

Step-1=>Install following packages through npm-

>bcryptjs,dotenv,ejs,express,fs,jsonwebtoken,mongoose,multer,nodemon,path

Step-2=>Create folders config,controller,models,routes,uploads,views

Step-3=>create **.js** files server.js, jwtconfig.js, config(folder)->db.js, model(folder)->book.js

and user.js(for login and registration),controller(folder)->controller.js,router(folder)-

>router.js, views(folder)->add.ejs,edit.ejs,list.ejs,login.ejs,registration.ejs

Step-4=>in config(folder)->db.js file write connection code

Step-5=>in model(folder)->book.js and user.js(for login and registration) file create your schema for given table and user

Step-6=>in controller(folder)->controller.js file create 7 functions create, edit, delete, update, list, login, registration under login function jwt is written as below

First declare 3 const like below then followed code

```
const config = require('../jwtconfig');
const bcrypt = require('bcryptjs');
const jwt = require('jsonwebtoken');

login : async(req,res)=>{
  try{
    user.findOne({ email: req.body.email }, (err, user) => {
      if (user) {
        let validPassword =
bcrypt.compare(req.body.pass,user.password,(err, isMatch) => {
          if (isMatch) {
            let token = jwt.sign({ userId: user._id },
              config.secret, { expiresIn: '24h' });
            //saving token to the cookie for the user and
redirecting to student page
            res.cookie('token', token, {
              maxAge: 900000,
              httpOnly: true
            });
            res.redirect('/books/list');
            //res.json({ success: true, message:
'Success!',token: token,user: { email: user.email } });
          } else {
            res.json({ success: false, message: 'Invalid
password' });
          }
        });
      }
    });
  }
}
```

```

    }
  });
}
catch(error){
  console.log(error.message);
}
}

```

Step-7=>in router.js file set routers for function in ejs files and code for multer for pic uploads as below

```

require('fs');
const multer = require("multer");
const path = require("path");

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    // console.log(file);
    cb(null, path.join(__dirname, "../uploads"));
  },
  filename: function (req, file, cb) {
    if (file.mimetype === "image/jpg" || file.mimetype === "image/png" ||
file.mimetype === "image/jpeg")
      cb(null, `${file.originalname}`);
    else
      cb(new Error("Invalid file format"));
  }
});
const multerUpload = multer({ storage: storage });

```

Step-8=>in jwtconfig.js file export 2 variables database and secret as below

```

module.exports={
  database: 'mongodb://localhost:27017/testdb',
  secret: 'secret'
}

```

Step-9=>in server.js file set listening port and url for routers