This is a technical guide on how the database , ORM, backend, and front end work for this application.

**Exposed API 🡪 ORM 🡪 Database:**

**The API Endpoints of the project are exposed here.**

*\reimbursementAPP\Reimbursement\_Vuetify\_Go\GolangPostGresBE\routes\routes.go*

Text

Description automatically generated

*This is where all new endpoints will be listed. Currently all endpoints logic is in the controllers/AuthControllers.go file. (This can be abstracted out to separate files later if need be.)*

**Reviewing the login endpoint, backend validation, and creation of JWT in cookie response.**

*Below is backend validation before JWT is created.*

Text

Description automatically generated

*Once validation is finished JWT is created and returned as a cookie.*

Text

Description automatically generated

**Using GORM (GO’s Object Relational Mapper)**

Main.go calls ConnectToDB function. When application is first started.

Text

Description automatically generated

Once connected to Database GORM automatically creates and connects to the data transition objects (DTO’s).

Text

Description automatically generated

Example of DTOs

Text

Description automatically generated

Text

Description automatically generated

You can now call GORM methods to access database.

Text

Description automatically generated

**Authentication User via JWT cookie before performing logic is in “checkCookieGetToken” function.**

Returns a token or error depending.

Text

Description automatically generated

Text

Description automatically generated

**Frontend VueX Store.**

**The VueX store allows you to store states of variables to be referenced through out the application.**

The state is the variables property, the mutation is a setter, and the getter is just that a getter.

A picture containing timeline

Description automatically generated

When the Dashboard component is created it called the get User API by passing in the JWT Cookie.

The Backend responds with the users ID and whether they are an Admin or a Regular User.

We then store this data in the VueX store.

Text

Description automatically generated

**Vue Router via VueX store.**

The routing functionality references the VueX store to verify if the user is logged in or already logged in.Text

Description automatically generated

Routes check this logic before proceeding.

Text

Description automatically generated

Dashboard displays based on these vueX store properties.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence