Version 5

authDemo

Version 5 mei original changes

So now hum user ko picture mei lekar aaenge using a tool **Passport.** **(for authentication)**

hum bcrypt naam ka hashing fucntion use krenge.

go to **passport.js** (only for nodejs)

show **documentation** —> **authenticate**

1. hum **local strategy** use krne vaale hai (strategy)
2. **passport** is a tool (using tool)

strategies mei jaao index ke andar and make them see the content.\

task 58:

Ab create a schema and yaad rakho schema ke andar password directly store nhi ho sakte we need to hash the password pehle using a package namely **passport-local-mongoose** hashing function.

💡 ab **passport-local-mongoose** ka kaam hai jab aap signup krte ho and aapki detials like username and password store krte ho DB mei to ye **PLM** usse encrypt krke store krvaega bcrypt bhi krdeta hai same lekin **PLM** easily krdega for us i.e seedha hash krdega but one thing to keep in mind is ki ye username aur password kaun add krega in my schema to ye **PLM** krdega so hume iske liye schema banane ki need nhi hai.

task 59: hume **PLM** ko install krke plugin bhi add krne hote hai (documentation in npm)

taaki **PLM** kisaari power ko use kr sake.

```
//terminal
npm i passport-local-mongoose




//models --> user.js
const mongoose = require('mongoose');
const passportLocalMongoose = require('passport-local-mongoose'); //add later
```

Samarth Vohra

```
const userSchema = new mongoose.Schema({
    email: {
        type: String,
        trim: true,
        required: true
    }
});

userSchema.plugin(passportLocalMongoose); //add later


const User = mongoose.model('User', userSchema);

module.exports = User;
```

task 60: ab new route banana hai for user i.e **auth.js** inside routes folder.

and use **app.js mei require krlo**.

```
//routes--> auth.js
const express = require('express');
const router = express.Router();


module.exports = router;




//app.js file

const productRoutes = require('./routes/product');
const reviewRoutes = require('./routes/review');
const authRoutes = require('./routes/auth');       //added


app.use(productRoutes);
app.use(reviewRoutes);
app.use(authRoutes);       //added
```

task 61:

ab documentation mei **static methods** dikhao and **register** dikhao.

static methods directly aapke schema par apply hojaae hai chahe unhe aap banao ya koi aur.

```
//auth.js
//isme hum password alag se dedenge not variable ke saath mei

const express = require('express');
const router = express.Router();
const User = require('../models/user');


router.get('/fakeuser', async(req, res) => {
```

Samarth Vohra

```
    const user = {
        email: 'samarth@gmail.com',
        username:'samarth'
    }
    const newUser = await User.register(user, 'sam123');

    res.send(newUser);
});



module.exports = router;
```

—> Ab hum db mei dekh sakte hai humare db mei ki user aaya hai ya nhi jo aapka PLM automatically kr rha hai sab and 2 fields bhi khud se dera hai.

IMPORTANT: COMMENT KRDIA HAI BACCHO FAKEUSER KO TO NOTES AACHE SE DEKHO

task 62:

ab register() to **PLM** ne krdia ab jab hum login form bharenge to humara **passport** kya krega seedha user jo register ho rakhe hai unke andar se authenticate() method ki madad se behind the scene match krva kr **login** krva dega.

passport.js ke docs mei jaakr **config** dikhao and batao ki it accepts a local strategy ka function and simply authenticate the user.

—> hum khud ka bhi function add krsakte hai just like in passport ka scene but PLM hume derha hai to we wil simply use the method **authentcate().**

```
//terminal
npm i passport  passport-local



//app.js
const passport =  require('passport'); //added all
const LocalStrategy =  require('passport-local');
const User = require('./models/user');


...code...

app.use(session(sessionConfig));
app.use(flash());


passport.use(new LocalStrategy( //added from here

));
```

Samarth Vohra

ab hume middleware set krna hai after **authentcate (jo passport ko bola gya hai to use)**

ab we want persistant login to uske liye we have **passport.initialize()** middleware. and to store it we have **passport.session().**

```
//app.js
app.use(flash());

app.use(passport.initialize()); //added
app.use(passport.session());  //added
```

task 62: ab hume session ka kaam krna hai to for that we need to make use of **serialize** the user and **deserialize** the user.

```
//app.js

app.use(session(sessionConfig));
app.use(flash());

app.use(passport.initialize());
app.use(passport.session());
passport.serializeUser(User.serializeUser()); //added
passport.deserializeUser(User.deserializeUser()); //added



passport.use(new LocalStrategy(User.authenticate()));
```

ab humara kaam sirf pages banane ka hai and unke routes banane hai.

task 63:

ek route hit kro to get the **signup** form.

ek folder banao **auth** and simply uske andar file banao signup.js

```
//auth.js

router.get('/register' , (req,res)=>{
    res.render('auth/signup');
})



//views --> auth --> signup.js
<% layout('layouts/boilerplate') -%>


<div class="row">
    <div class="col-md-6 mx-auto">
        <%- include('../partials/flash')  %>
```

Samarth Vohra

```
        <h1 class="display-6">Sign Up</h1>
        <form action="/register" method="post">
            <div class="mb-3">
                <label class="form-label" for="username">Username</label>
                <input class="form-control" type="text" name="username" id="username" placeholder="Username">
            </div>
            <div class="mb-3">
                <label class="form-label" for="email">Email</label>
                <input class="form-control" type="email" name="email" id="email" placeholder="Email">
            </div>
            <div class="mb-3">
                <label class="form-label" for="password">Password</label>
                <input class="form-control" type="password" name="password" id="password" placeholder="Password">
            </div>
            <button type="submit" class="btn btn-success btn-small">SignUp</button>
        </form>
    </div>
</div>
```

task 64: ab hume signup krna hai actually i.e data ko macth krvana hai to post request bhejunga.

taaki us data ke through mai user bana aku to simply i would need to make use of a class to make that user

—> form ka methods **POST** and route **/register**.

```
//auth.js

//form adding
router.get('/register' , (req,res)=>{
    res.render('auth/signup');
})


//actually adding
router.post('/register' , async(req,res)=>{
    let {email,username,password} = req.body;
    const user = new User({email,username});
    const newUser = await User.register(user,password);
    res.send(newUser);
})
```

Ab hum check kr sakte hai ki hum user create kar paa rhe hai.

task 65:

ab hume login form banana hai and simply usse match krvana hai data.

make a file login.ejs in auth folder

```
//views --> auth --> login.ejs

//same as signup bas email hta do
```

Samarth Vohra

```
<% layout('layouts/boilerplate') -%>


<div class="row">
    <div class="col-md-5 mx-auto">
        <%- include('../partials/flash')  %>
        <div class="card p-3 shadow-sm m-2">
            <h1 class="display-6">Login</h1>
            <form action="/login" method="post">
                <div class="mb-3">
                    <label class="form-label" for="username">Username</label>
                    <input class="form-control" type="text" name="username" id="username" placeholder="Username">
                </div>
                <div class="mb-3">
                    <label class="form-label" for="password">Password</label>
                    <input class="form-control" type="password" name="password" id="password" placeholder="Password">
                </div>
                <button type="submit" class="btn btn-success btn-block btn-small">Login</button>
                <p class="fw-light mt-2">Don't have an account yet ? <a href="/register">SignUp</a> </p>
            </form>
        </div>
    </div>
</div>
```

```
//auth.js

//route to get login form
router.get('/login' , (req,res)=>{
    res.render('auth/login');
})
```

task 66: ab sabse important kaam ki user ko login krvana hai.

ab jab post request jaegi mera passport app.js file ke andar user.authenticate() naam ka middleware chalaega and aap usme simply authenticate krva dega aapke user ko.

```
//auth.js

router.get('/login' , (req,res)=>{
    res.render('auth/login');
})

//to check in the authentication waala system when ur authenticate
//will go to app.js and hit it.
//passport.js se copy
router.post('/login',
    passport.authenticate('local', {
       failureRedirect: '/login',
       failureMessage: true ,
    }),
    function(req, res) {
        req.flash('success' , 'welcome back')
        res.redirect('/products');
    }
);
```

Samarth Vohra

```
//passport ko auth.js mei bhi require kro
const User = require('../models/user');
const passport = require('passport');      //added
```

Ab hum login kar paa rhe hai we can see it by simply redirecting it to /products and see all the products.

task 67: navbar ke andar login and logout mei path dedo.

```
//navbar.ejs
<div class="navbar-nav ml-auto">
          <a href="/login" class="nav-link">Login</a>
          <a href="/logout" class="nav-link"><i class="fas fa-shopping-cart"></i>
            <sup><span class="badge bg-danger">2</span></sup>
          </a>
          <a href="#" class="nav-link">Logout</a>
 </div>
```

task 68: ab we need to logout the user.

```
//auth.js

//logout session
router.get('/logout',(req,res)=>{
    ()=>{
        req.logout();
    }
    req.flash('success' , 'goodbye friend');
    res.redirect('/products');
});
```

jab bhi aap authenticate hoge to ek property apne aap automatically aapke req ke andar user add krdega i.e **req.user.**

to aap console mei dekh sakte ho and simply uske naam se print kr sakte ho.

```
//auth.js

router.post('/login',
    passport.authenticate('local', {
       failureRedirect: '/login',
       failureMessage: true ,
    }),
    function(req, res) {
        // console.log(req.user); //adding
        req.flash('success' , `welcome back ${req.user.username}`). //adding
        res.redirect('/products');
```

Samarth Vohra

```
    }
);
```

task 69: ab we want ki jo ye user hai req.user() ke andar, to usme hum access kar paae to hum usse simply bhej dete hai in middleware in app.js

```
//app.js
app.use((req, res, next) => {
    res.locals.currentUser = req.user;      //added new
    res.locals.success = req.flash('success');
    res.locals.error = req.flash('error');
    next();
})
```

ab mai simply check lga sakta hu ki agar user login hai to usse signup na dikhau & agar nhi hai to login dikhau.

```
//navbar.ejs

<div class="navbar-nav ml-auto">

        <%if(!currentUser){%>      //added from here
          <a href="/login" class="nav-link">Login</a>
        <%}else{%>
          <a href="#" class="nav-link"><i class="fas fa-shopping-cart"></i>
            <sup><span class="badge bg-danger">2</span></sup>
          </a>
          <a href="/logout" class="nav-link">Logout</a>
        <%}%>


</div>
```

task 70: ab hum simply add krna chahate hai ek middleware kyu? taaki mai simply uske andar ek check lga saku ki kya band aloggedin hai agar loggedin nahi hai to hum simply login kre bina koi page open nhi krne denge and agar login hua to hum usme saare pages access krne denge,

```
//middleware.js

const isLoggedIn = (req,res,next)=>{
    if(!req.isAuthenticated()){
        req.flash('error' , 'you need to login first');
        return res.redirect('/login');
    }
    next();
}
...

module.exports = {validateProduct ,validateReview , isLoggedIn} ; //changed
```

ab hume isse routes mei use bhi krna hai to simply isse aap product.js vaale mei use kro.

Samarth Vohra

```
//product.js

const { validateProduct , isLoggedIn } = require('../middleware'); //chnged

//ab sab routes mei add krdo middleware
```

task 71: ab i want ki login functionality mera signup form fill krne ke baad chl jaae.

ab req ke andar login ka access bhi rehta hai so for that we can use the following function which is in passport.

```
//auth.js
router.post('/register' , async(req,res)=>{
    try{
        let {email,username,password} = req.body;
        const user = new User({email,username});
        const newUser = await User.register(user,password);
        // res.send(newUser);
        req.login(newUser, function(err) {        //addig below
            if (err) { return next(err); }
            req.flash('success' , 'welcome, you are registered successfully')
            return res.redirect('/products');
        });
    }
    catch(e){
        req.flash('error' , e.message);
        return res.redirect('/products');
    }
})
```

task 72: ab session ko end krne ke liye cookie ko hum humare **expressSession** vaale middleware mei use kr sakte hai.

💡 basically session end krne ke liye you can see in application part to vahan nhi hota session mei end to aap khud se kr rhe ho end.

```
//app.js

const sessionConfig = {
    secret: 'weneedsomebettersecret',
    resave: false,
    saveUninitialized: true,
    cookie:{           //added from here
        httpOnly:true,
        expires:Date.now() + 1000*60*60*24*7,
        maxAge: 1000*60*60*24*7
    }
}
```

Samarth Vohra