QUES: password: { type: String, required: true, select: false, } why there is select key

The `select` key in a Mongoose schema is used to control whether a field is included or excluded in query results by default.

Why `select: false`?

When `select: false` is set for a field (e.g., `password` in your case), it means:

- **Default Exclusion**: The `password` field will not be included in the results of queries unless explicitly requested. This is a common practice for sensitive data like passwords.
- **Security**: It reduces the risk of accidentally exposing sensitive data in responses.

Example in Mongoose:

Schema Definition:

```javascript
const userSchema = new mongoose.Schema({
  username: {
    type: String,
    required: true,
  },
  password: {
    type: String,
    required: true,
    select: false,
  },
});

const User = mongoose.model("User", userSchema);
```

Query Example:

1. Without Selecting Password:

```javascript
const user = await User.findOne({ username: "JohnDoe" });
console.log(user);
// Output: { _id: '...', username: 'JohnDoe' } (No password)
```

2. **Explicitly Including Password:**

```javascript
                                                                    Copy code
javascript

const user = await User.findOne({ username: "JohnDoe" }).select("+password");
console.log(user);
// Output: { _id: '...', username: 'JohnDoe', password: 'hashedPassword' }
```

QUES: - Why is this important to hide password?

The `select: false` key is important because it **hides sensitive information** like passwords from being shown automatically when you fetch user data from the database.

Simple Explanation:

- Imagine you're running a store, and you have a database with customer information.
- You don't want to accidentally show their **secret passwords** when you display their profiles, right?
- By setting `select: false`, the database **automatically skips** the password unless you specifically ask for it.

QUES - You don't want to accidentally show their secret passwords when you display their profiles, right? from whom we hidind data and to whom we displaying password

We are **hiding the password** from anyone who **doesn't need to see it**, such as:

1. **Clients (Users of Your App)**:
   - When you send user details (like their name or email) to the front-end or API responses, you don't want the user's password to be included by accident. For example:

```json
json

{
   "username": "JohnDoe",
   "email": "john@example.com"
   // no password here
}
```

Developers or Logs:

- If your app logs user data for debugging, you don't want sensitive data like passwords to appear in the logs.

Hackers or Security Risks:

- If your API or database gets exposed due to a bug, having passwords excluded by default reduces the chance of exposing sensitive data.

QUES : To whom are we displaying the password?

We explicitly include the password only when we **really need it**, such as:

1. **For Authentication**:
   - When a user logs in, the backend needs to compare the entered password with the one stored in the database (usually hashed). For this, you explicitly ask for the password: