MERN Backend Setup

1. Set up MongoDB Atlas

2. Create a `.gitignore` file and add `node_modules/` to the file.

3. Initialize the git repository `git init` in the project root directory.

4. Create a `README.md` file.

Install dependencies:

1. `npm i express express-validator bcryptjs config gravatar jsonwebtoken mongoose request`
2. `express`: Main web frame for the backend.
3. `express-validator`: Used for data validation. Will raise errors.
4. `bcryptjs`: Used for password encryption.
5. `config`: global variables.
6. `gravatar`: Profile avatars
7. `jsonwebtoken`: Using JWT to pass a token for validation.
8. `mongoose`: Used to interact with database.
9. `request`: Module to allow HTTP requests to API's. Installed for the GitHub repository profiles to be listed for the backend.

Install nodemon and concurrently

1. `npm i -D nodemon concurrently`
2. `nodemon` watches the server for any changes and automatically makes changes.
3. `concurrently` will allow the backend Express server and frontend React dev server at the same time.

Make the following file, `server.js`, in the root directory.

- This is the content of the server.js file

```
const express = require('express'); // Bring in Express

const app = express(); // Entialize app varialble with express

app.get('/', (req, res) => res.send('API Running')); // End point for testing

const PORT = process.env.PORT || 5000; // App varialble listining on a PORT

app.listen(PORT, () => console.log(Server started on port ${PORT})); // PORT varialbe
```

Create Main Entry File in the root directory:

- `server.js` Add the following code to the file.

```
const express = require('express');

const app = express();

app.get('/', (req, res) => res.send('API Running'));

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(Server started on port ${PORT}));
```

Modify package.json scripts

1. Re-palace `"test": \"Error: no test specified\" && exit 1` with:
2. `"start": "node server"`

and add: - `"server": "nodemon server"` used for development.

Start Server

- `npm run server`

Then go to Postman and run a GET http://localhost:5000 and SEND. You should get the following: `API Running`