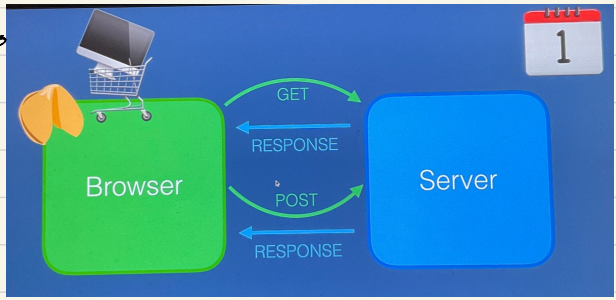
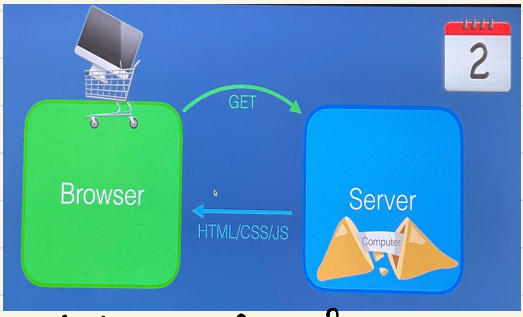




Cookie : Piece of Data

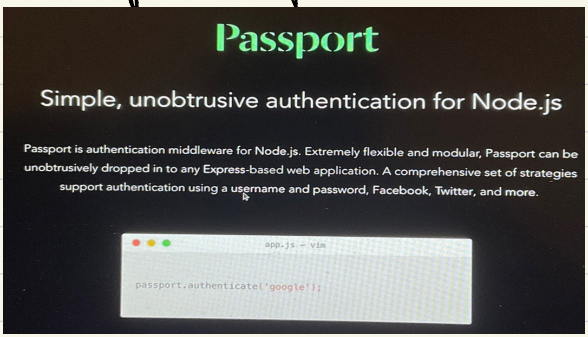
next time when Browser sends a get request

* Session

Time Browser interats with the server and that time cookie gets created

During session or log in cookies are there and when logout cookies get destroyed

Passport for authentication



* Use Passport

~ npm i passport passport-local passport-local-mongoose express-session

const session = require("express-session");

Const passport = require("passport");

const passportLocalMongoose = require("passport-local-mongoose")

```
app.use(express.static("public"));
app.set('view engine', 'ejs');
app.use(bodyParser.urlencoded({
  extended: true
}));

app.use(session({                          //Create session with some initial
  secret: "Our little secret.",             configuration
  resave: false,
  saveUninitialized: false
}));

app.use(passport.initialize()); // from here we can use passport
app.use(passport.session());    //use passport for session

mongoose.connect("mongodb://localhost:27017/userDB", {useNewUrlParser: true});
mongoose.set("useCreateIndex", true); //to avoid deprecation warning
                                        while running the server
const userSchema = new mongoose.Schema ({
  email: String,
  password: String
});

userSchema.plugin(passportLocalMongoose);//will hash & Salt our passwords
                                           and save users in database
const User = new mongoose.model("User", userSchema);

passport.use(User.createStrategy()); // to create local login strategy

passport.serializeUser(User.serializeUser());
passport.deserializeUser(User.deserializeUser());
```

* Serialise : Creates cookie and stuffs it up with information

Deserialise : Allows passport to crumble the cookie and discover the information of inside

```
app.post("/register", function(req, res){

  User.register({username: req.body.username}, req.body.password, function(err, user){
    if (err) {
      console.log(err);
      res.redirect("/register");
    } else {
      passport.authenticate("local")(req, res, function(){
        res.redirect("/secrets");
      });
    }
  });

});
```

method given by passport

name of strategy

if we are logged in we should stay logged in also in secrets route

```
app.get("/secrets", function(req, res){
  if (req.isAuthenticated()){
    res.render("secrets");
  } else {
    res.redirect("/login");
  }
});
```

```
app.post("/login", function(req, res){

  const user = new User({
    username: req.body.username,
    password: req.body.password
  });

  req.login(user, function(err){ // if login successful
    if (err) {
      console.log(err);
    } else {
      passport.authenticate("local")(req, res, function(){ // start authentication
        res.redirect("/secrets");
      });
    }
  });

});
```

```
app.get("/logout", function(req, res){
  req.logout();
  res.redirect("/");
});
```

* Note : Because our cookies are saved and authentication is on after I login I can use secrets without logging in again where login is required but after I logout I cannot use secrets without logging in

* Everytime we restart the server cookies are deleted