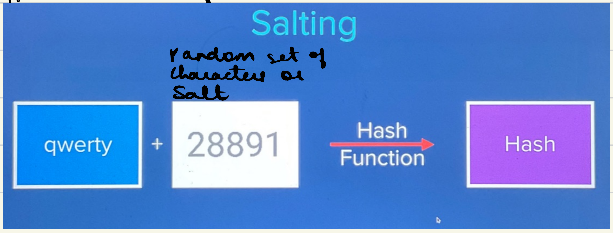
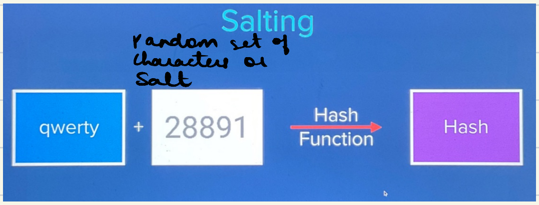
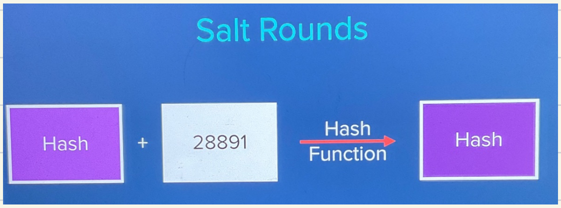
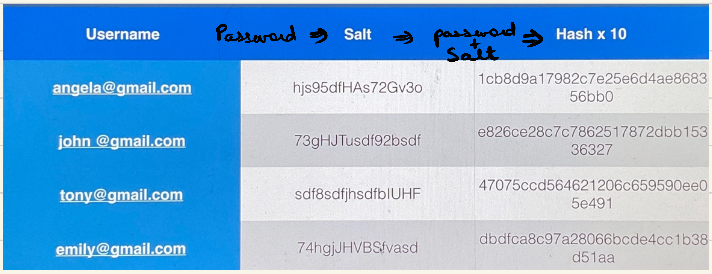
#Salting



*




*


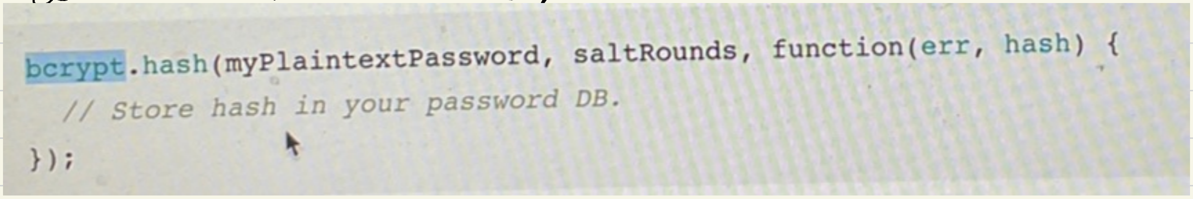
Username	Password ⇒ Salt ⇒	password + Salt ⇒ Hash x 10
angela@gmail.com	hjs95dfHAs72Gv3o	1cb8d9a17982c7e25e6d4ae8683 56bb0
john @gmail.com	73gHJTusdf92bsdf	e826ce28c7c7862517872dbb153 36327
tony@gmail.com	sdf8sdfjhsdfbIUHF	47075ccd564621206c659590ee0 5e491
emily@gmail.com	74hgjJHVBSfvasd	dbdfca8c97a28066bcde4cc1b38 d51aa

* node version should be stable version and not too old or not too new so to get back to stable version ⇒ install nvm then restart terminal

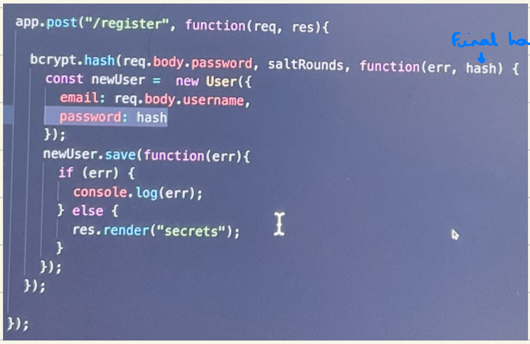
⇒ nvm install V10.15.0

We do the above to use bcrypt as it is version Sensitive

* const bcrypt = require ("bcrypt");
  const saltRounds = 10;

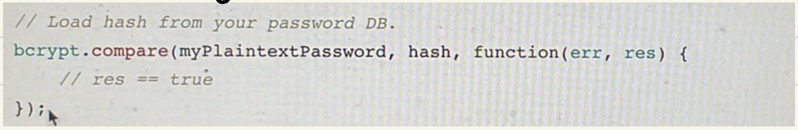


```
bcrypt.hash(myPlaintextPassword, saltRounds, function(err, hash) {
  // Store hash in your password DB.
});
```

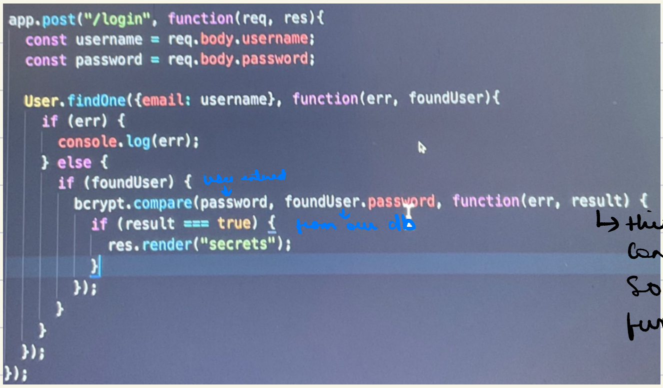
*


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

  bcrypt.hash(req.body.password, saltRounds, function(err, hash) {
    const newUser =  new User({
      email: req.body.username,
      password: hash
    });
    newUser.save(function(err){
      if (err) {
        console.log(err);
      } else {
        res.render("secrets");
      }
    });
  });

});
```

* To decrypt



```
// Load hash from your password DB.
bcrypt.compare(myPlaintextPassword, hash, function(err, res) {
    // res == true
});
```

*


```
app.post("/login", function(req, res){
  const username = req.body.username;
  const password = req.body.password;

  User.findOne({email: username}, function(err, foundUser){
    if (err) {
      console.log(err);
    } else {
      if (foundUser) {
        bcrypt.compare(password, foundUser.password, function(err, result) {
          if (result === true) {
            res.render("secrets");
          }
        });
      }
    }
  });
});
```

↳ this is also res but not to be confused we name it result
So if we get response we go further and render secrets file