Fullstack Development

Authentication / Authorization

Authentication

- A process of verifying user identity.
- Who is the user?
- Is the user really who he/she represents himself to be?

Authorization

- A process of verifying a user's access level.
- Is user `X` authorized to access resource `R`?
- Is user `X` authorized to perform operation `P`?

Let's think about it.

- User signing in with username/password.
  - `Authen` (`App`'s view)
- User signing in with Google account.
  - `Authen` (`App`'s view)
  - `Author` (`Google`'s view, `user` = `App`)
    - OAuth 2
- User requesting profile page.
  - `Author` (role-protected route)
  - `Authen` (query for user to display information)

Part 1: Signing up/in with credential

Part 1: Signing up/in with credential

> Section A: How to store password

Situation

- User fill in username and password.
- Your app creates user entry in database.
- *How do you store password?*

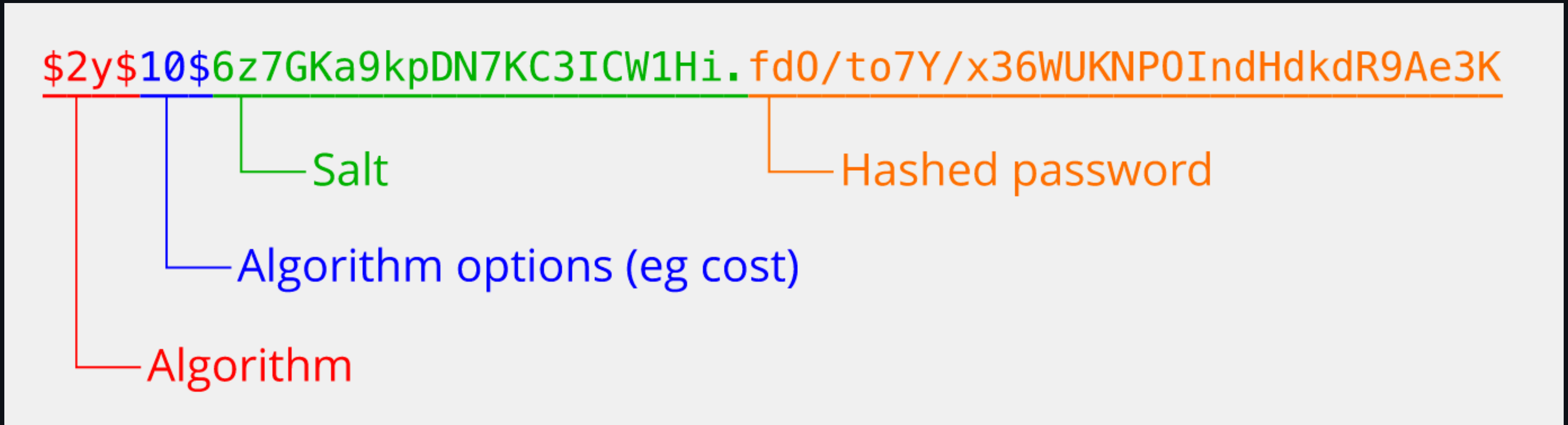
6 levels of safety

Technique	**Ranking**	**Vunerability**
Plain text	F	All
Encryption	D	Stolen key
Hashing	C	Rainbow table attack
Salting	B	Fast computer
Salting + Cost Factor (`bcrypt`)	B+	*Infinity stone* 🤣
?	A	

Adapted from source

Note

- SHA256
- Rawinbow table attack
- `bcrypt` hash



```
$2y$10$6z7GKa9kpDN7KC3ICW1Hi.fd0/to7Y/x36WUKNPOIndHdkdR9Ae3K
```

`bcrypt` example

- `git clone -b bcrypt https://github.com/fullstack-67/auth-mpa-v2.git auth-bcrypt`
- `pnpm i`
- `npx tsx ./src/hash.ts`
- `npx tsx ./src/compare.ts`

Part 1: Signing up/in with credential

> Section B: Implementation with `passport`

`passport`

- Most popular authentication middleware for `express`.
- Minimal and modular
- 500+ strategies (click at button)
- Confusing and poor documented 🤣
    - Hidden manual

Let's see it

- `git clone -b signin-credential https://github.com/fullstack-67/auth-mpa-v2.git auth-signin-credential`
- `pnpm i`
- `npm run db:reset`
- `npm run dev`

Side note about the project

- MPA - HTMX
- Use `SQLite` + `drizzle`.
  - Checkout the schema.
- Use `debug` package.
- Try debugging in VSCode.
  - See `launch.json`.

Highlighed packages

`package.json`

```
{
  "passport": "^0.7.0",
  "passport-local": "^1.0.0"
}
```

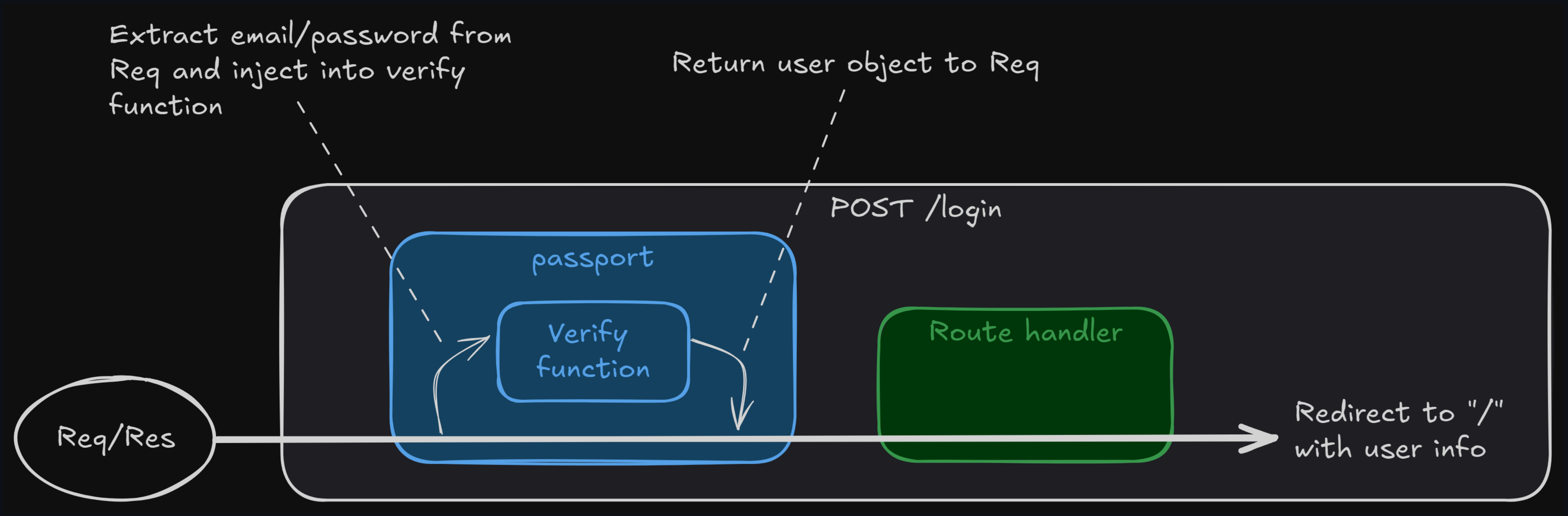
Middleware

src/index.ts

```
passport.use(
  new LocalStrategy(
    {
      // Options
    },
    async function (email, password, done) {
      // Verify email / password
    }
  )
);
//
app.use(passport.initialize());
```

Route

```
app.post(
  "/login",
  passport.authenticate("local", { session: false }),
  function (req, res) {
    // * Passport will attach user object in the request
  }
);
```



Extract email/password from Req and inject into verify function
Return user object to Req
POST /login
passport
Verify function
Route handler
Req/Res
Redirect to "/" with user info