

CSE3002 – Internet and Web Programming

Slot: L5 + L6

Name: Rathnam Sasidhar Achari

ID: 17BCE0895

Lab Assignment – 5

Connecting to Database:

```
var MongoClient = require('mongodb').MongoClient;
```

```
var url = "mongodb://localhost:27017/mydb";
```

```
MongoClient.connect(url, function(err, db) {
```

```
  if (err) throw err;
```

```
  console.log("Database created!");
```

```
  db.close();
```

```
});
```

Screenshot:



```
Command Prompt
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Sasidhar>cd Desktop
C:\Users\Sasidhar\Desktop>node 17BCE0895.js
(node:16416) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a
future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the M
ongoClient constructor.
Database created!

C:\Users\Sasidhar\Desktop>
```

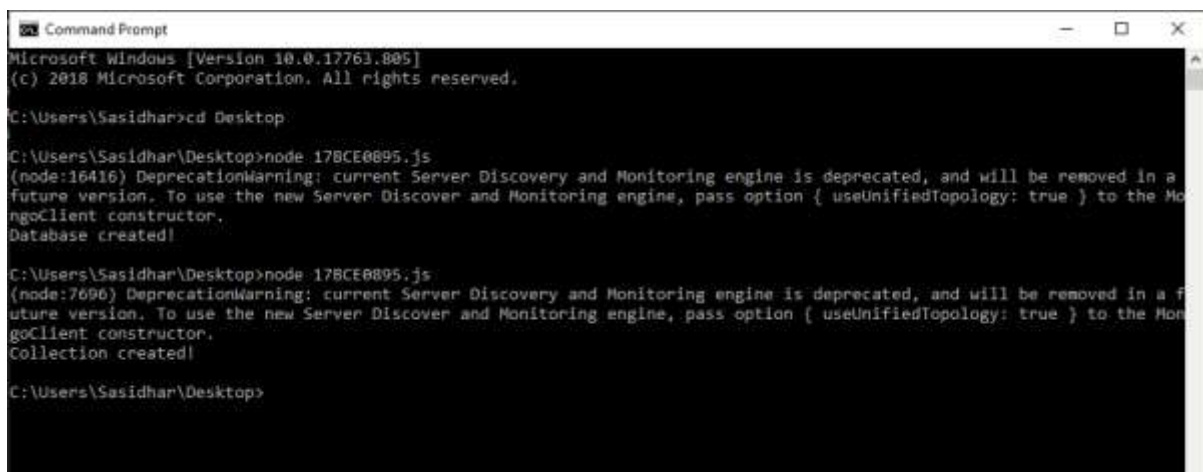
Creating a Collection:

```
var MongoClient = require('mongodb').MongoClient;
```

```
var url = "mongodb://localhost:27017/";
```

```
MongoClient.connect(url, function(err, db) {  
  if (err) throw err;  
  var dbo = db.db("mydb");  
  dbo.createCollection("customers", function(err, res) {  
    if (err) throw err;  
    console.log("Collection created!");  
    db.close();  
  });  
});
```

Screenshot:



```
Command Prompt  
Microsoft Windows [Version 10.0.17763.805]  
(c) 2018 Microsoft Corporation. All rights reserved.  
  
C:\Users\Sasidhar>cd Desktop  
  
C:\Users\Sasidhar\Desktop>node 17BCE0895.js  
(node:16416) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
Database created!  
  
C:\Users\Sasidhar\Desktop>node 17BCE0895.js  
(node:7696) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
Collection created!  
  
C:\Users\Sasidhar\Desktop>
```

Inserting Data:

```
var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

  if (err) throw err;

  var dbo = db.db("mydb");

  var myobj = [

    { name: 'sasi', address: 'Highway 71', phone: '6878487878'},
    { name: 'susanth', address: 'Lowstreet 4', phone: '687864688'},
    { name: 'vikas', address: 'Apple st 652', phone: '8787768578'},
    { name: 'jagan', address: 'Mountain 21', phone: '6878487878'},
    { name: 'Michael', address: 'Valley 345', phone: '5468764878'},
    { name: 'Sandy', address: 'Ocean blvd 2', phone: '7876865487'}

  ];

  dbo.collection("customers").insertMany(myobj, function(err, res) {

    if (err) throw err;


    console.log("Number of documents inserted: " + res.insertedCount);

    db.close();

  });

});
```

Screenshot:



```
Microsoft Windows [Version 10.0.17763.005]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Sasidhar>cd Desktop

C:\Users\Sasidhar\Desktop>node I78CE8895.js
(node:18416) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Database created!

C:\Users\Sasidhar\Desktop>node I78CE8895.js
(node:7896) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Collection created!

C:\Users\Sasidhar\Desktop>node I78CE8895.js
(node:1040) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Number of documents inserted: 6

C:\Users\Sasidhar\Desktop>
```

Displaying Data:

```
var MongoClient = require('mongodb').MongoClient;
```

```
var url = "mongodb://localhost:27017/";
```

```
MongoClient.connect(url, function(err, db) {
```

```
  if (err) throw err;
```

```
  var dbo = db.db("mydb");
```

```
  dbo.collection("customers").find({}).toArray(function(err, result) {
```

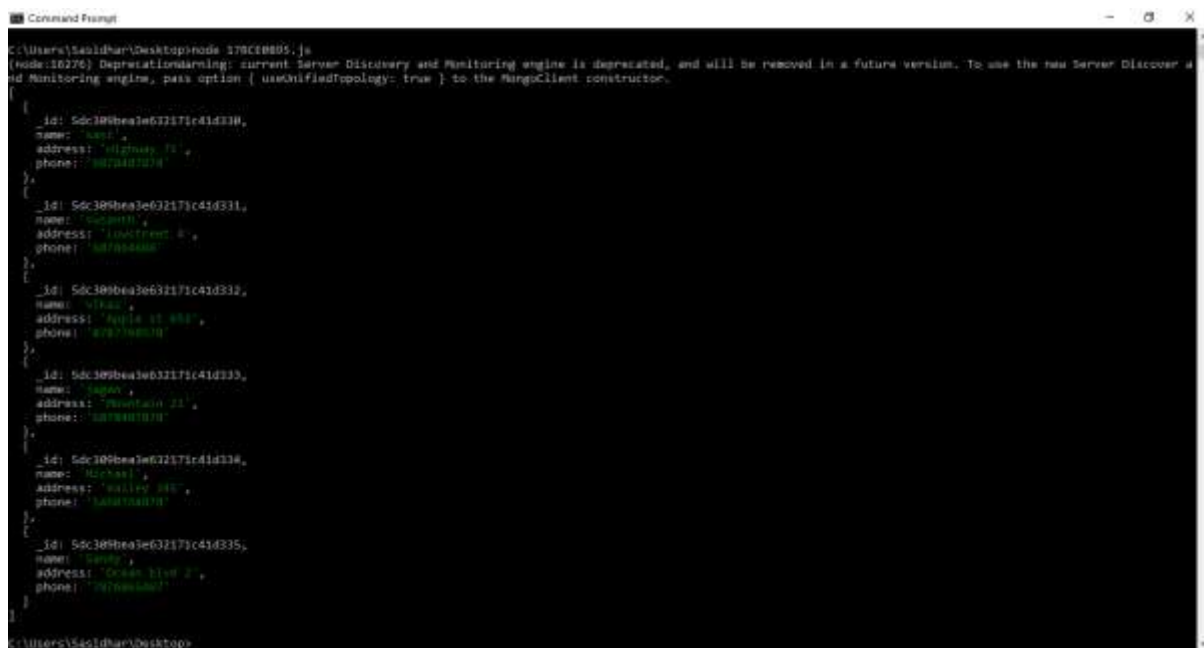
```
    if (err) throw err;
```

```
    console.log(result);
```

```
    db.close();
```

```
  });
```

```
});
```



```
C:\Users\Sasidhar\Desktop>node 179C88805.js
(node:16276) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[
  {
    _id: '5dc389bea3e632171c41d330',
    name: 'keel',
    address: 'Highway 71',
    phone: '9876410274'
  },
  {
    _id: '5dc389bea3e632171c41d331',
    name: 'madison',
    address: 'Lincolnton 8',
    phone: '9876543210'
  },
  {
    _id: '5dc389bea3e632171c41d332',
    name: 'vikki',
    address: 'Appl 15 892',
    phone: '8765432109'
  },
  {
    _id: '5dc389bea3e632171c41d333',
    name: 'sagun',
    address: 'Mountain 21',
    phone: '1234567890'
  },
  {
    _id: '5dc389bea3e632171c41d334',
    name: 'Michael',
    address: 'Valley 345',
    phone: '1234567890'
  },
  {
    _id: '5dc389bea3e632171c41d335',
    name: 'arya',
    address: 'Green Blvd 2',
    phone: '9876543210'
  }
]
C:\Users\Sasidhar\Desktop>
```

Display registered users:

```
app.get('/view', function(req, res){
    user.find({}, function(err, docs){
        if(err) res.json(err);
        else res.render('index', {users: docs});
    });
});
```

To Display User Information:

```
ul
each user in users
li #{user.name}: #{user.age}: #{user._id}
```

Fetch individual user information:

```
app.get('/view', function(req, res){
    user.find({_id: 'kiran@gmail.com'}, function(err, docs){
        if(err) res.json(err);
        else res.render('index', {users: docs});
    });
});
```

Code to fetch the data and displaying in webpage:

```
var express = require('express');
var http = require('http');
var path = require('path');
var mongoose = require('mongoose');

var app = express();

// all environments
app.set('port', process.env.PORT || 3000);
app.set('views', __dirname + '/views');
app.set('view engine', 'jade');
```

```

app.use(express.bodyParser());
app.use(express.methodOverride());
app.use(app.router);
app.use(express.static(path.join(__dirname, 'public')));

mongoose.connect('mongodb://localhost/Company');

var Schema = new mongoose.Schema({
  _id : String,
  name: String,
  age : Number
});

var user = mongoose.model('emp', Schema);

app.get('/view', function(req, res){
  user.find({}, function(err, docs){
    if(err) res.json(err);
    else res.render('index', {users: docs});
  });
});

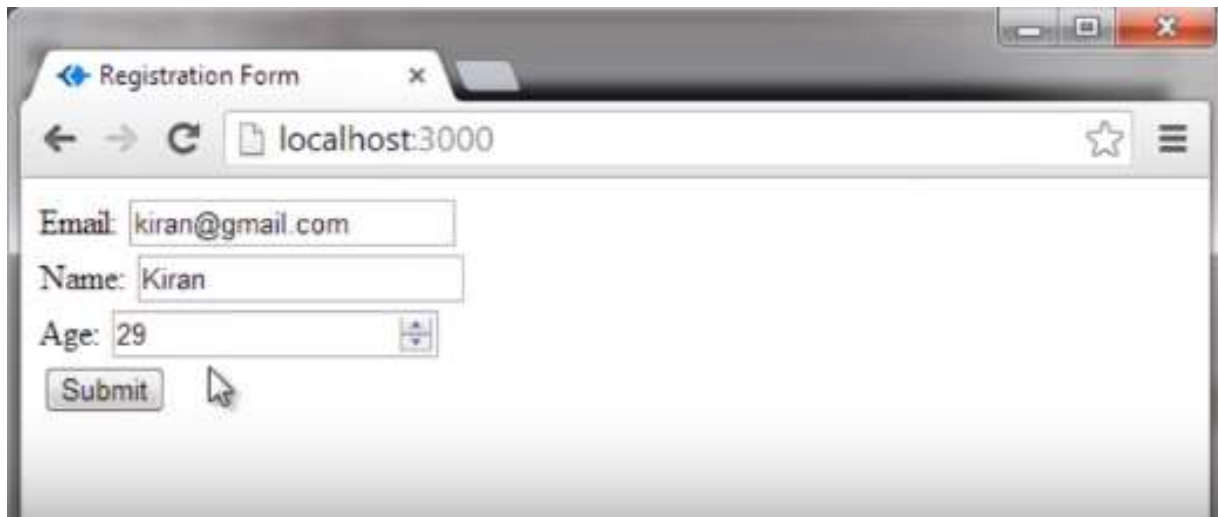
app.post('/new', function(req, res){
  new user({
    _id : req.body.email,
    name: req.body.name,
    age : req.body.age
  }).save(function(err, doc){
    if(err) res.json(err);
    else res.redirect('/view');
  });
});

var server = http.createServer(app).listen(app.get('port'), function(){
  console.log('Express server listening on port ' + app.get('port'));
});

```

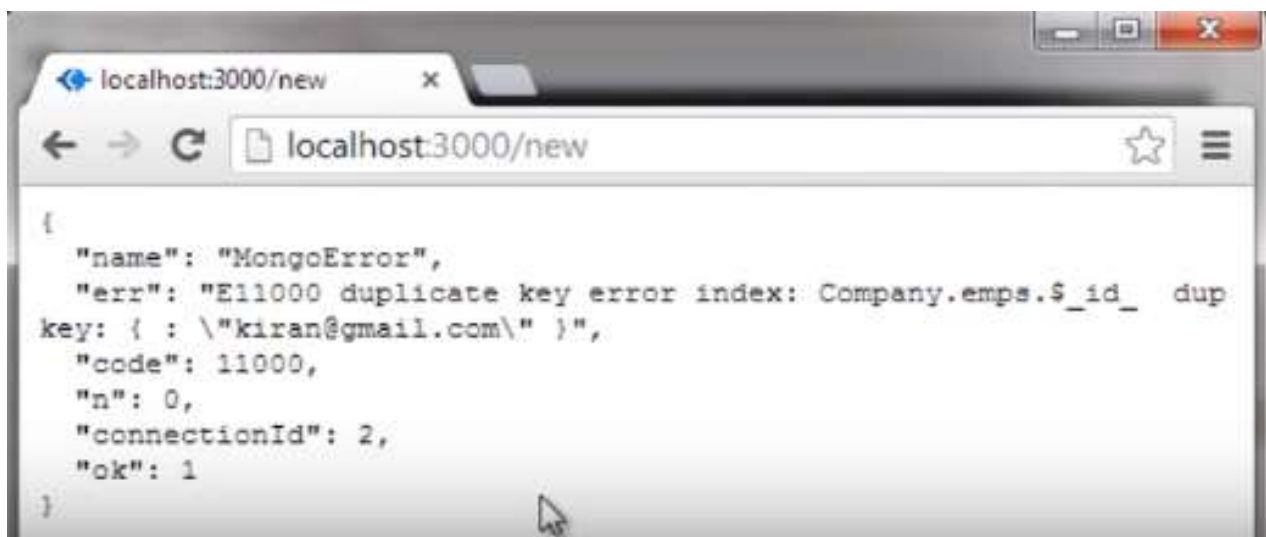
Screenshots:

Trying to register the name which is already present in database:



A screenshot of a web browser window titled "Registration Form". The address bar shows "localhost:3000". The form contains three input fields: "Email" with the value "kiran@gmail.com", "Name" with the value "Kiran", and "Age" with the value "29". Below the fields is a "Submit" button. A mouse cursor is hovering over the "Submit" button.

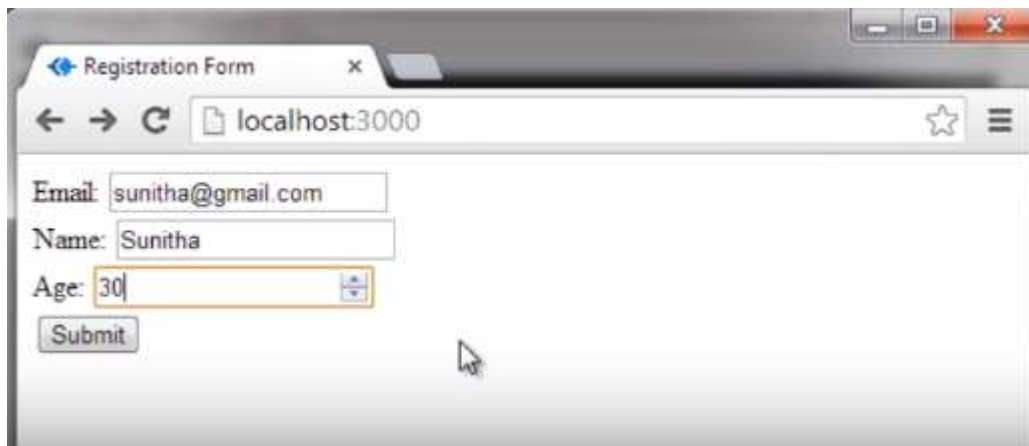
It shows the error:



```
{
  "name": "MongoError",
  "err": "E11000 duplicate key error index: Company.emps.$_id_ dup
key: { : \"kiran@gmail.com\" }",
  "code": 11000,
  "n": 0,
  "connectionId": 2,
  "ok": 1
}
```

A screenshot of a web browser window titled "localhost:3000/new". The address bar shows "localhost:3000/new". The page displays a JSON error response from a MongoDB database. The error message indicates a duplicate key error for the index "Company.emps.\$_id_" with the key value "kiran@gmail.com". The response includes fields for "name", "err", "code", "n", "connectionId", and "ok".

Trying to register with new name:

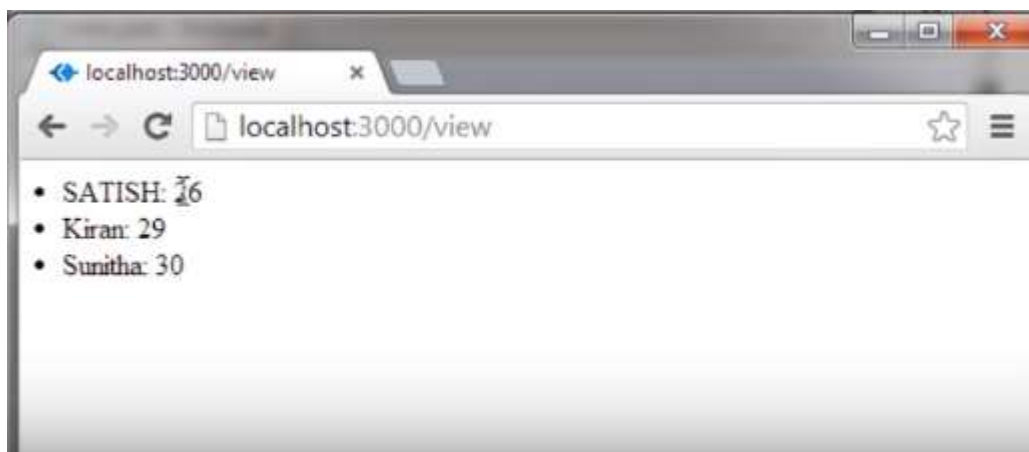


The screenshot shows a web browser window with a single tab titled "Registration Form". The address bar displays "localhost:3000". The page contains a registration form with the following fields and values:

- Email:
- Name:
- Age:

Below the fields is a "Submit" button. A mouse cursor is visible over the button.

Registered successfully and data is fetched:



The screenshot shows a web browser window with a single tab titled "localhost:3000/view". The address bar displays "localhost:3000/view". The page displays a list of registered users:

- SATISH: 26
- Kiran: 29
- Sunitha: 30

Sample Outputs:

Online Food Delivery System



Non Veg



Break Fast



Starters



User Login

Email:

Password:

Login

New Registration