* **Create Project**:

> npx express-generator project3-library --view=pug

> cd project3-library

> npm install

> npm install --save-dev nodemon

* **Mongoose Setup**:

> npm install mongoose

**Add to app.js**:

//Set up mongoose connection

var mongoose = require('mongoose');

var mongoDB = 'insert\_your\_database\_url\_here';

mongoose.connect(mongoDB, { useNewUrlParser: true, useUnifiedTopology: true});

var db = mongoose.connection;

db.on('error', console.error.bind(console, 'MongoDB connection error:'));

**MongoDB URL Example**:

mongodb+srv://[namyun:061301noho@cluster0.ljboq.mongodb.net/local\_library?retryWrites=true&w=majority](http://namyun:061301noho@cluster0.ljboq.mongodb.net/local_library?retryWrites=true&w=majority)

* **Models**:

Graphical user interface, text

Description automatically generated

* **Populate Database**:

Save populatedb.js to project3-library

> npm install async

> node populatedb <your mongodb url>

**MongoDB Terminal URL Example**:

"mongodb+srv://[namyun:061301noho@cluster0.ljboq.mongodb.net/local\_library?retryWrites=true](http://namyun:061301noho@cluster0.ljboq.mongodb.net/local_library?retryWrites=true&w=majority)"

* **Controllers**:

Text

Description automatically generated

* **Routes**:

Text

Description automatically generated with medium confidence

* **Update app.js**:

var indexRouter = require('./routes/index');

var usersRouter = require('./routes/users');

var catalogRouter = require('./routes/catalog');

…

app.use('/', indexRouter);

app.use('/users', usersRouter);

app.use('/catalog', catalogRouter);

Views

* **Replace /views/layout.pug**

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Displaying_data/LocalLibrary_base_template>

* **Replace index() function in /controllers/bookController.js**

**Replace /views/index.pug**

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Displaying_data/Home_page>

* **Replace book\_list() function in /controllers/bookController.js**

**Replace /views/book\_list.pug**

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Displaying_data/Book_list_page>

* **Replace bookinstance\_list() function in /controllers/bookinstanceController.js**

**Replace /views/bookinstance\_list.pug**

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Displaying_data/BookInstance_list_page>

* **Date formatting using luxon**:

> npm install luxon

**Add in /models/bookinstance.js**:

const { DateTime } = require('luxon');

BookInstanceSchema

.virtual('due\_back\_formatted')

.get(function () {

return DateTime.fromJSDate(this.due\_back).toLocaleString(DateTime.DATE\_MED);

});

**Replace due\_back with due\_back\_formatted in /views/bookinstance\_list.pug**:

if val.status != 'Available'

//span (Due: #{val.due\_back})

span (Due: #{val.due\_back\_formatted})

* **Further instructions**:

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Displaying_data>

Deployment

* **Compression**: reduce the time required for the client to get and load the page.

> npm install compression

**Add code to app.js**:

var compression = require('compression');

…

app.use(compression()); // Compress all routes

* **Helmet**: protect app from well-known web vulnerabilities

> npm install helmet

**Add code to app.js**:

var helmet = require('helmet');

…

app.use(helmet());