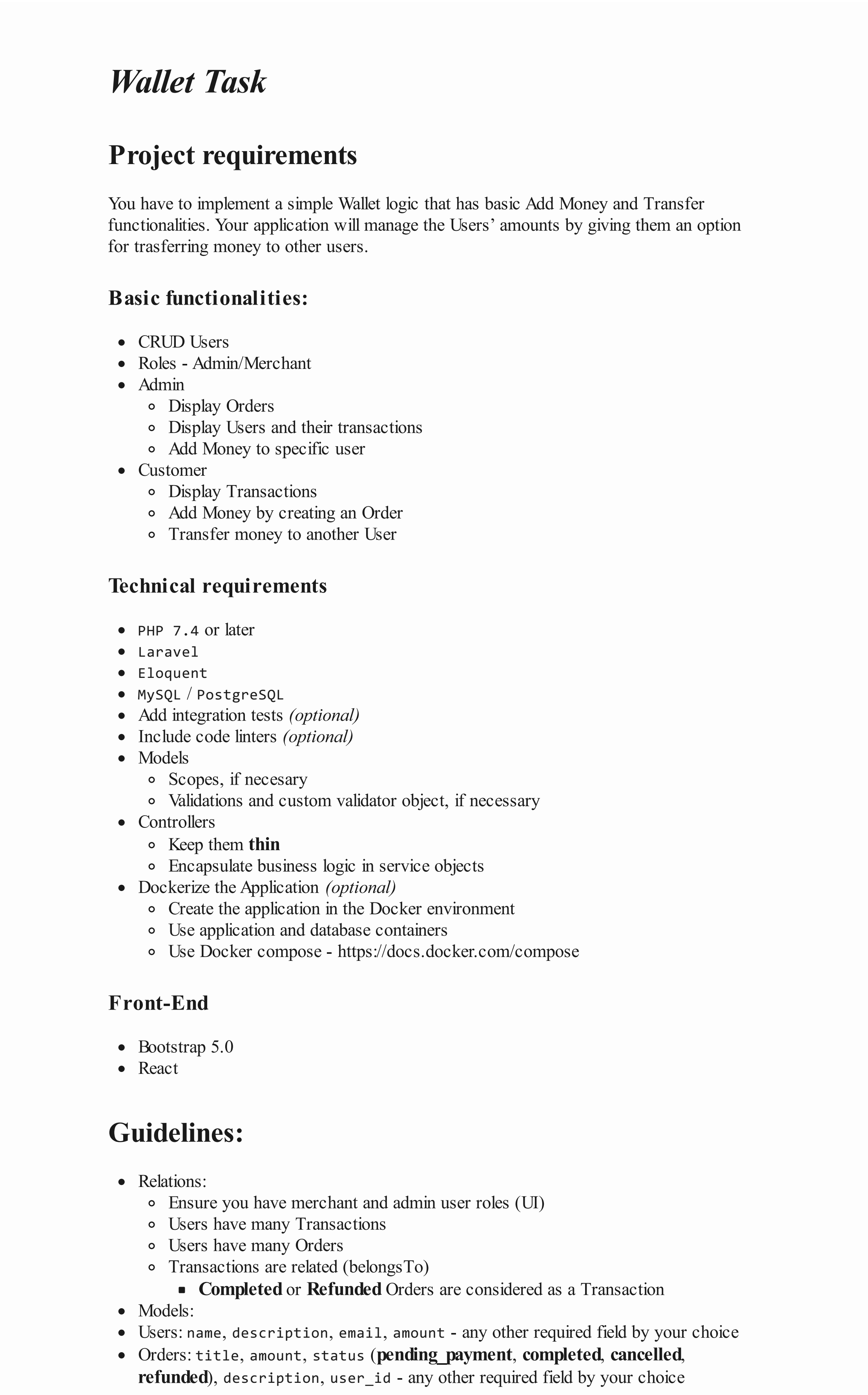

Wallet Task

Project requirements

You have to implement a simple Wallet logic that has basic Add Money and Transfer functionalities. Your application will manage the Users' amounts by giving them an option for trasferring money to other users.

Basic functionalities:

- CRUD Users
- Roles - Admin/Merchant
- Admin
  - Display Orders
  - Display Users and their transactions
  - Add Money to specific user
- Customer
  - Display Transactions
  - Add Money by creating an Order
  - Transfer money to another User

Technical requirements

- `PHP 7.4` or later
- `Laravel`
- `Eloquent`
- `MySQL` / `PostgreSQL`
- Add integration tests *(optional)*
- Include code linters *(optional)*
- Models
  - Scopes, if necesary
  - Validations and custom validator object, if necessary
- Controllers
  - Keep them **thin**
  - Encapsulate business logic in service objects
- Dockerize the Application *(optional)*
  - Create the application in the Docker environment
  - Use application and database containers
  - Use Docker compose - https://docs.docker.com/compose

Front-End

- Bootstrap 5.0
- React

Guidelines:

- Relations:
  - Ensure you have merchant and admin user roles (UI)
  - Users have many Transactions
  - Users have many Orders
  - Transactions are related (belongsTo)
    - **Completed** or **Refunded** Orders are considered as a Transaction
- Models:
- Users: `name`, `description`, `email`, `amount` - any other required field by your choice
- Orders: `title`, `amount`, `status` (**pending_payment**, **completed**, **cancelled**, **refunded**), `description`, `user_id` - any other required field by your choice

  - Orders can be considered like “external” way of adding money by various sources like TopUp from bank accout, credit card or even from paying to a cashier. In terms of simplicity the external sources are not in the scope of the task.
  - Upon **completed** status **credit** transaction will be created
  - Upon **refunded** status **debit** transaction will be created
  - Upon **cancelled** status no actions will be taken
  - Use `events` *(optional)*
  - Merchant *Add Money* will create an Order where the Admin will be able to change its status, in terms of simplicity the Merchant will receive a success message that Order is received (**pending_payment**)
- Trasactions: `user_id`, `type` (**credit**, **debit**), `amount`, `description`, `created_by` (**user_id**) - any other required field by your choice
  - **credit** type will raise the User amount
  - **debit** type will decrease the User amount
  - Every user will be able to view all of the Wallet Transactions
  - Admins will be able to crate **credit** or **debit** transactions - those will be like internal, no Orders required for them, no **amount** is required to satisfy the **credit** transaction
  - **description** should contain infomartion like, *Received fund from* `user_email` or if the amount is received by Order *Purchased funds #order_id*

Submit the task

It is required to upload the code to a public/private GitHub repo or archive the git folder of the project by sending it to us via email.

- Git records (commits) should be grouped in the following way:
  - All subsequent commits should be logically organized, reflecting the steps you’ve taken developing the application.
  - Neither one large commit with all changes nor a multitude of smaller commits for every little tiny change.
  - Also, the task should have a simple setup guide. If you like, you can use the README for that purpose.