

The screenshot shows a code editor with `server.js` and a browser window at `localhost:5000`. The browser displays a simple website with a navigation bar and a welcome message. Handwritten annotations show the value of `locals` as `currentPage: 'home'` and `userName: '_____'`.

```

server.js
14 app.use(bodyParser.urlencoded({ extended: false }));
15
16 const data = {
17   favDish: "Chhole Bhature",
18 };
19
20 app.get('/', (req, res) => {
21   res.render('Home', { currentPage: 'home' });
22 })
23
24 app.get('/aboutus', (req, res) => {
25   res.render('AboutUs', { currentPage: 'aboutus' });
26 })
27
28 app.get('/contactus', (req, res) => {
29   res.render('ContactUs', { currentPage: 'contactus' });
30 })
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

```

The screenshot shows two code editors. The left editor has `server.js` open, and the right editor has `MessagePage.ejs` open. A green arrow points from the `req.body` reference in `server.js` to the `formCredentials` variable in `MessagePage.ejs`. Handwritten notes include a stick figure pointing to the database icon.

```

server.js
34 app.get('/', (req, res) => {
35   res.render('Form');
36 });
37
38 app.post('/', (req, res) => {
39   const formCredentials = req.body;
40   if (formCredentials.userName === loginCredentials.username && formCredentials.password === loginCredentials.password) {
41     res.render('MessagePage', { userName: formCredentials.userName });
42   } else {
43     res.render('MessagePage', { userName: '' });
44   }
45 });
46
47 app.get('/aboutus', (req, res) => {
48   res.render('AboutUs', { currentPage: 'aboutus' });
49 });
50
51
52

```



① Structured Data (Tabular)

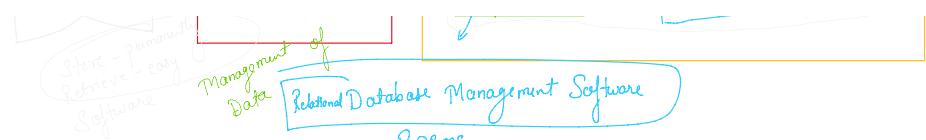
name	age	dept	marks	year
Kunal	23	AIML	72	2009
Shreyas	20	CSE	70	2010
Mayuri	20	AIML	65	2022
Kayan	18	AIML	81	2025

② Semi-Structured (document)

name : Kunal
age : 23
dept : AIML
marks : 72
year : 2009

③ Unstructured

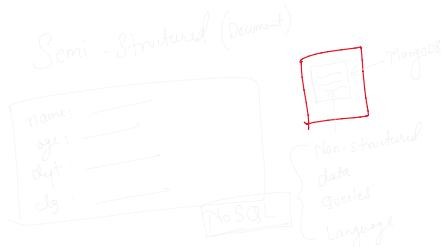




RDBMS

MySQL → Oracle } SQL

PostgreSQL

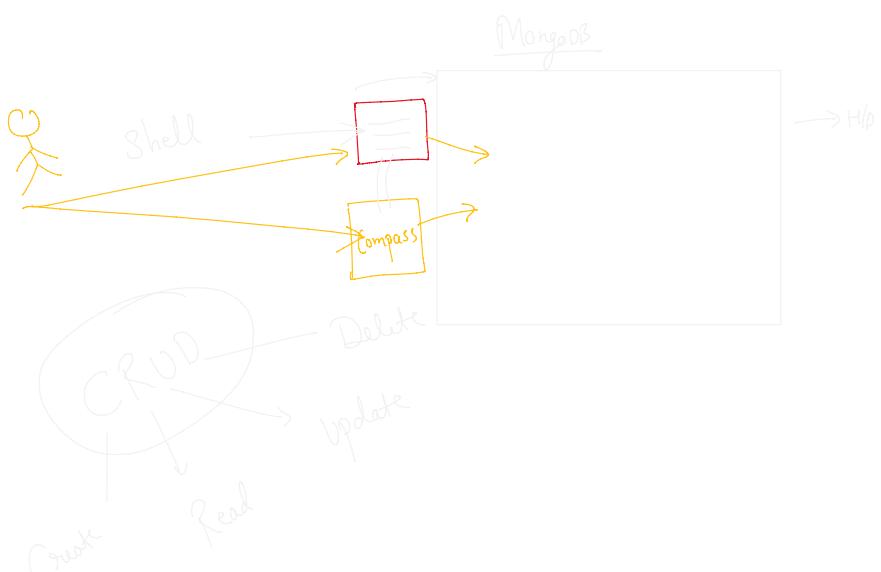


MySQL

- Database
- Tables
- Row

MongoDB

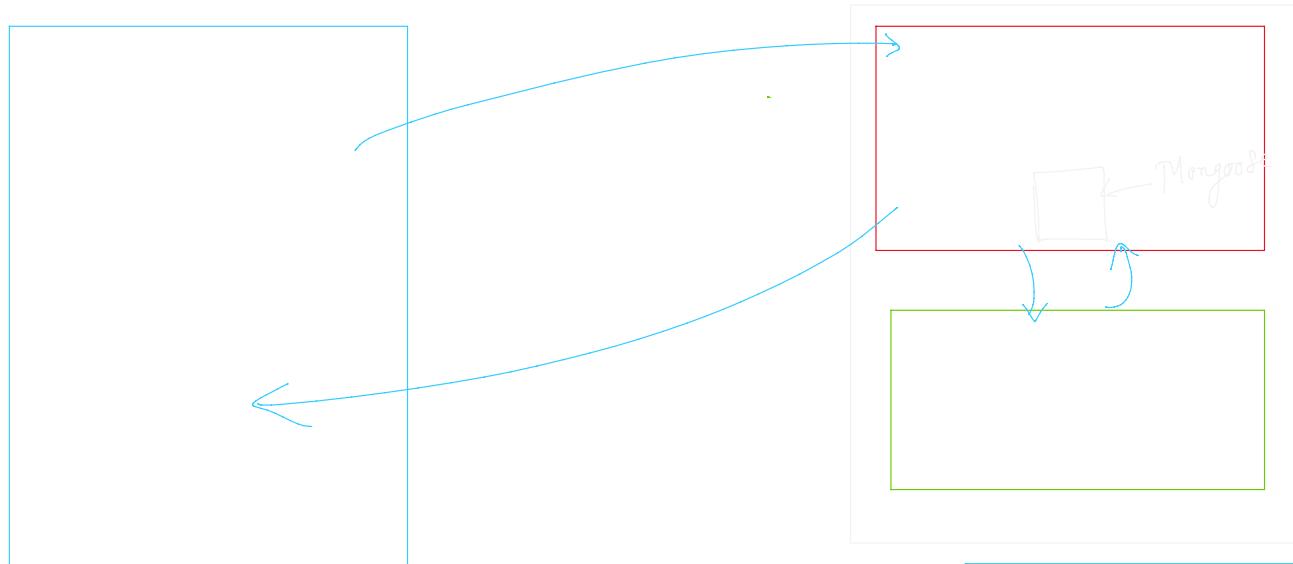
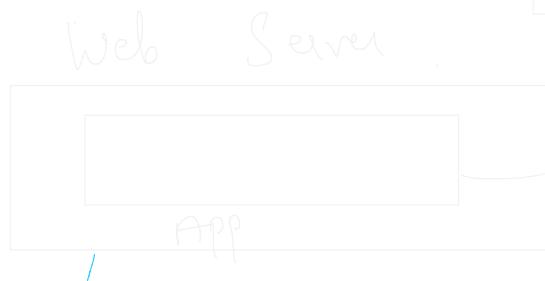
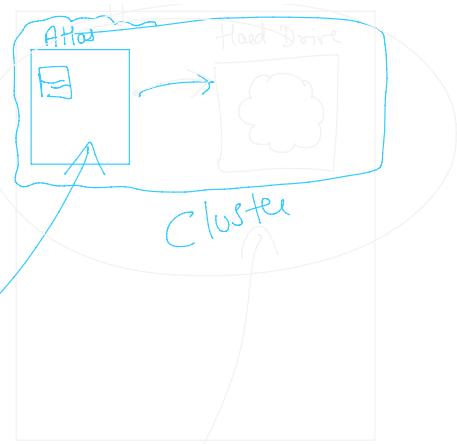
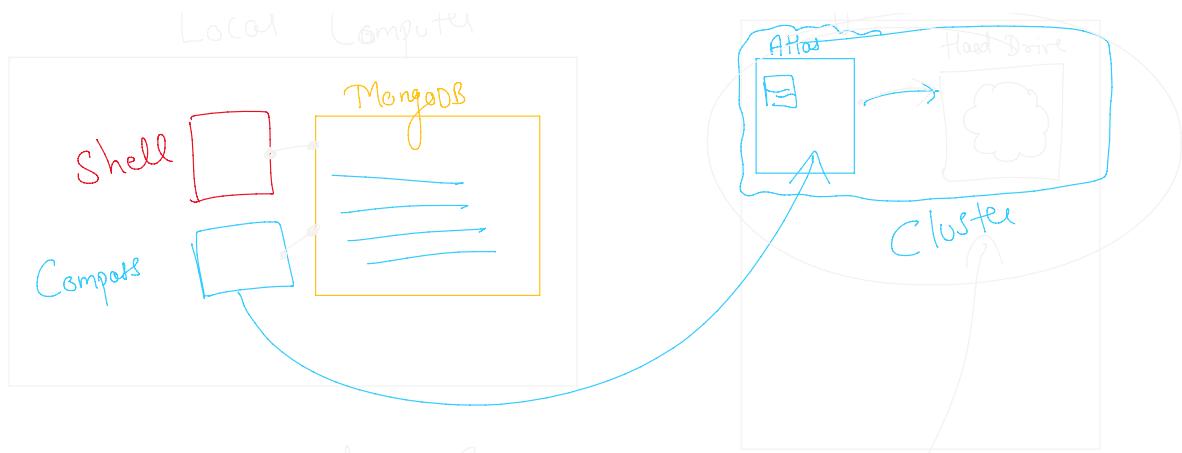
- Database
- Collections
- Documents



Local Computer

MongoDB





**server.js**

```

37 // 5. Basic Route: Home page (list all user profiles)
38 app.get('/', async (req, res) => {
39   try {
40     const profiles = await UserProfile.find({});
41     // Render 'index.ejs' returning an array of profiles
42     res.render('index', { profiles });
43   } catch (error) {
44     res.status(500).send(error.message);
45   }
46 });
47
48 // 6. Create (POST) a new user profile
49 app.post('/users', async (req, res) => {
50   try {
51     const newUser = await UserProfile.create(req.body);
52     // Redirect back to some page or return 500
53     res.redirect('/');
54   } catch (error) {
55     res.status(400).send(`Error creating user: ${error.message}`);
56   }
57 });
58
59 // 7. Start the server
60

```

The code editor shows several annotations with arrows and boxes:

- An annotation points from the "server.js" tab to the "UserProfile.js" file in the Explorer.
- Annotations highlight the `find()` and `create()` methods in the code.
- A large green box highlights the entire code block.
- A yellow box highlights the `res.redirect('/')` line.
- Annotations point from the code to a screenshot of a browser window showing a user profile.
- Annotations point from the code to a screenshot of a terminal window showing the command `rahul@Romeo-Alpha09 MINGW64 /r/MERN Stack VAP - Jan 25 ~ Course Resources/BackEnd-Tutorial/Mongoose-Tutorial`.

```
rahul@Romeo-Alpha09 MINGW64 /r/MERN Stack VAP - Jan 25 - Course Resources/BackEnd-Tutorial/Mongoose-Tutorial (main)
```

