VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DIGITAL ASSINGMENT 1

BOOK COLLECTION WEB APPLICATION

WEB TECHNOLOGIES-ITE1002

Submitted to

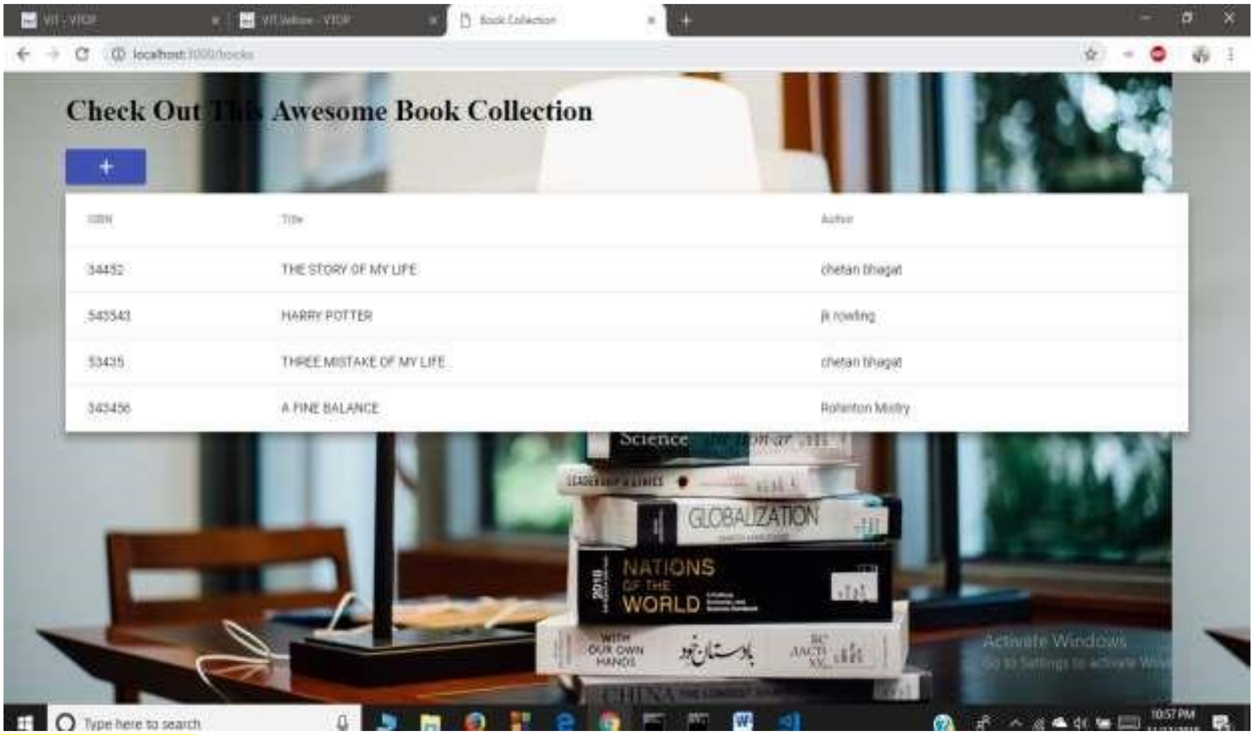
MAREESWARI V

submitted by

AKASH KUMAR(17BIT0055)

Output:

Book collection

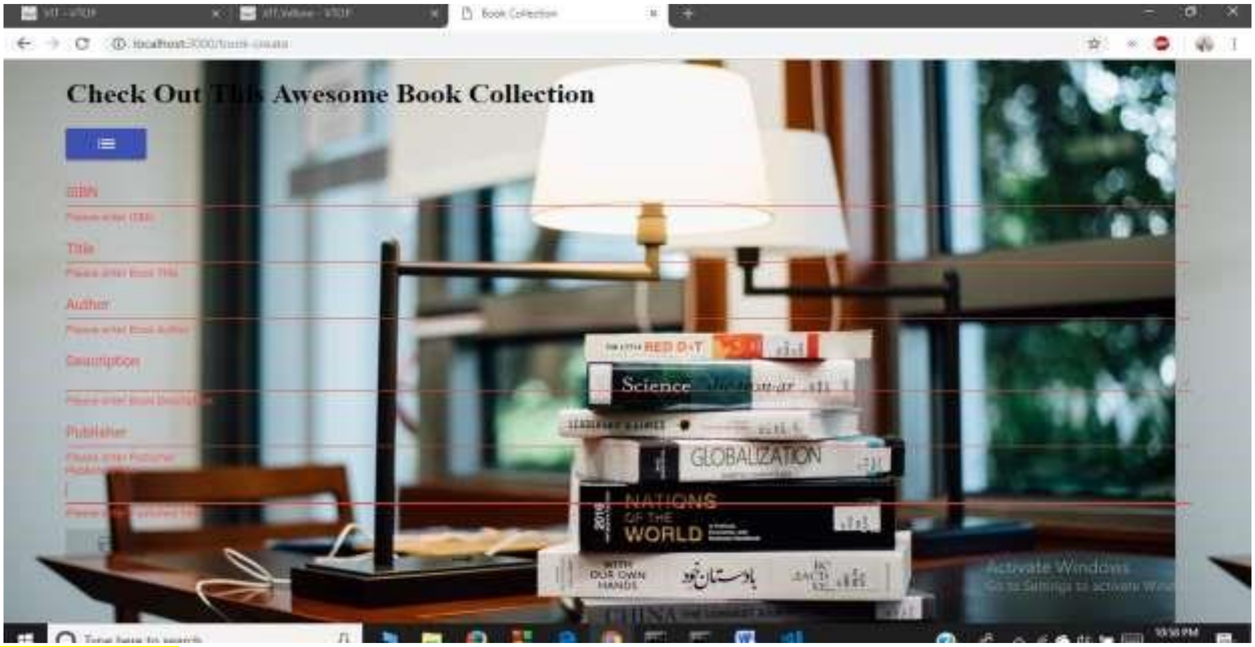


localhost:3000/books

Check Out This Awesome Book Collection

ISBN	Title	Author
34452	THE STORY OF MY LIFE	chetan bhagat
543543	HARRY POTTER	jk rowling
53435	THREE MISTAKE OF MY LIFE	chetan bhagat
343456	A FINE BALANCE	Rohinton Mistry

Book-create



localhost:3000/book-create

Check Out This Awesome Book Collection

ISBN

Please enter ISBN

Title

Please enter Book Title

Author

Please enter Book Author

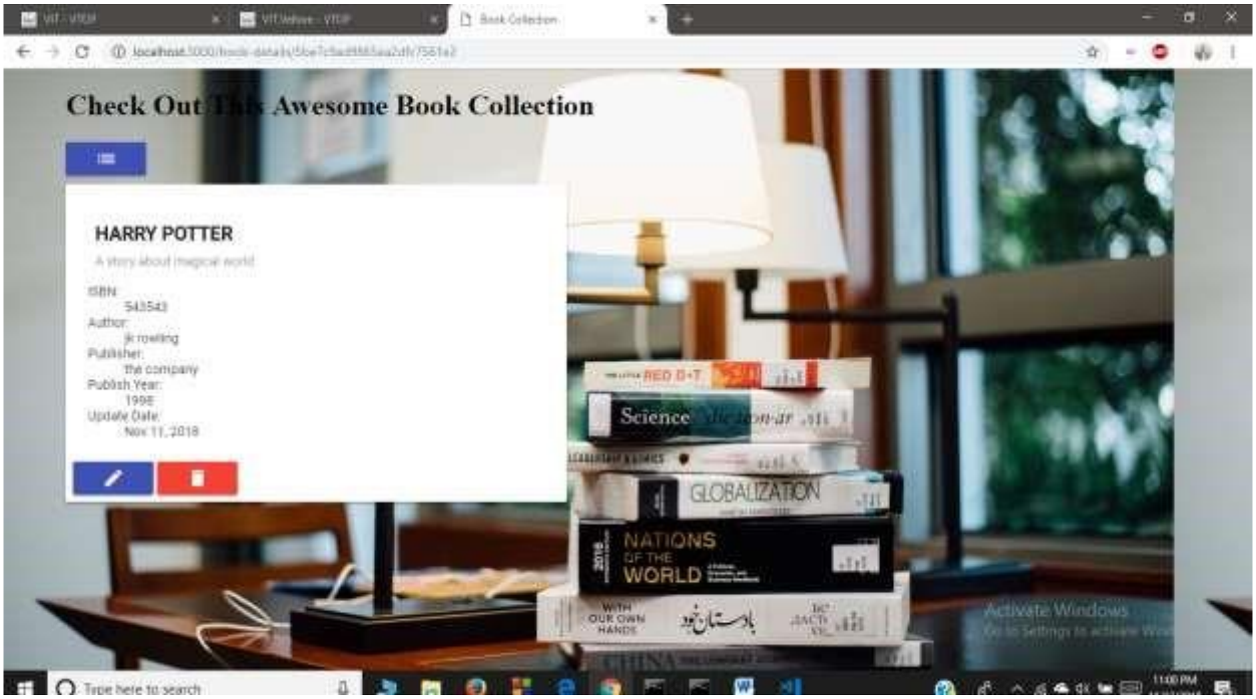
Description

Please enter Book Description

Publisher

Please enter Publisher

Book-details


Book Collection
localhost:3000/book-details/5be7c5ad8885aa2dfc7561a2
Type here to search
Activate Windows
Go to Settings to activate Windows.

Check Out This Awesome Book Collection

HARRY POTTER

A story about magical world

ISBN:
543543
Author:
jk rowling
Publisher:
the company
Publish Year:
1998
Update Date:
Nov 11, 2018

Frontend codes:

Html , css and angular code:

1.Book collection

Html:

```
<div class="button-row">
  <a mat-raised-button color="primary" [routerLink]="['/book-
create']"><maticon>add</mat-icon></a>
</div>
<div class="example-container mat-elevation-z8">
  <table mat-table #table [dataSource]="dataSource">
    <!-- Title Column -->
    <ng-container matColumnDef="isbn">
      <th mat-header-cell *matHeaderCellDef> ISBN </th>
      <td mat-cell *matCellDef="let element" class="isbn-col"> {{element.isbn}} </td>
    </ng-container>

    <!-- Title Column -->
    <ng-container matColumnDef="title">
      <th mat-header-cell *matHeaderCellDef> Title </th>
      <td mat-cell *matCellDef="let element"> {{element.title}} </td>
    </ng-container>

    <!-- Author Column -->
    <ng-container matColumnDef="author">
      <th mat-header-cell *matHeaderCellDef> Author </th>
      <td mat-cell *matCellDef="let element"> {{element.author}} </td>
    </ng-container>

    <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
    <tr mat-row *matRowDef="let row; columns: displayedColumns;"
[routerLink]="['/book-details/', row._id]"></tr>
  </table>
</div>
```

Css:

```
.example-container {
display: flex;   flex-
direction: column;   max-
height: 500px;   min-width:
300px;   overflow: auto;
}

.isbn-col {   flex: 0 0 100px
!important;   white-space:
unset !important;
}

.button-row {
margin: 10px 0;
}
```

2.Book-create

Html:

```
<div class="button-row">
  <a mat-raised-button color="primary" [routerLink]="['/books']"><maticon>list</mat-
icon></a>
</div>
```

```
<form [formGroup]="bookForm" (ngSubmit)="onFormSubmit(bookForm.value)">
  <mat-form-field class="example-full-width">
    <input matInput placeholder="ISBN" formControlName="isbn"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('isbn').valid && bookForm.get('isbn').touched">Please
enter ISBN</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Title" formControlName="title"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('title').valid &&
bookForm.get('title').touched">Please enter Book Title</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Author" formControlName="author"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('author').valid &&
bookForm.get('author').touched">Please enter Book Author</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <textarea matInput placeholder="Description" formControlName="description"
           [errorStateMatcher]="matcher"></textarea>
    <mat-error>
      <span *ngIf="!bookForm.get('description').valid &&
bookForm.get('description').touched">Please enter Book Description</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Publisher" formControlName="publisher"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('publisher').valid &&
bookForm.get('publisher').touched">Please enter Publisher</span>     </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Published Year" formControlName="published_year"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('published_year').valid &&
bookForm.get('published_year').touched">Please enter Published Year</span>
```

```
    </mat-error>
  </mat-form-field>
  <div class="button-row">
    <button type="submit" [disabled]="!bookForm.valid" mat-raised-button
color="primary"><mat-icon>save</mat-icon></button>
  </div>
</form>
```

Css :

```
.example-form {   min-
width: 150px;   max-width:
500px;   width: 100%;
}

.example-full-width {
width: 100%;
}

.example-full-width:nth-last-child() {   margin-
bottom: 10px;
}

.button-row {
margin: 10px 0;
}
```

3.Book detail Html:

```
<div class="button-row">
```

```
  <a mat-raised-button color="primary"
[routerLink]="['/books']"><maticon>list</mat-icon></a>
</div>
<mat-card class="example-card">
  <mat-card-header>
    <mat-card-title><h2>{{book.title}}</h2></mat-card-title>
    <mat-card-subtitle>{{book.description}}</mat-card-subtitle>
  </mat-card-header>
  <mat-card-content>
<dl>
      <dt>ISBN:</dt>
      <dd>{{book.isbn}}</dd>
      <dt>Author:</dt>
      <dd>{{book.author}}</dd>
      <dt>Publisher:</dt>
      <dd>{{book.publisher}}</dd>
      <dt>Publish Year:</dt>
      <dd>{{book.published_year}}</dd>
      <dt>Update Date:</dt>
      <dd>{{book.updated_date | date}}</dd>
    </dl>
  </mat-card-content>
  <mat-card-actions>
    <a mat-raised-button color="primary" [routerLink]="['/book-edit',
book._id]"><mat-icon>edit</mat-icon></a>
    <a mat-raised-button color="warn"
(click)="deleteBook(book._id)"><maticon>delete</mat-icon></a>
  </mat-card-actions>
</mat-card>
```

Css:

```
.example-card {   max-width: 500px;
}
 .button-row {   margin: 10px 0; }
```

4.Book edit Html:

```
<div class="button-row">
  <a mat-raised-button color="primary" (click)="bookDetails()"><maticon>show</mat-
icon></a>
</div>
<form [formGroup]="bookForm" (ngSubmit)="onFormSubmit(bookForm.value)">
```

```
  <mat-form-field class="example-full-width">
    <input matInput placeholder="ISBN" formControlName="isbn"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('isbn').valid && bookForm.get('isbn').touched">Please
enter ISBN</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Title" formControlName="title"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('title').valid &&
bookForm.get('title').touched">Please enter Book Title</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Author" formControlName="author"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('author').valid &&
bookForm.get('author').touched">Please enter Book Author</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <textarea matInput placeholder="Description" formControlName="description"
           [errorStateMatcher]="matcher"></textarea>
    <mat-error>
      <span *ngIf="!bookForm.get('description').valid &&
bookForm.get('description').touched">Please enter Book Description</span>
    </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Publisher" formControlName="publisher"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('publisher').valid &&
bookForm.get('publisher').touched">Please enter Publisher</span>     </mat-error>
  </mat-form-field>
  <mat-form-field class="example-full-width">
    <input matInput placeholder="Published Year" formControlName="published_year"
           [errorStateMatcher]="matcher">
    <mat-error>
      <span *ngIf="!bookForm.get('published_year').valid &&
bookForm.get('published_year').touched">Please enter Published Year</span>
    </mat-error>
  </mat-form-field>
  <div class="button-row">
    <button type="submit" [disabled]="!bookForm.valid" mat-raised-button
color="primary"><mat-icon>save</mat-icon></button>
  </div>
</form>
```

Css:

```
.example-form {   min-
width: 150px;   max-width:
500px;   width: 100%;
}

.example-full-width {
width: 100%;
}

.example-full-width:nth-last-child() {   margin-
bottom: 10px;
}

.button-row {
margin: 10px 0;
}
```

5.main Index .html : link all page togetter

```
<!DOCTYPE html><html lang="en"><head>
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500"
rel="stylesheet">
  <meta charset="utf-8">
  <title>Book Collection</title>
  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="books.png">
```

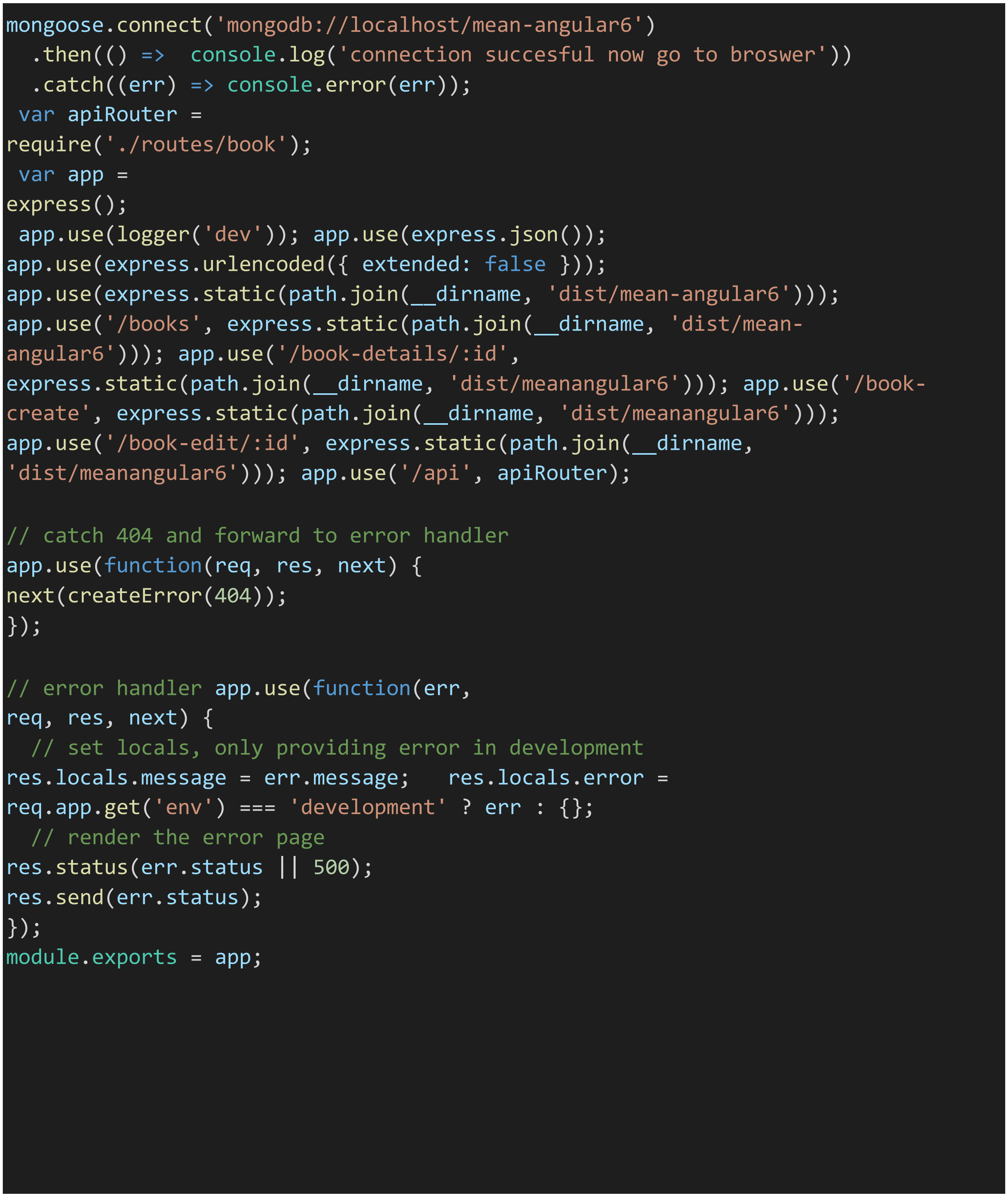
```
</head>
<body style="background-image: url('wallpaper.jpg');">
  <h1>Check Out This Awesome Book Collection</h1>
  <app-root></app-root>


<script type="text/javascript" src="runtime.js"></script><script
type="text/javascript" src="polyfills.js"></script><script type="text/javascript"
src="styles.js"></script><script type="text/javascript"
src="vendor.js"></script><script type="text/javascript" src="main.js">
</script>
</body>
</html>
```

BACKEND CODE:

Node js code:

1.App.js

```
var createError = require('http-
errors'); var express =
require('express'); var path =
require('path'); var favicon =
require('serve-favicon'); var logger =
require('morgan');
 var mongoose =
require('mongoose');
```



```
mongoose.connect('mongodb://localhost/mean-angular6')
  .then(() =>  console.log('connection succesful now go to broswer'))
  .catch((err) => console.error(err));
 var apiRouter =
require('./routes/book');
 var app =
express();
 app.use(logger('dev')); app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(express.static(path.join(__dirname, 'dist/mean-angular6')));
app.use('/books', express.static(path.join(__dirname, 'dist/mean-
angular6'))); app.use('/book-details/:id',
express.static(path.join(__dirname, 'dist/meanangular6'))); app.use('/book-
create', express.static(path.join(__dirname, 'dist/meanangular6')));
app.use('/book-edit/:id', express.static(path.join(__dirname,
'dist/meanangular6'))); app.use('/api', apiRouter);

// catch 404 and forward to error handler
app.use(function(req, res, next) {
next(createError(404));
});

// error handler app.use(function(err,
req, res, next) {
  // set locals, only providing error in development
res.locals.message = err.message;   res.locals.error =
req.app.get('env') === 'development' ? err : {};
  // render the error page
res.status(err.status || 500);
res.send(err.status);
});
module.exports = app;
```

2. route.js

```
var express = require('express'); var
router = express.Router(); var mongoose
= require('mongoose'); var Book =
require('../models/Book.js');

/* GET ALL BOOKS */
router.get('/', function(req, res, next)
{   Book.find(function (err, products) {
if (err) return next(err);
res.json(products);
  });
});

/* GET SINGLE BOOK BY ID */
router.get('/:id', function(req, res, next) {
  Book.findById(req.params.id, function (err, post)
{     if (err) return next(err);     res.json(post);
  });
});

/* SAVE BOOK */
router.post('/', function(req, res, next) {
Book.create(req.body, function (err, post) {
if (err) return next(err);
res.json(post);
  });
});

/* UPDATE BOOK */
router.put('/:id', function(req, res, next) {
  Book.findByIdAndUpdate(req.params.id, req.body, function (err, post)
{     if (err) return next(err);     res.json(post);
  });
});

/* DELETE BOOK */
router.delete('/:id', function(req, res, next) {
  Book.findByIdAndRemove(req.params.id, req.body, function (err, post)
{     if (err) return next(err);     res.json(post);



  });
});
module.exports = router;
```

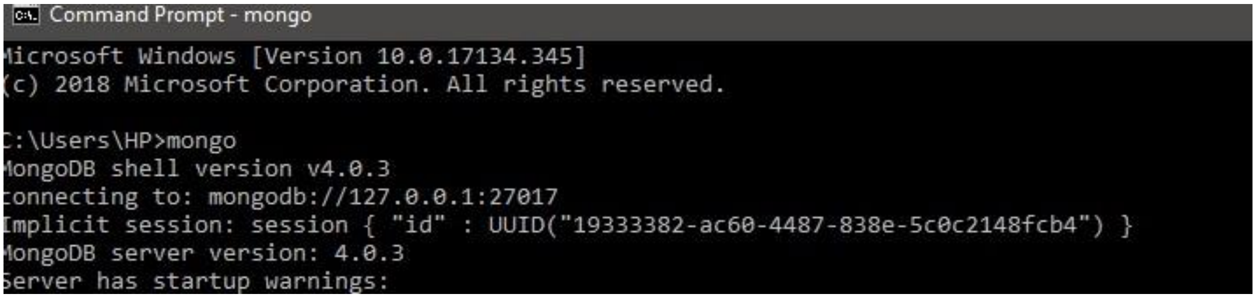
Mongo db code:

```
var mongoose = require('mongoose');
 var BookSchema = new mongoose.Schema({   isbn:
String,   title: String,   author: String,
description: String,   published_year: String,
publisher: String,   updated_date: { type: Date,
default: Date.now },
});  module.exports = mongoose.model('Book',
BookSchema);
```

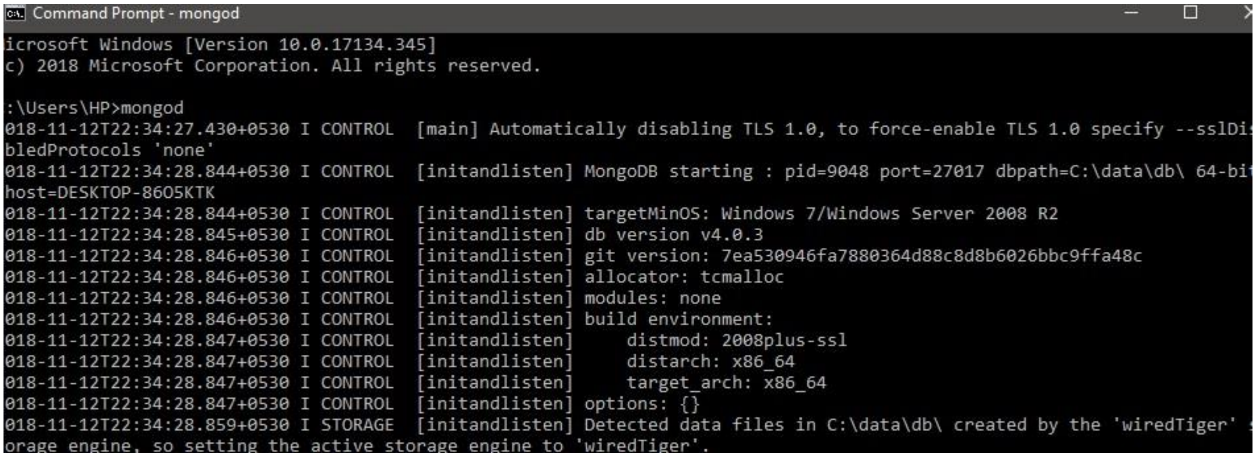
```
var mongoose = require('mongoose'); mongoose.connect('mongodb://localhost/mean-
angular6')
  .then(() =>  console.log('connection succesful now go to broswer'))
  .catch((err) => console.error(err));
```

Command promt codes:

1. For starting mongodb server:



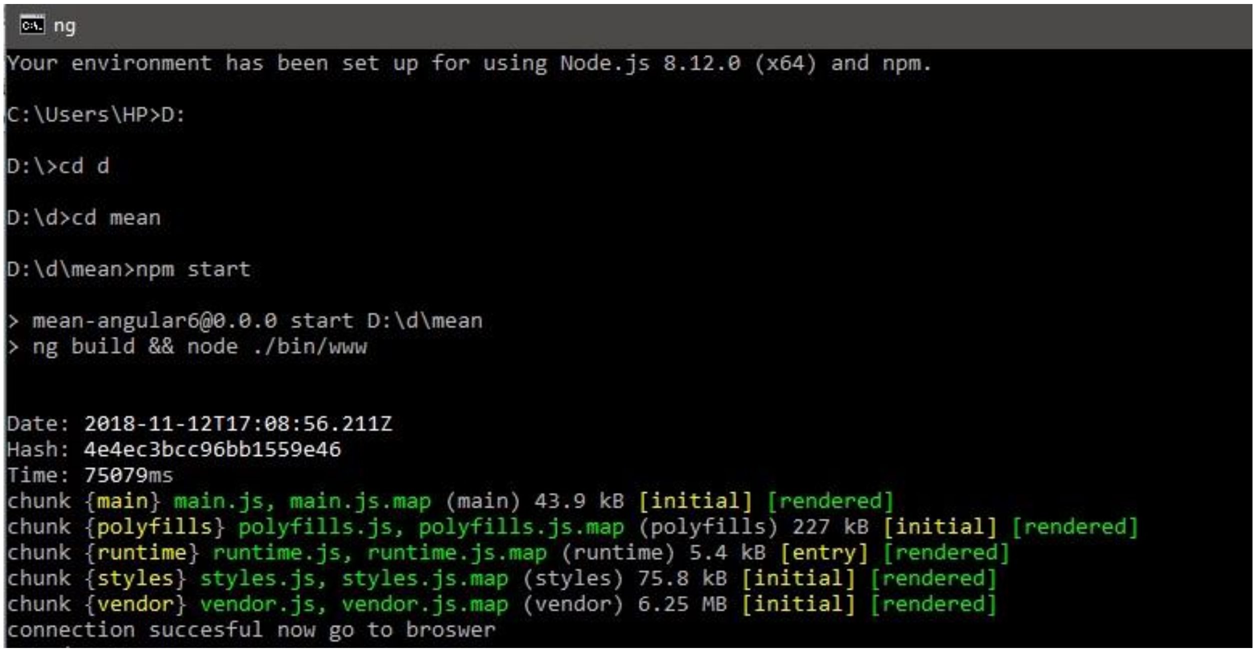
```
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\HP>mongo
MongoDB shell version v4.0.3
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("19333382-ac60-4487-838e-5c0c2148fcb4") }
MongoDB server version: 4.0.3
Server has startup warnings:
```



```
icrosoft Windows [Version 10.0.17134.345]
c) 2018 Microsoft Corporation. All rights reserved.

:\Users\HP>mongod
018-11-12T22:34:27.430+0530 I CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDi
bledProtocols 'none'
018-11-12T22:34:28.844+0530 I CONTROL  [initandlisten] MongoDB starting : pid=9048 port=27017 dbpath=C:\data\db\ 64-bi
host=DESKTOP-8605KTK
018-11-12T22:34:28.844+0530 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
018-11-12T22:34:28.845+0530 I CONTROL  [initandlisten] db version v4.0.3
018-11-12T22:34:28.846+0530 I CONTROL  [initandlisten] git version: 7ea530946fa7880364d88c8d8b6026bbc9ffa48c
018-11-12T22:34:28.846+0530 I CONTROL  [initandlisten] allocator: tcmalloc
018-11-12T22:34:28.846+0530 I CONTROL  [initandlisten] modules: none
018-11-12T22:34:28.846+0530 I CONTROL  [initandlisten] build environment:
018-11-12T22:34:28.847+0530 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
018-11-12T22:34:28.847+0530 I CONTROL  [initandlisten]     distarch: x86_64
018-11-12T22:34:28.847+0530 I CONTROL  [initandlisten]     target_arch: x86_64
018-11-12T22:34:28.847+0530 I CONTROL  [initandlisten] options: {}
018-11-12T22:34:28.859+0530 I STORAGE  [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' 
orage engine, so setting the active storage engine to 'wiredTiger'.
```

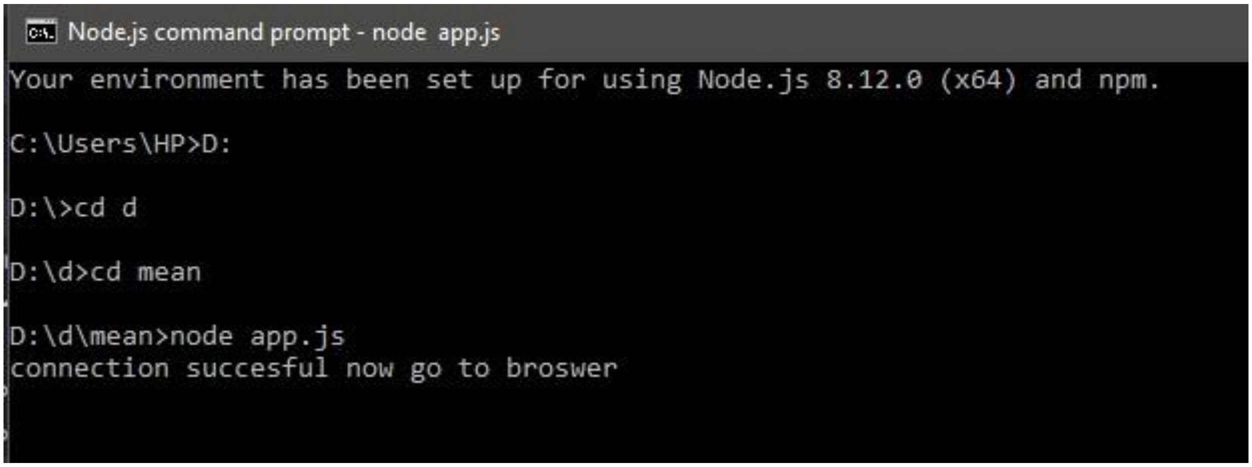
2. Code to start nodejs server

To start forntend



```
Your environment has been set up for using Node.js 8.12.0 (x64) and npm.

C:\Users\HP>D:

D:\>cd d

D:\d>cd mean

D:\d\mean>npm start

> mean-angular6@0.0.0 start D:\d\mean
> ng build && node ./bin/www


Date: 2018-11-12T17:08:56.211Z
Hash: 4e4ec3bcc96bb1559e46
Time: 75079ms
chunk {main} main.js, main.js.map (main) 43.9 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 227 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 5.4 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 75.8 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 6.25 MB [initial] [rendered]
connection succesful now go to broswer
```

To start backend server



```
Your environment has been set up for using Node.js 8.12.0 (x64) and npm.

C:\Users\HP>D:

D:\>cd d

D:\d>cd mean

D:\d\mean>node app.js
connection succesful now go to broswer
```