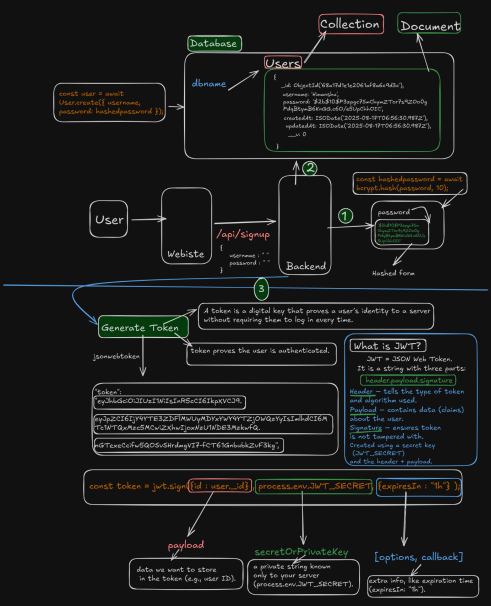
Authentication

1. Signup

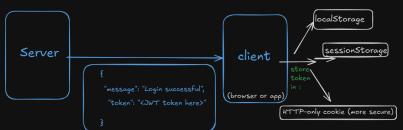


Why is it created?

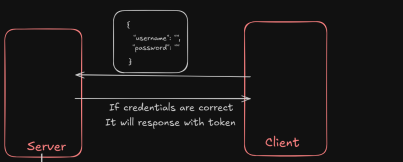
Authentication: After login/signup, server gives the token to the client.

Stateless sessions: Server does not need to store login info in memory or database.

Security: Only the server with the secret can verify the token.



2. Signin



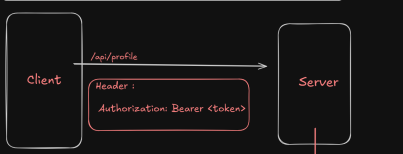
Task 1 `const user = await User.findOne({ username });`

To find the user exist or not in the database

Task 2 `const match = await bcrypt.compare(password, user.password);`

Match the password is correct or wrong.

3. JWT authentication middleware to protect routes



TASK 1 To check header

```
const authHeader = req.headers.authorization;
if (!authHeader || !authHeader.startsWith("Bearer ")) {
    return res.status(401).json({ message: "No token provided" });
}
```

TASK 2 To check Token

```
const token = authHeader.split(" ")[1];
var decoded = jwt.verify(token, process.env.JWT_SECRET);
if (!decoded) {
    return res.status(401).json({ message: "Invalid token" });
}
```

Task 3 Find user in DB → else 401.
Attach user to req.user.
Call next() to continue.

```
const user = await User.findById(decoded.id).select("-password");
if (!user) {
    return res.status(401).json({ message: "User not found" });
}
req.user = user;
next();
```

NOTE

`const user = await User.findById(decoded.id).select("-password");`

.select() is a Mongoose method to include or exclude fields in the returned document.
"-password" means: exclude the password field.

Why exclude password?

Security reason:

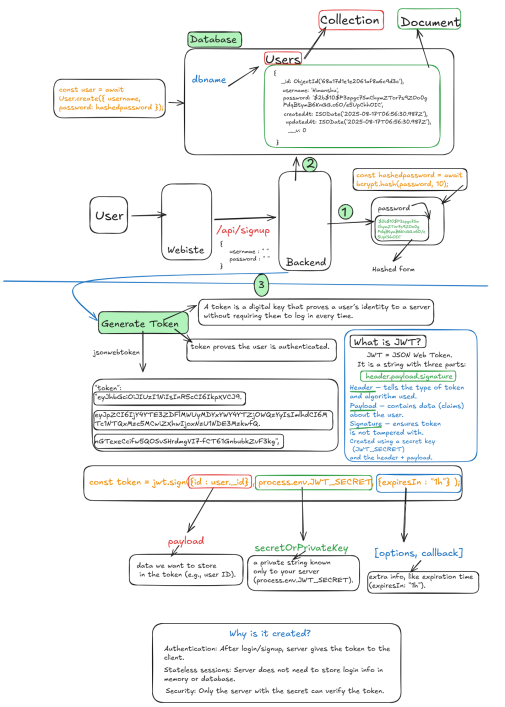
We never want to send hashed passwords to the client.

Even in internal middleware, it's good practice to not expose sensitive data.

req.user will contain only safe information that your routes can use.

Authentication

1. Signup

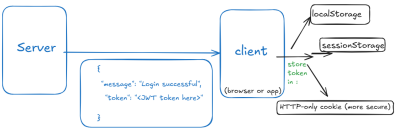


Why is it created?

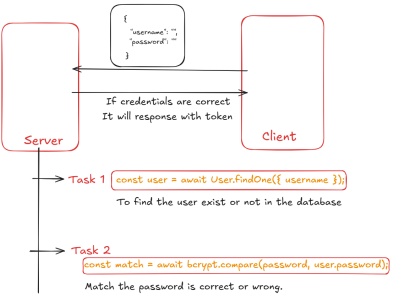
Authentication: After login/signup, server gives the token to the client.

Stateless sessions: Server does not need to store login info in memory or database.

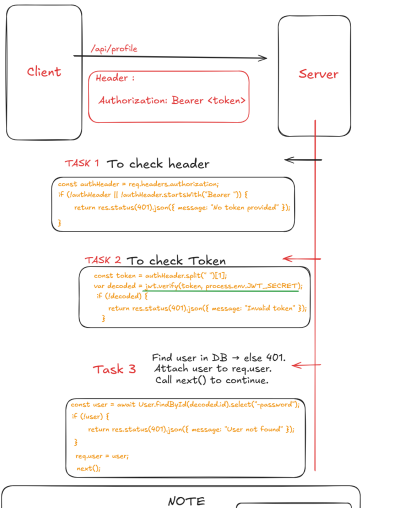
Security: Only the server with the secret can verify the token.



2. Signin



3. JWT authentication middleware to protect routes



NOTE

const user = await User.findById(decoded.id).select("-password");

.select() is a Mongoose method to include or exclude fields in the returned document.
"-password" means: exclude the password field.

Why exclude password?

Security reason:

We never want to send hashed passwords to the client.

Even in internal middleware, it's good practice to not expose sensitive data.

req.user will contain only safe information that your routes can use.