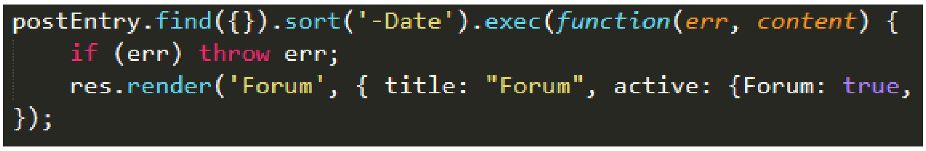
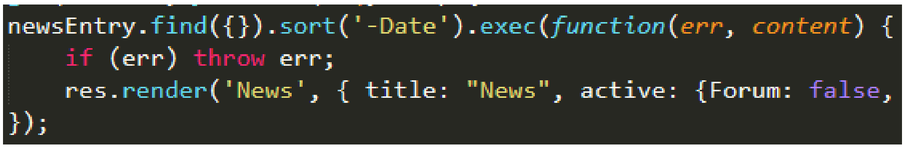


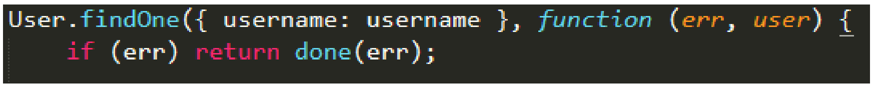
```
postEntry.find({}).sort('-Date').exec(function(err, content) {
    if (err) throw err;
    res.render('Forum', { title: "Forum", active: {Forum: true,
});
```

Searches for all documents inside the collection PostEntry and it sorts from the youngest to the oldest.



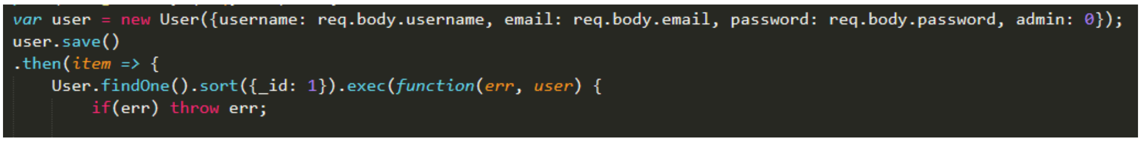
```
newsEntry.find({}).sort('-Date').exec(function(err, content) {
    if (err) throw err;
    res.render('News', { title: "News", active: {Forum: false,
});
```

Searches for all documents inside the collection NewsEntry and it sorts from the youngest to the oldest.



```
User.findOne({ username: username }, function (err, user) {
    if (err) return done(err);
```

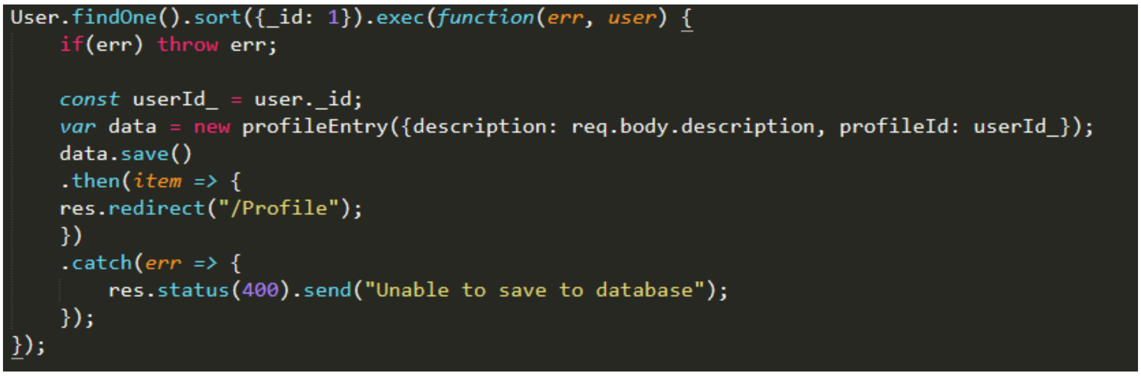
Searches for a document inside the collection User that has the username property matching the variable "username"



```
var user = new User({username: req.body.username, email: req.body.email, password: req.body.password, admin: 0});
user.save()
.then(item => {
    User.findOne().sort({_id: 1}).exec(function(err, user) {
        if(err) throw err;
```

The first line creates an instance of User and assigns it the properties accordingly to the form that was filed. It is then saved to the database.

Searches the collection of User and finds one and sorts the ids from newest to oldest.

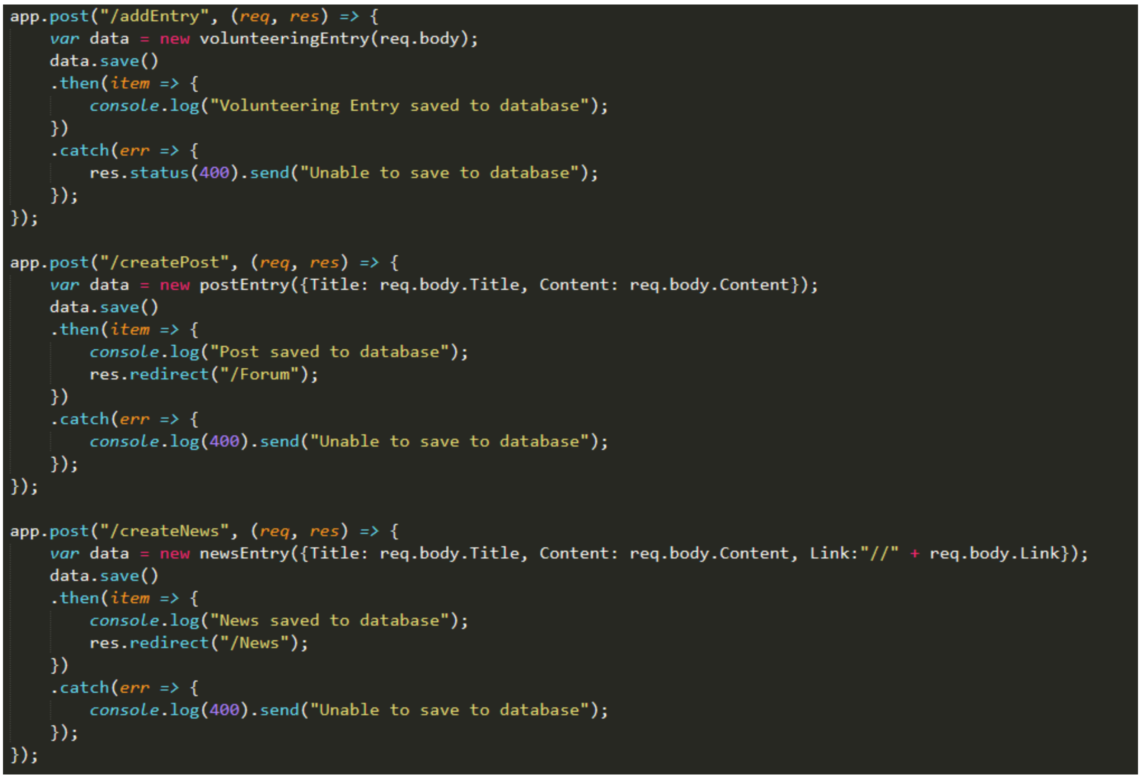


```
User.findOne().sort({_id: 1}).exec(function(err, user) {
    if(err) throw err;

    const userId_ = user._id;
    var data = new profileEntry({description: req.body.description, profileId: userId_});
    data.save()
    .then(item => {
    res.redirect("/Profile");
    })
    .catch(err => {
        res.status(400).send("Unable to save to database");
    });
});
```

Searches the collection of User and finds one and sorts the ids from newest to oldest.

Creates an instance of a profile entry and assigns it the properties accordingly to the form that was filed. It is then saved to the database.



```
app.post("/addEntry", (req, res) => {
    var data = new volunteeringEntry(req.body);
    data.save()
    .then(item => {
        console.log("Volunteering Entry saved to database");
    })
    .catch(err => {
        res.status(400).send("Unable to save to database");
    });
});

app.post("/createPost", (req, res) => {
    var data = new postEntry({Title: req.body.Title, Content: req.body.Content});
    data.save()
    .then(item => {
        console.log("Post saved to database");
        res.redirect("/Forum");
    })
    .catch(err => {
        console.log(400).send("Unable to save to database");
    });
});

app.post("/createNews", (req, res) => {
    var data = new newsEntry({Title: req.body.Title, Content: req.body.Content, Link:"//" + req.body.Link});
    data.save()
    .then(item => {
        console.log("News saved to database");
        res.redirect("/News");
    })
    .catch(err => {
        console.log(400).send("Unable to save to database");
    });
});
```

Creates an instance of a volunteering entry and assigns it the properties accordingly to the form that was filed. It is then saved to the database.

Creates an instance of a post entry and assigns it the properties accordingly to the form that was filed. It is then saved to the database.

Creates an instance of a news entry and assigns it the properties accordingly to the form that was filed. It is then saved to the database.