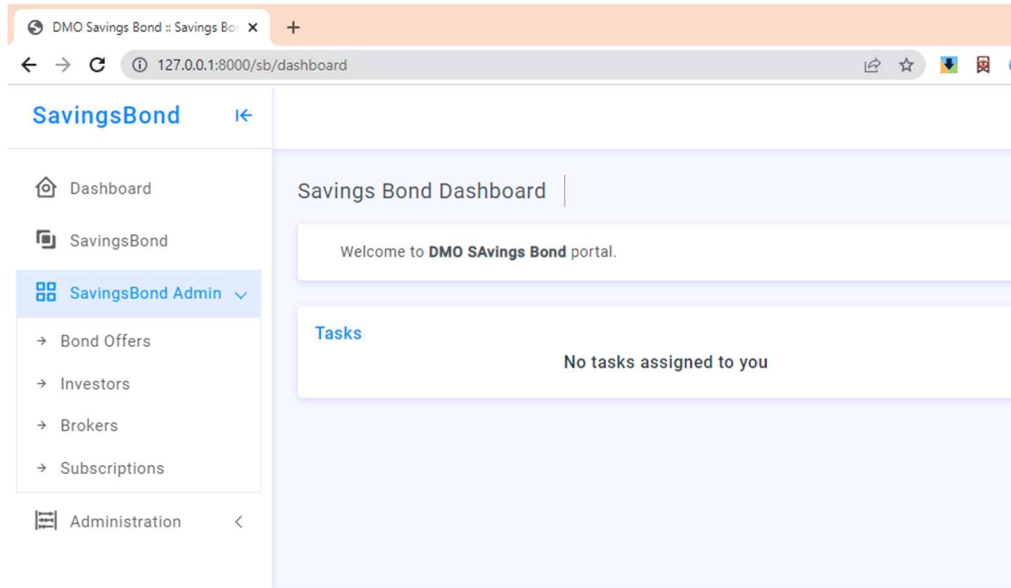


DMO Savings Bond Paid Task

1. The DMO Savings bond is a portal that allows investors to buy bonds.
2. The application comprises of 3 main packages
 - a. **dmo-savings-bond-portal**:- a web application
 - b. **dmo-savings-bond-module**:- a library containing the models, controllers, etc. for the savings bond app.
 - c. **hasob-foundation-core-bs-5**:- a helper library with reusable components.
3. The zip file contains 3 packages. Please unzip the file and confirm the following:
 - a. The `composer.json` file in **dmo-savings-bond-portal** contains local dependency on **dmo-savings-bond-module** and **hasob-foundation-core-bs-5**
 - b. The `composer.json` file in **dmo-savings-bond-module** contains local dependency on **dmo-savings-bond-module** and **hasob-foundation-core-bs-5**
4. Run *composer update* from inside **hasob-foundation-core-bs-5** package and then from inside **dmo-savings-bond-module**, and then from inside **dmo-savings-bond-portal**. This should resolve your package dependencies.
5. Then from inside **dmo-savings-bond-portal**, ensure you have setup your local database, then run *php artisan migrate* to setup your tables in the database.
6. Finally, you may run *php artisan serve* from inside **dmo-savings-bond-portal** to start the application.
7. The application should be running on your localhost at <http://127.0.0.1:8000>

8. Please login with the following credentials, [admin@app.com/password](mailto:admin@app.com)

9. This will bring you to the main dashboard as shown below.



10. The application contains several entities:

- a. **Bond Offer**:- A bond being offered for subscription by investors
- b. **Investor**:- A person subscribing to a bond offer.
- c. **Broker**:- Every Investor has a broker that manages their account.
- d. **Subscription**:- A successfully purchased bond offer by an investor.

11. The application is built in a package module style, the models, controllers, events, requests, etc. are all located in the **dmo-savings-bond-module** package.

12. Kindly, study the migration files, Controllers, Models, the Form Request Classes, the Events and Listener Classes.

13. Please you should concern your paid assignment to only the **Subscription** entity, its model, controllers, events, listeners, and UI.

- a. <http://127.0.0.1:8000/sb/subscriptions>

14. For your specific assignment you will be required to the following:

- a. Create test cases in dmo-savings-bond-portal/tests folder:
 - i. CRUD test to ensure the model creates a record, retrieves records, updates records, and delete records.
 - ii. Relationship tests to ensure that the relationship functions in the models return the appropriate relationships.
 - iii. Event/Listener test to ensure that the events are being raised and fired correctly.
- b. User Interface improvements
 - i. Each entity has 3 main user interfaces, the list page, all the records, with links to view, edit, and delete.
 - ii. A create user interface to create a new record
 - iii. A detail page user interface to display details a record.
 - iv. You will be required to complete
 1. The listing user interface.
 2. The create user interface, this is a modal screen accessible by clicking the new button on the top right corner of the page.
 3. The edit user interface, same modal screen as the create user interface, but handling edit functionality.
 4. The detail page, showing the entity and its related records.