MongoDB

Installation :

Brew : Through this node and mongoDB : https://brew.sh

RoboMongo : It is used for browsing the content of mongoDB. Very helpful development tool

**brew install node**

**brew install mongo**

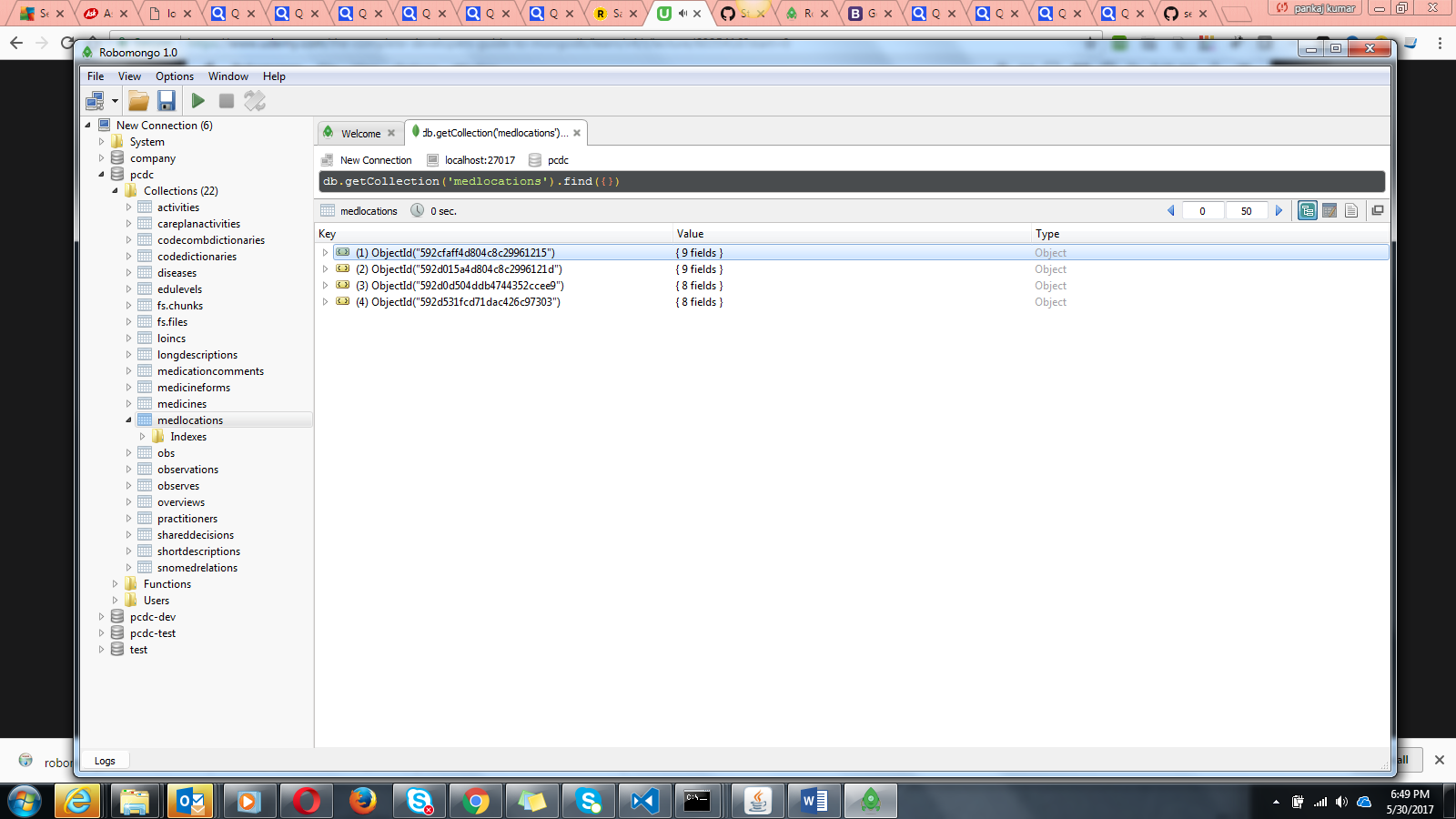
mongod : Local mongo server

/data/db inside inside c drive : **Imp**

Give permission to db directory.

Robomongo for graphical interface for mongoDB

**robomongo.org and install it.**



We can found the version inside the MongoDb/server/3.2

RoboMongo setup : (version 1.0)

Select connection and start :

Promises :

Promises interect in number one way with mongoDb.

document.querySelector(‘button’).addEventListener(‘click’,function(){

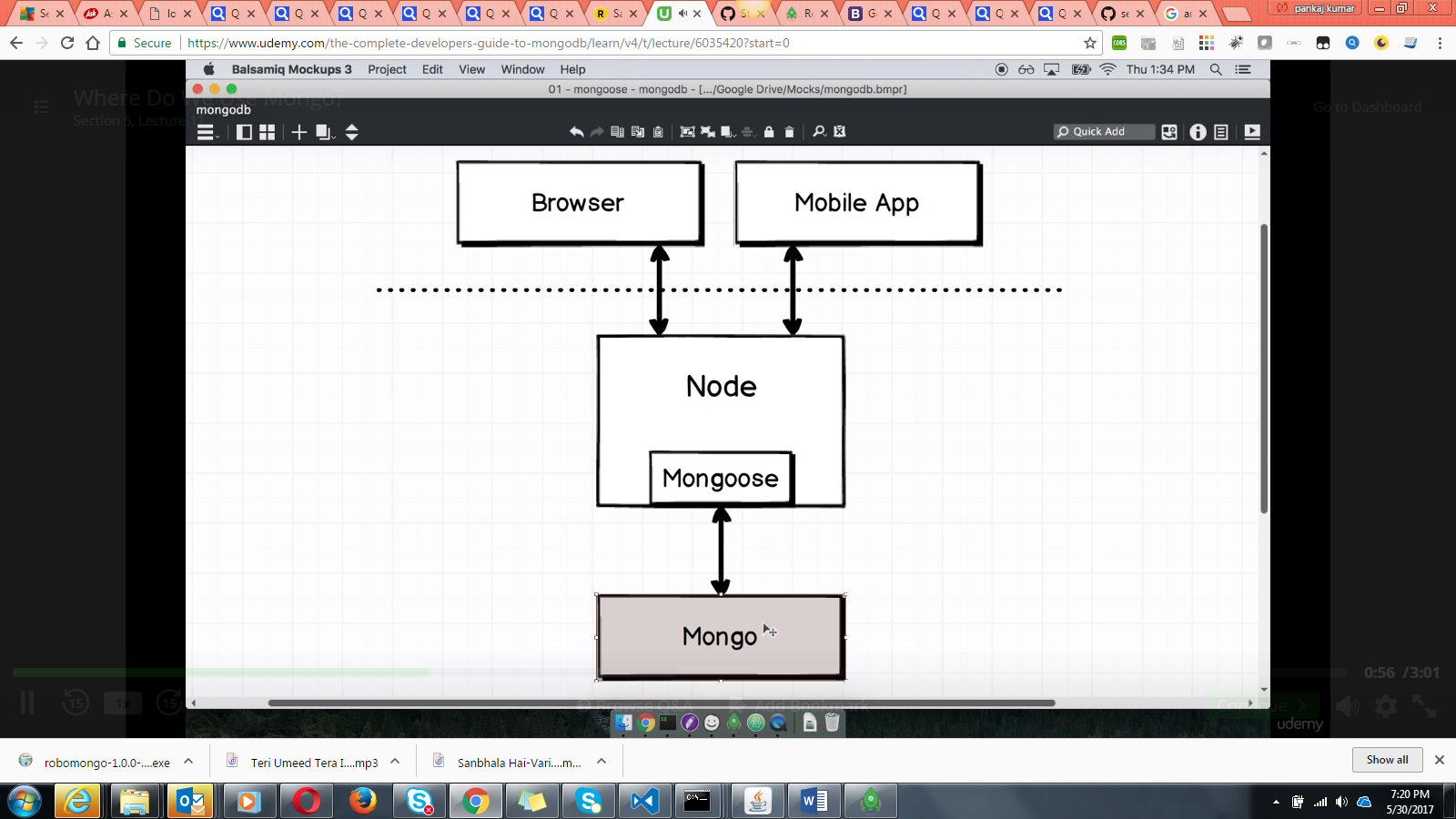
})

Or

document.querySelector(‘button’).addEventListener(‘click’, (){

})

**Where we use mongoDB**

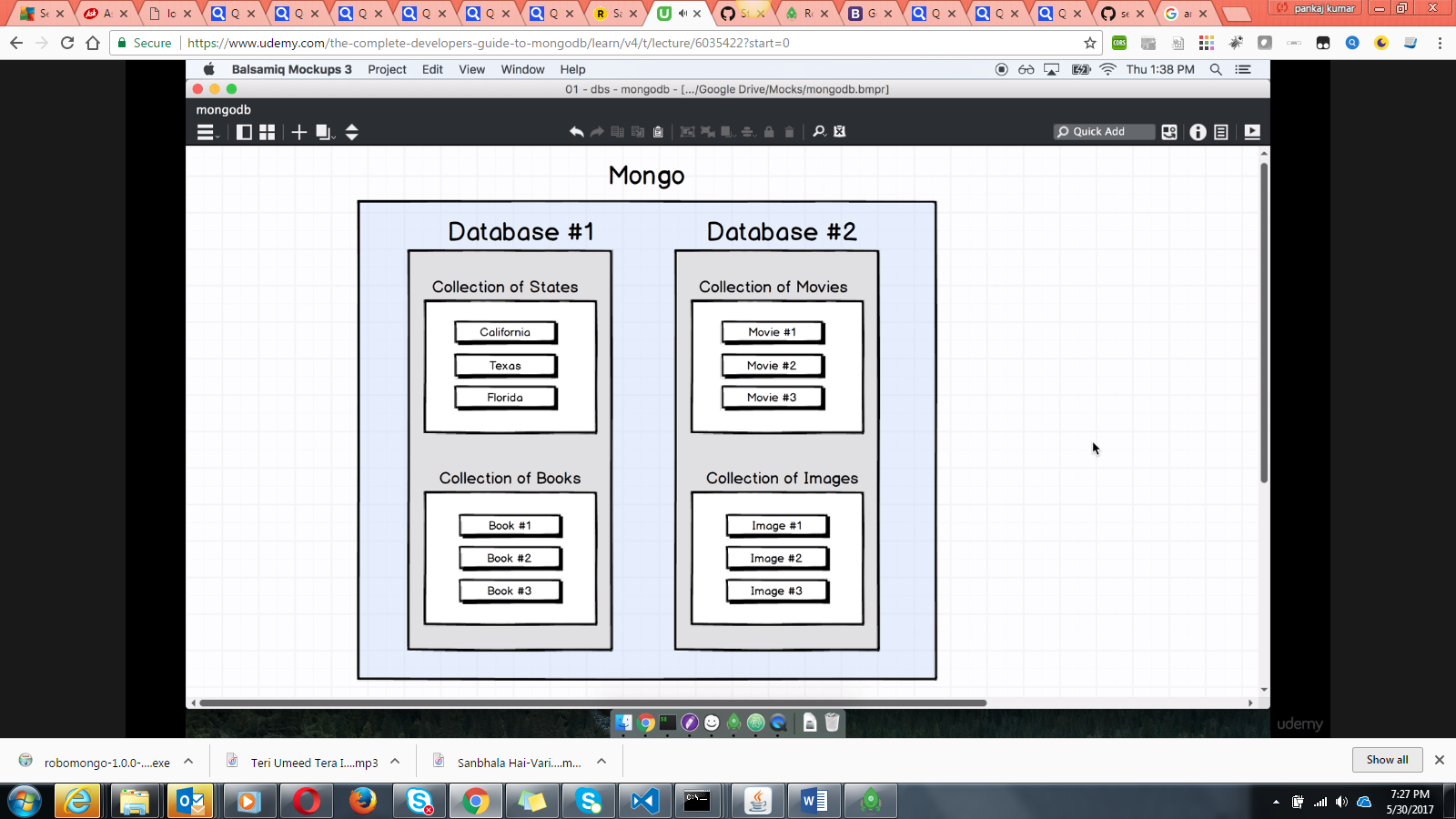


Any language work with mongo : Mongo is an orm or odm. Orm means (object relational mapper) or (object data mapping)

Note : How can we directly connect the web browser/mobile with mongoDB . No we can not do it directly . We need some webserver in between them.

Mongo allows us to have multiple internal databases internally.

We can create multiple database inside same mongo instance :

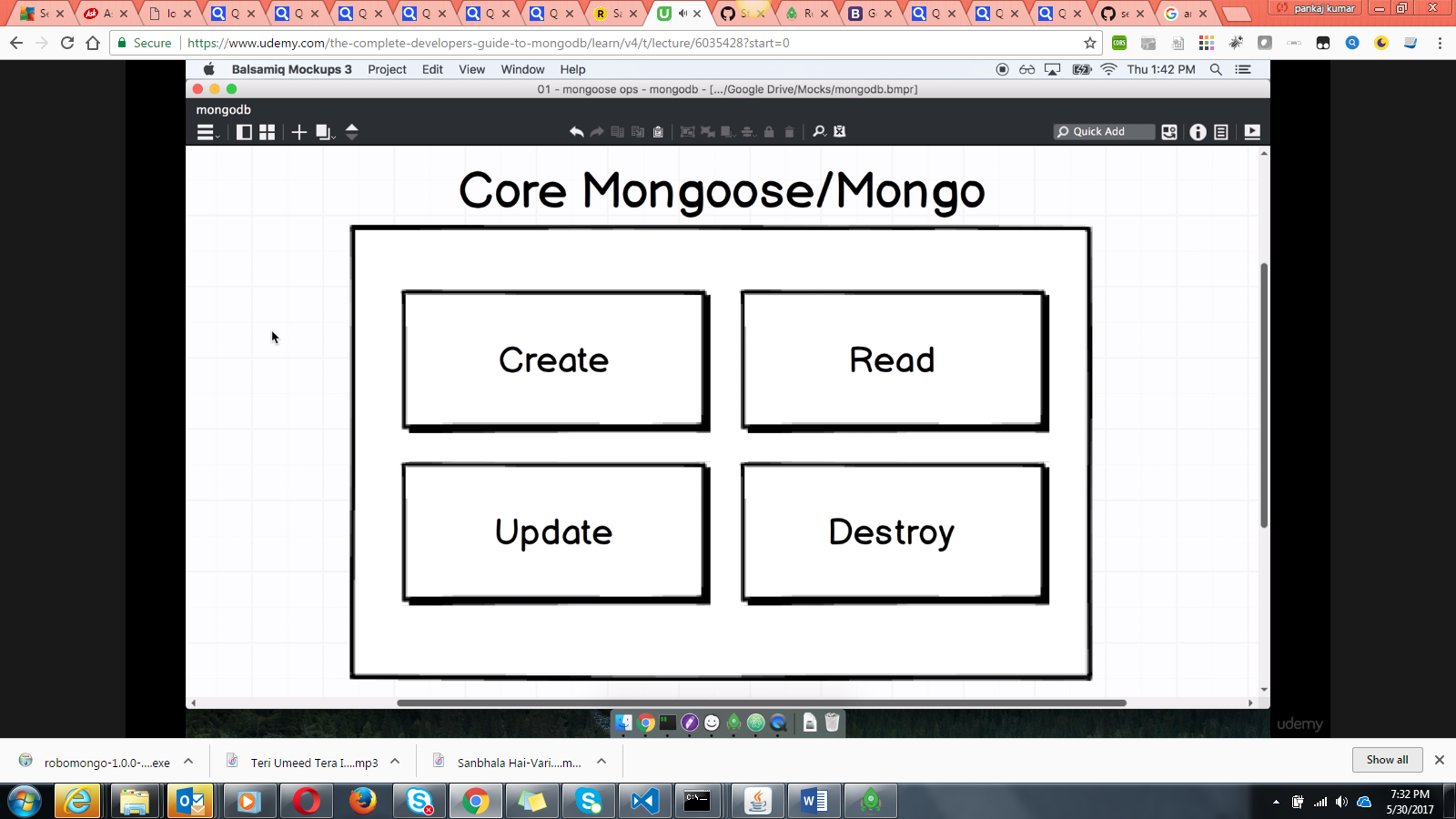


How to connect more then one database in mongoDb.?

Collections of states and collections of books



**What to do ?**



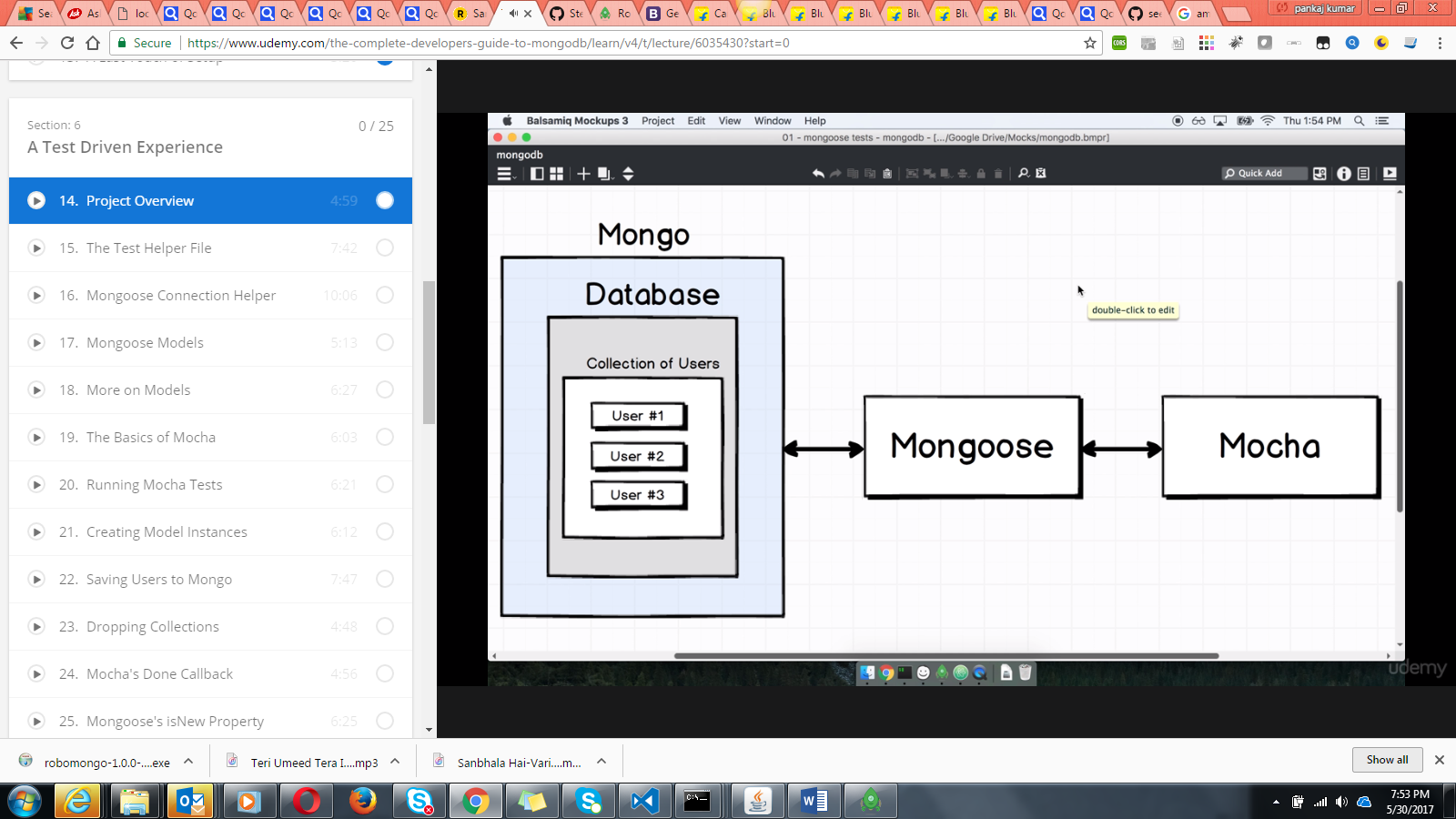
Note : Mongoose is an library that provide us api to interact with mongo.

**Test driven approach**

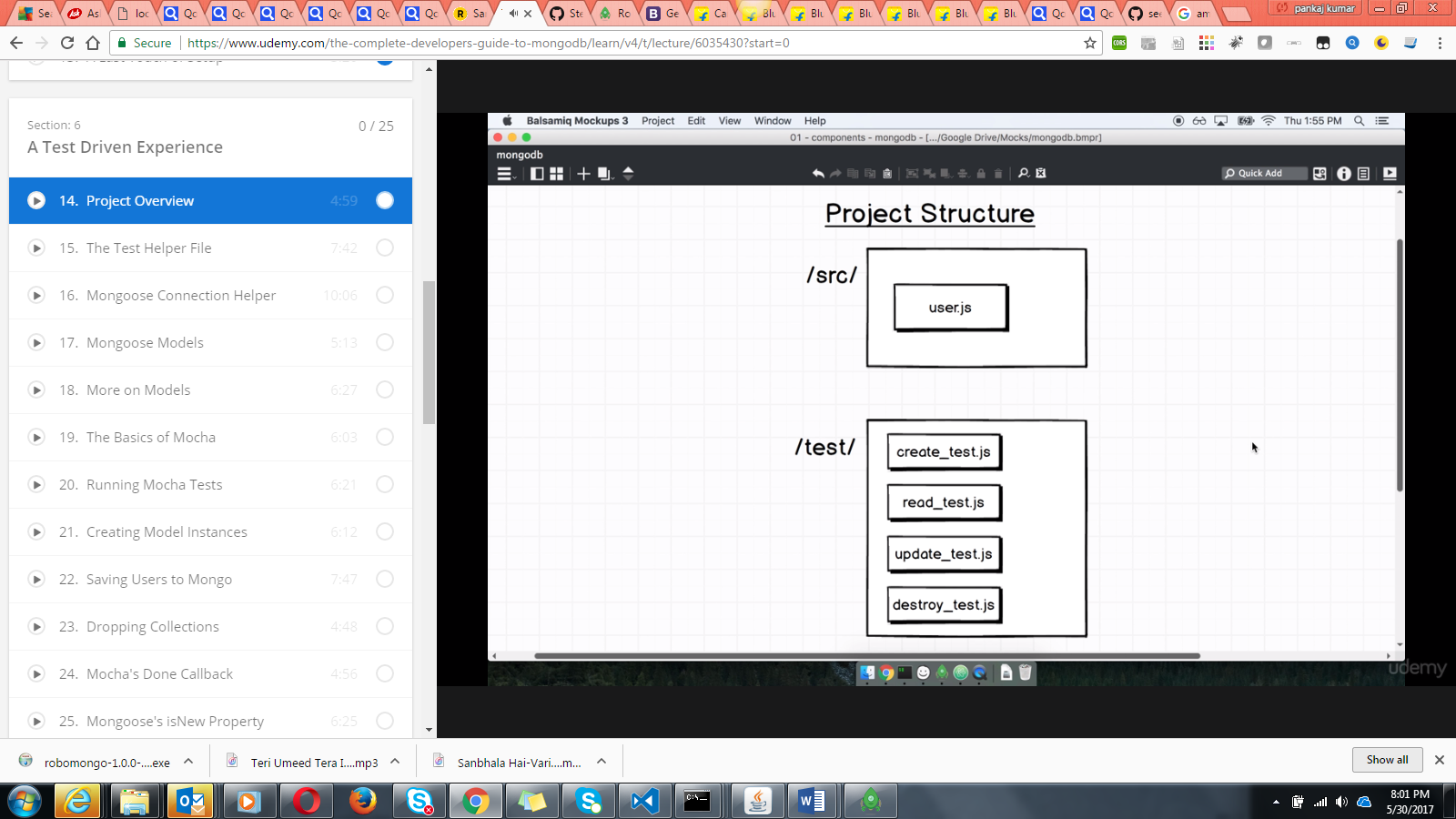
Create a project folder and then fire **npm init** inside it.

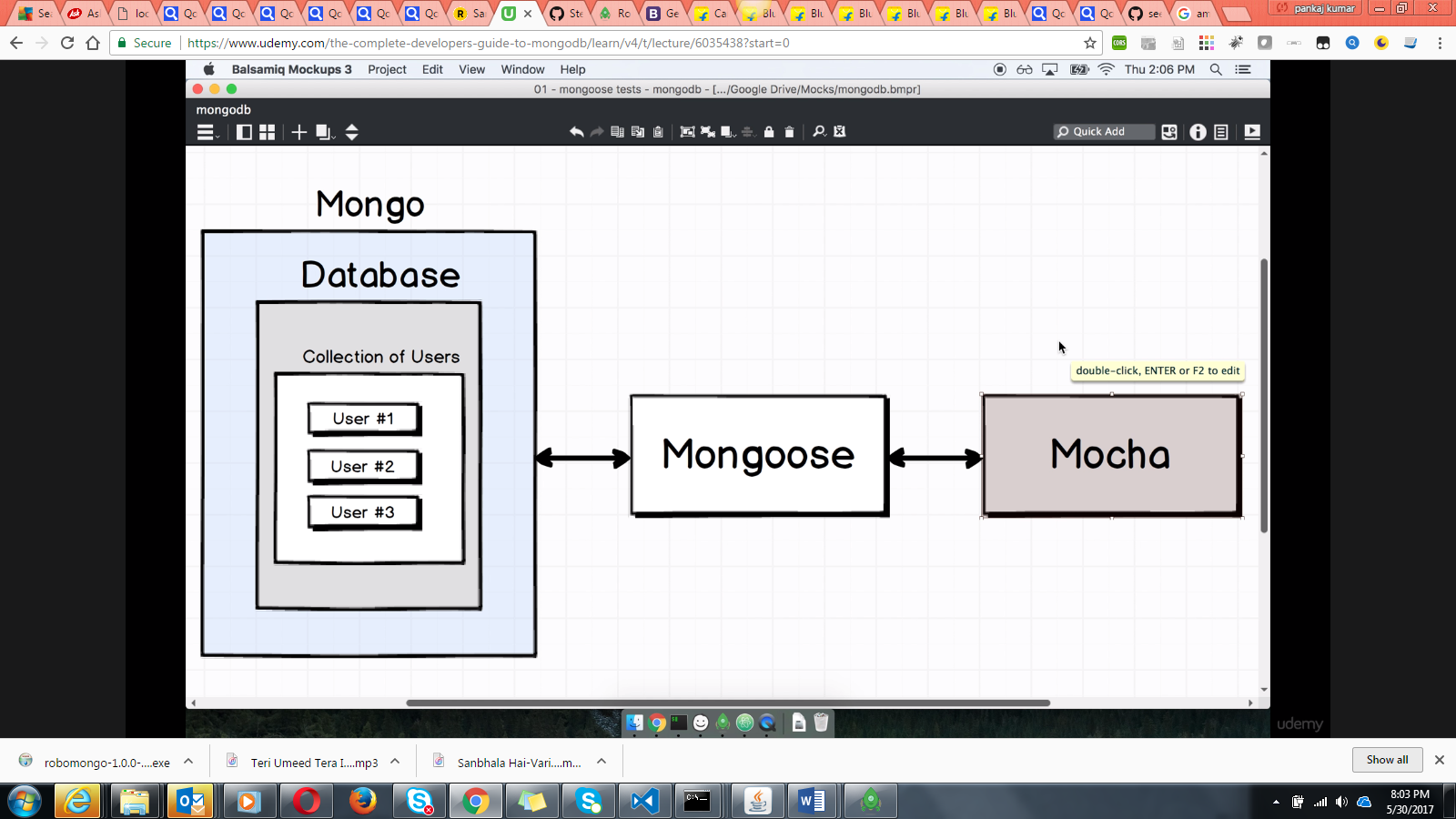
npm install –save mocha nodemon mongoose

What we are making :



The project structure :



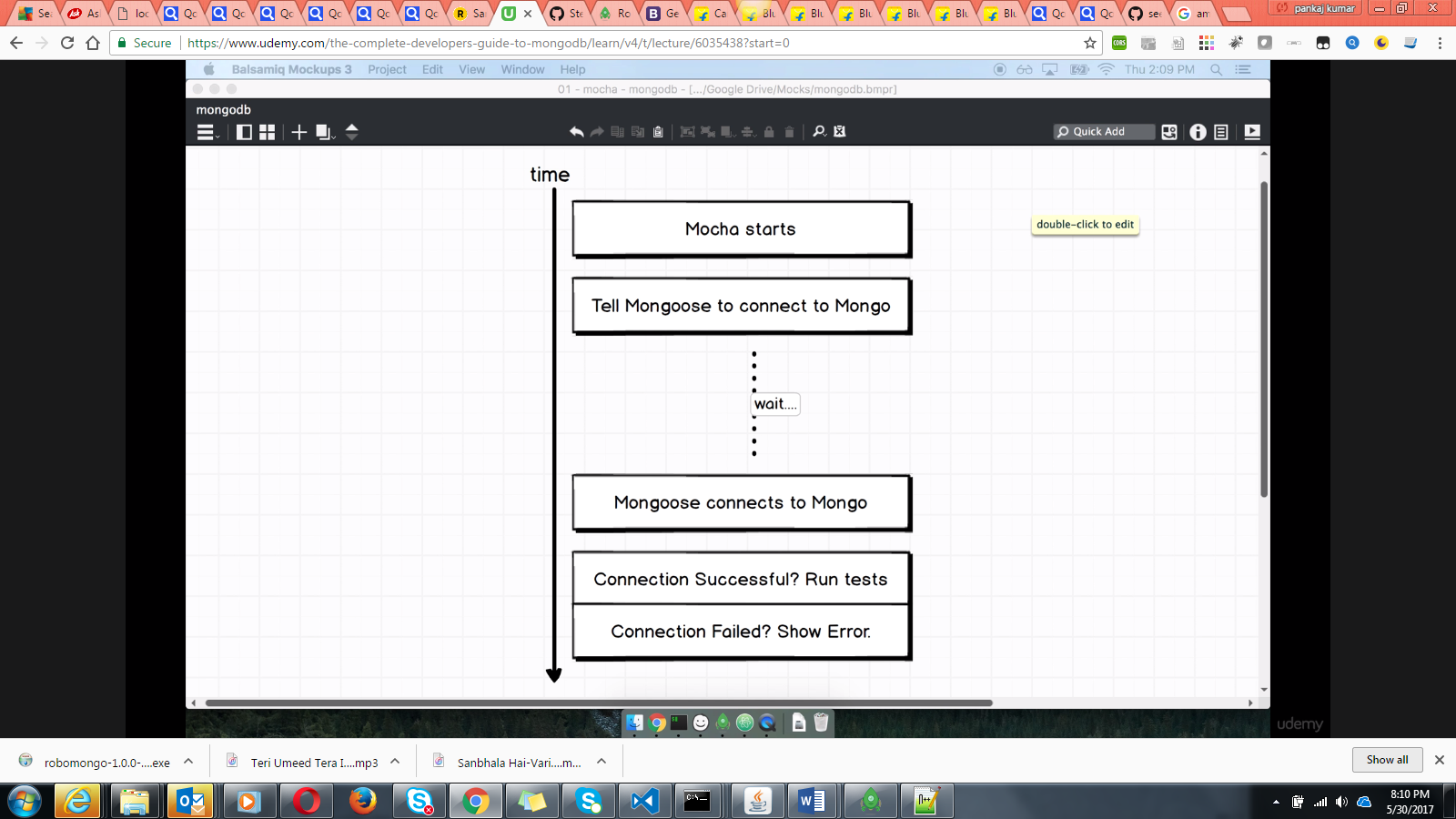


Here we are going to connect mongoDb through the Mongoose and then test it through the (**mocha testing framework**).

Mocha is used for any type of javascript testing in the node environment.

**Nodemon helps us in running the tests**

The flow :



Create afolder test and then create file **test\_helper.js**

const mongoose = require('mongoose');//const ES6 code

mongoose.connect('mongodb://localhost/eshop'); // If eshop database will not be there it will create it for us.

mongoose.connection.once('open',function(){//Here once and on are the events. It watches the mongoose once to fire the open event

console.log('Good to go !')

})

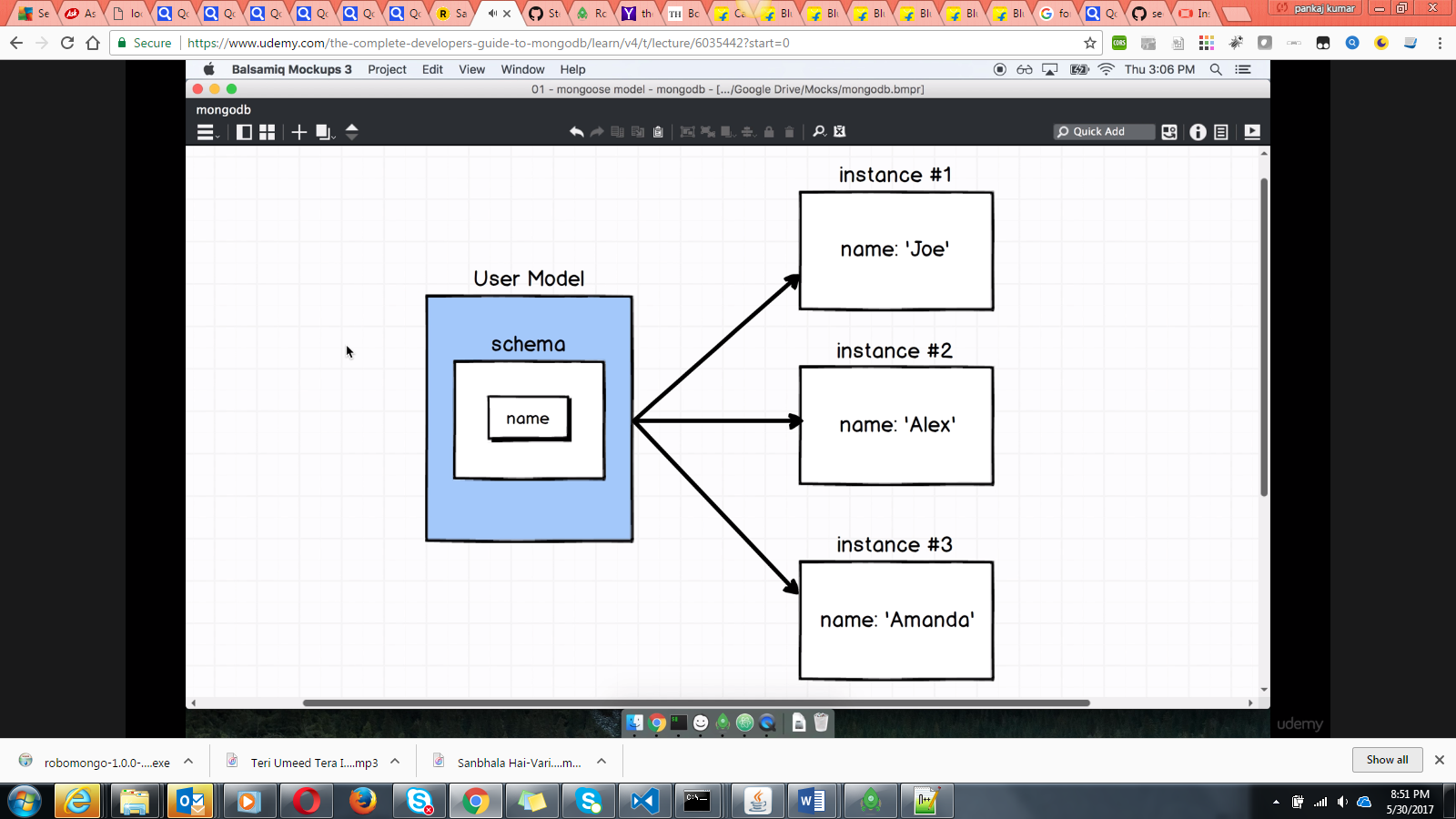
.on('error',function(err){//Also watch for mongoose to emit an event error

console.log('The error is ')

console.log(err)

});

**Mongoose models :**



Create a src folder and inside it create the user.js

const mongoose = require('mongoose');

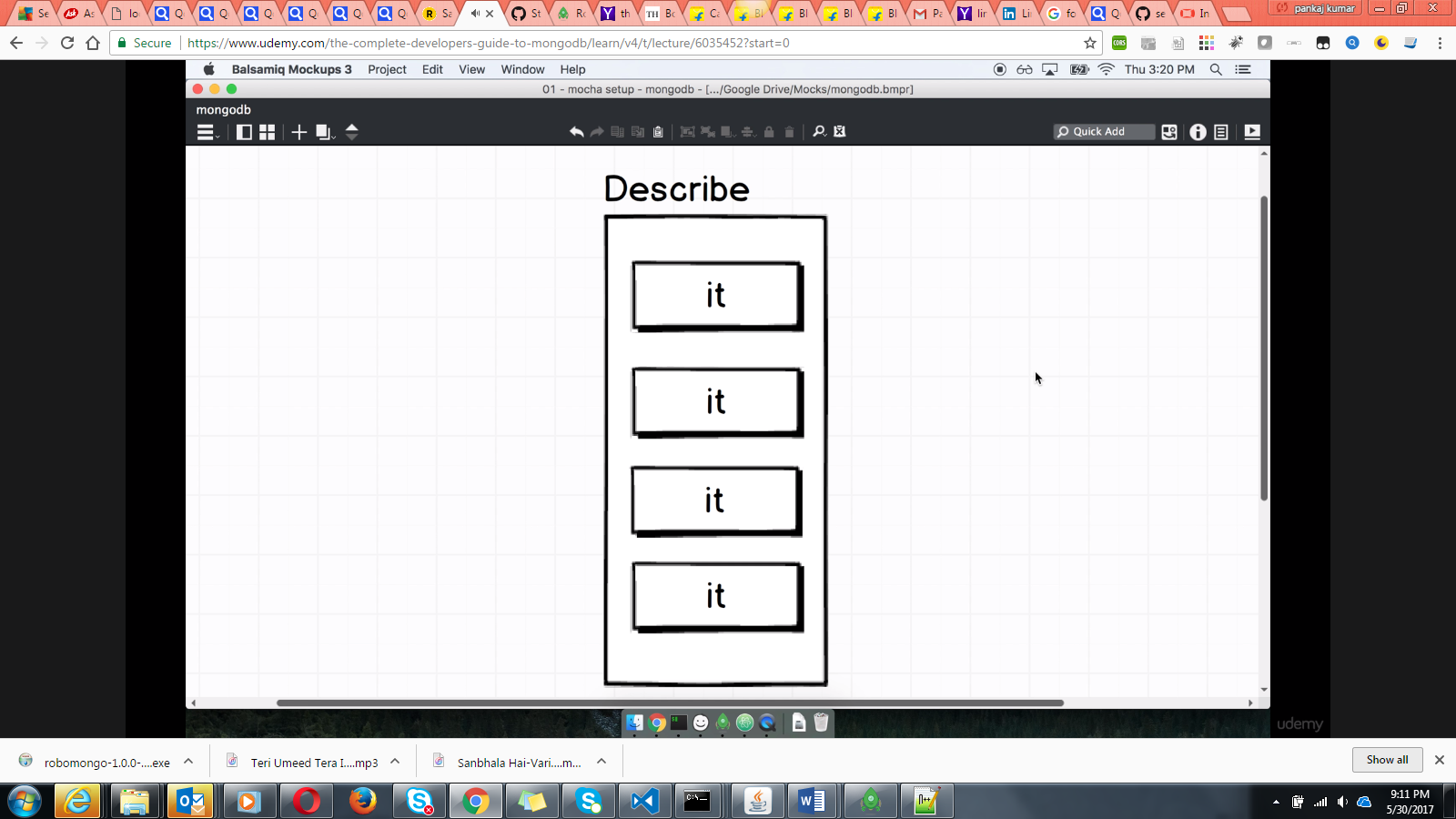
const userSchema = new mongoose.Schema({name : String});

module.exports = mongoose.model(**'User'**,userSchema);

//User : This means what is the name of collection inside the mongodb database.

//mongoose.model(**'User'**,userSchema) : refers to entire collection of data

**The basic of mocha**



Inside the describe function we have a set of it blocks. **or it functions**

Inside the it we have a very specific set of code.The describe and the it function are automatically provided to us by mocha.

**Note :** It refers to the assertions. It means this value should be equal to that value.

Create file create\_test.js inside the test folder