

Valencia 20 de sept. de 2024

Sr. José Mario Garzón.

Presente.

Estimado Señor José Mario Garzón, es un gusto saludarle y deseándole lo mejor en todas y cada uno de sus actividades y aspiraciones. Le escribo para presentarle el reporte mensual de actividades junto con las metas, logros, inconvenientes, sugerencias y perspectivas del proyecto escolar que desarrollo para usted dedicando el mayor esfuerzo y dedicación para obtener óptimos resultados.

Le informo que su proyecto se está desarrollando con un framework de PHP llamado Laravel en su versión 11 y es complementado por otros frameworks y librerías tales como livewire v3, Alpine.js, JQuery.js, JQuery Datatables.js y estilizados con Tailwindcss todo con el fin de potenciar su funcionamiento y versatilidad.

1. Instalación del estado actual del proyecto:
 - a. Se debe tener instalado GIT para poder clonar el repositorio.
 - b. Se usará VsCode como editor y debe estar instalado.
 - c. Se debe instalar Xamp para correr base de datos
 - d. Ubicarse en la carpeta C:\xampp\htdocs
 - e. Abrir un nuevo terminal o abrir vscode new terminal.
 - f. Ejecutar en terminal el siguiente código para la clonación del repositorio:Git clone <https://github.com/burrodecarga/jmg.git>
 - g. Ingresar en la carpeta reada y buscar .env.example y renombrarlo quitando el .example
 - h. En el archivo .env hacer los siguientes cambios:

```
2. DB_CONNECTION=mysql
3. DB_HOST=127.0.0.1
4. DB_PORT=3306
5. DB_DATABASE=jmg
6. DB_USERNAME=root
7. DB_PASSWORD=
```

- i. Se de crear la base de datos, abrir MysqlAdmin y crear una base de datos llamada jmg.
- a. Ingresar en la carpeta clonada y ejecutar los siguientes comandos:
 - i. Composer Install
 - ii. Php artisan migrate:refresh --seed
 - iii. Npm run build
 - iv. Npm run dev
 - v. Php artisan serve
 - vi. Pulsar en localhost 127.0.0.1:8000 y esperar que inicie la aplicación.
 - vii.

Sobre lo desarrollado:

Sistema de logueo y seguridad:

Para la seguridad y acceso se usan diferentes métodos de control:

1. verificación de email.
2. Email único
3. Sistema de roles y permisos.
4. Framework Fortify y Jetstream
5. Middleware, Policy y lógica de programación.

Los métodos 1,2 y 4 son proporcionados por Laravel y complementos diseñados específicamente para seguridad y control de logueo.

Hasta el momento se han creado 65 permisos para accesos a rutas y 14 roles a los que se le asignarán esos permisos.

Roles:

id	name	guard_name
14	no role	web
13	inactive	web
12	employee	web
11	data-manager	web
10	secretary	web
9	parent	web
8	student-high	web
7	student-middle	web
6	student-basic	web
5	teacher	web
4	manager	web
3	coordinator	web
2	administrator	web
1	super-admin	web

Permisos:

id	name	privilege	guard_name
1	roles.index	role list	web
2	roles.create	role create	web
3	roles.store	role create	web
4	roles.edit	role edit	web
5	roles.update	role edit	web
6	roles.destroy	role delete	web
7	roles.show	role view	web
8	coordinators.index	coordinator list	web
9	coordinators.create	coordinator create	web
10	coordinators.store	coordinator create	web
11	coordinators.edit	coordinator edit	web
12	coordinators.update	coordinator edit	web
13	coordinators.destroy	coordinator delete	web
14	coordinators.show	coordinator view	web
15	users.index	user list	web
16	users.create	user create	web
17	users.store	user create	web
18	users.edit	user edit	web
19	users.update	user edit	web
20	users.destroy	user delete	web
21	users.show	user view	web
22	schools.index	school list	web
23	schools.create	school create	web
24	schools.store	school create	web
25	schools.edit	school edit	web
26	schools.update	school edit	web
27	schools.destroy	school delete	web
28	schools.show	school view	web
29	sedes.index	sede list	web
30	sedes.create	sede create	web
31	sedes.store	sede create	web
32	sedes.edit	sede edit	web
33	sedes.update	sede edit	web
34	sedes.destroy	sede delete	web
35	sedes.show	sede view	web
36	resources.index	resource list	web
37	resources.create	resource create	web
38	resources.store	resource create	web
39	resources.edit	resource edit	web
40	resources.update	resource edit	web
41	resources.destroy	resource delete	web
42	resources.show	resource view	web
43	grados.index	grado list	web
44	grados.create	grado create	web
45	grados.store	grado create	web
46	grados.edit	grado edit	web
47	grados.update	grado edit	web
48	grados.destroy	grado delete	web

49	grados.show	grado view	web
50	courses.index	course list	web
51	courses.create	course create	web
52	courses.store	course create	web
53	courses.edit	course edit	web
54	courses.update	course edit	web
55	courses.destroy	course delete	web
56	courses.show	course view	web
57	padres.index	padre list	web
58	padres.create	padre create	web
59	padres.store	padre create	web
60	padres.edit	padre edit	web
61	padres.update	padre edit	web
62	padres.destroy	padre delete	web
63	padres.show	padre view	web
64	padre.asignar	padre asignar	web
65	padres	padre inscribir	web

Estado de la Aplicación.

1. Modelo Permisos: Como se puede intuir es el modelo que da o no acceso a una determinada ruta o función.
100% operativo y crece en número cada vez que se desarrolla una función que implique una actividad que deba realizar un usuario.
2. Modelo Roles: Permite agrupar una serie de permisos a múltiples usuario.
Está 100% operativo y se cree que no se necesitan más roles pero igualmente el usuario superadministrador podrá crear, editar y eliminar los roles que él mismo cree. Los 14 roles originales no podrán ser modificados o eliminados para preservar la integridad del sistema.
3. Modelo User: Es el modelo básico de todos los usuarios del sistema excluyendo a niños menores (edad a discutirse) pero incluyendo a padres y representantes.
100% operativo por parte del role superadministrador que puede modificar su rol dentro del sistema.
Se debe discutir que puede modificar cada usuario de su perfil.
4. Modelo School: Es el modelo que representa a las escuelas y contiene la información básica de todas sus sedes, es único. 100% operativo.
5. Modelo Sede: Es la representación de cada escuela física que es sede del modelo school.
100% operativo, cada modelo sede tiene las siguientes características:
 - a. Una sede tiene muchas aulas 100%
 - b. Una sede tiene muchos profesores 100%
 - c. Una sede tiene muchos recursos 0% operativo 100% lógica
 - d. Una sede tiene muchos estudiantes 0% se debe llegar a este punto.
 - e. Una sede tiene muchos grados escolares 100%.
 - f. Una sede tiene muchos coordinadores 100%.
6. El modelo Room (aula) representa los espacios físicos de cada sede y permiten cálculos presupuestarios y carga humana pues maneja largo, ancho y alto a futuro puede servir como base de ubicación, cada aula puede albergar diferentes grados en diferentes tiempos. 100% funcional 0% en implantación de uso, no se tiene claro cómo se usará.

7. El modelo Teacher es una modificación de la estrategia inicial que era usar sólo un modelo para profesores, alumnos, usuarios, etc; se espera mayor complejidad. Un profesor puede dar varias materias y trabajar en diferentes sedes, esa lógica está 100%.
En las relaciones y necesidades del sistema y es mucho más práctico y eficiente pues separa las funcionalidades de los modelos operativos que representan a humanos.
8. El modelo Grado representa a todos los grados exintendentes, no son modificables ni eliminables, pero se pueden agregar nuevos grados que si son modificables y eliminables. Una sede puede tener disponible la totalidad o parte de los grados existentes. 100% operativo y lógico.
9. El Role de superadministrador y sus funcionalidades 100% operativos
10. SE trabaja con el Role Coordinador que es quien controla las sedes
 - a. 100% agregar profesores.
 - b. 100% agregar grados.

Hasta este punto se ha llegado a la fecha de este reporte.

