12	Lilavati Prabhakar	MAD011
13	Rama Yamuna	MAD012

	A	B
1	CourseID	Name
2	EE1001	Introduction to Electrical Engineering
3	AM1100	Engineering Mechanics
4	MA1020	Functions of Several Variables
5	ME1100	Thermodynamics
6	BT1010	Life Sciences

And this was sort of the structure that we use in order to represent it, basically tables, you have one table, which has names and ID numbers for students and other one which has course IDs and course names and other things that sort of had the relationships between them and so on.

(Refer Slide Time: 13:08)



Common Operations

- Create a new student - add name, roll number, date of birth, ...
- Create a new course
- Assign student to course
- Enter marks for student / Update marks of student
- View summaries / charts / histograms
- Archive an old course
- Remove graduated students

Some essential functions can be distilled...

But the important point is that, you can think of a certain typical set of operations. As a user of such a great book application, one of the things I would want to do is, to be able to add new students to the system. So, then what do I need to do, I need to type in their name, roll number, date of birth, maybe address things of that sort, I want to create a new course. Which department? Which year is it offered? What is the name of the course? Are there any prerequisites? Things of that sort.

I want to be able to assign students to courses. I want to be able to enter marks for a student at the end of the course or maybe, let us say there was a makeup exam or a supplementary, update certain marks, change the marks of the student, see what happens. I also want to be able to view summaries or histograms of marks, charts showing the past percentage or the average marks.

Once I am done with a course I would probably want to archive an old course, make it either read only or even get rid of it altogether and similarly, once a student graduates probably there is no further need to have them around in the system, can I sort of remove their information? Obviously, I take a backup, but they do not need to be in the live system at some point.

Now, these are obvious sort of common ideas that you can think of and if you think carefully about it, there are some core functionality that can be distilled out of this. Which leads us to the idea of crud.