

H To-do App.

Haitham Alhaji

Std. ID 1801886

* **Pages:**
  + Home
  + Login
  + Register
  + Admin. Panel (only for admin account)
  + MyToDo
* **Features**:
  + Session Mana.
  + Database MySQL (Storing ToDo list)
  + User types (isolation permissions)
  + Securing from SQL Injection (with escape methodology)
  + Securing from XSS (with form validation)
  + Securing GLOBAL Variables with Environment Variables
* **Technologies**:
  + NodeJS with Express Server
  + MySQL Database
  + My own CentOS 7 Server with Apache Server
  + HTML5 With CSS Enhancement (SASS)
  + Handlebars (code Segmentation)
* Links
  + Website (live demo): <http://todo.haitham.xyz/>
    - Admin. Info. (Email: [g@y.com](mailto:g@y.com), password: g@y.com)
  + Source code: <https://github.com/HaithamAlhaji/WebHToDo>

**Securing from SQL Injection (with escape methodology)**

Direct variable in SQL Query will make attacker implement his injection directly, therefore, there is no doubt to use escape method to clear your variable from any kinds of injection. For example:

1. Home / Login (POST)



1. Home / Register (POST)



1. Home / AddTask (POST)



1. Home / Search for task (POST)



1. Home / List tasks (POST)



**Securing from XSS (with form validation)**

Although Inputs are a compulsory channel that is letting the user communicate with our app., Strict entering inputs are mandatory. Therefore, form validation is a technique which plays a vital role to secure that has mentioned before. For example:

1. Login (HTML)
2. Login (JS)
3. Regsiter (HTML)
4. Regsiter (JS)
5. AddTask (HTML)
6. AddTask (JS)
7. Search (HTML)
8. Search (JS)