Instructions

We have used the latest technologies and tools to implement our product. The app is divided into three components react_app, python_service, and php_server.
The zipped file contains the code for software, which is divided into three sections as mentioned earlier.

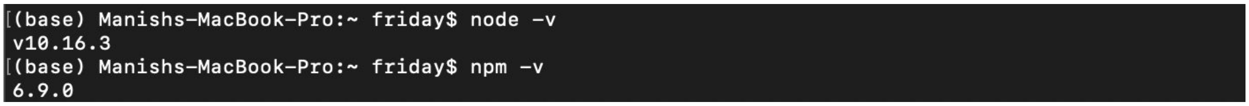
We will separately install all three components

React Installation

We used ReactJS for the development of interface, to install react make sure you have node and npm installed.

1. Check the node and npm by running the following commands -

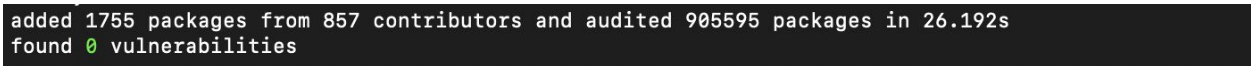
```
$ npm -v
$ node-v
```



```
[(base) Manishs-MacBook-Pro:~ friday$ node -v
v10.16.3
[(base) Manishs-MacBook-Pro:~ friday$ npm -v
6.9.0
```

2. To install the dependencies, make sure to change the directory to the react app.

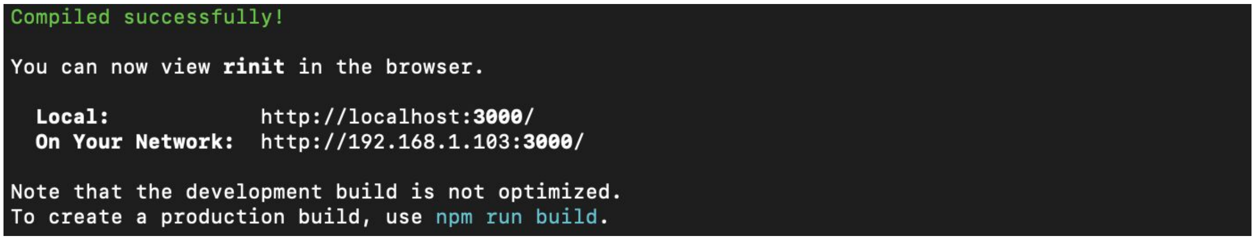
```
$ cd react_app
$ npm install
```



```
added 1755 packages from 857 contributors and audited 905595 packages in 26.192s
found 0 vulnerabilities
```

3. To start the Reactjs App

```
$ npm start
```



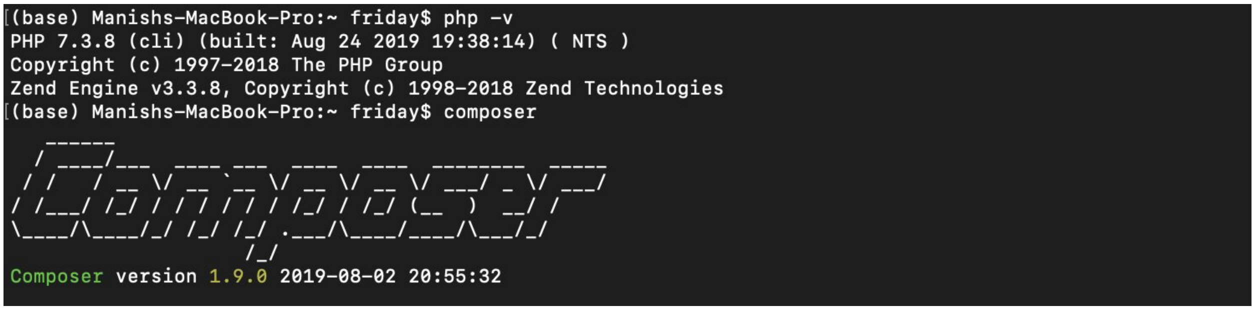
```
Compiled successfully!

You can now view rinit in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.1.103:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Laravel Installation

We used Laravel for the development of the backend server, to install Laravel make sure you have PHP and composer installed.

1. Check the node and npm by running the following commands -

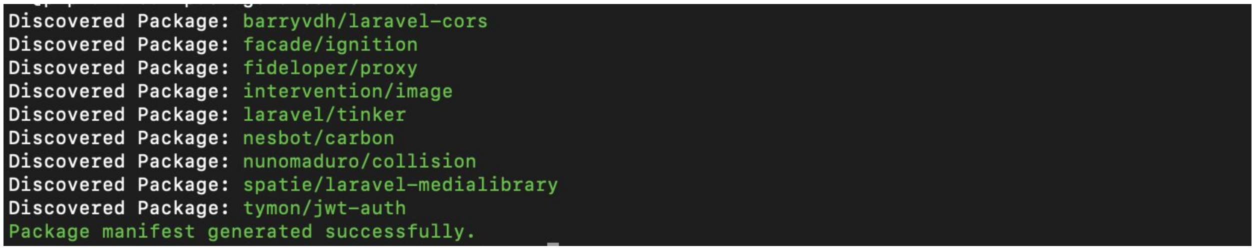
```
$ php -v
$ composer -v
```



```
[(base) Manishs-MacBook-Pro:~ friday$ php -v
PHP 7.3.8 (cli) (built: Aug 24 2019 19:38:14) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.8, Copyright (c) 1998-2018 Zend Technologies
[(base) Manishs-MacBook-Pro:~ friday$ composer
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 1.9.0 2019-08-02 20:55:32
```

2. To install the dependencies, go inside the php_server dire

```
$ cd php_server
$ composer install
```



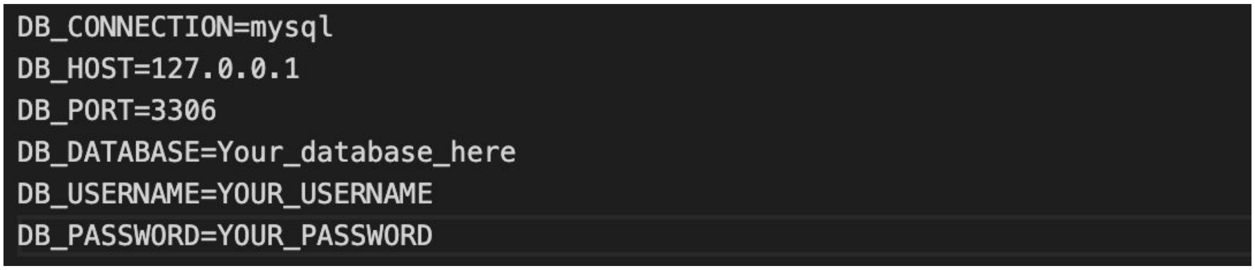
```
Discovered Package: barryvdh/laravel-cors
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: intervention/image
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: spatie/laravel-medialibrary
Discovered Package: tymon/jwt-auth
Package manifest generated successfully.
```

3. Create and env file by copying .env.example to .env.

```
$ cp .evn.example .env
```

Make the following changes to .env

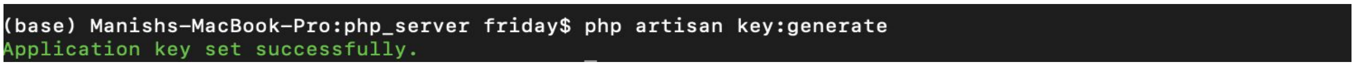
1. Change the Database parameters



```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=Your_database_here
DB_USERNAME=YOUR_USERNAME
DB_PASSWORD=YOUR_PASSWORD
```

4. Set App key by running

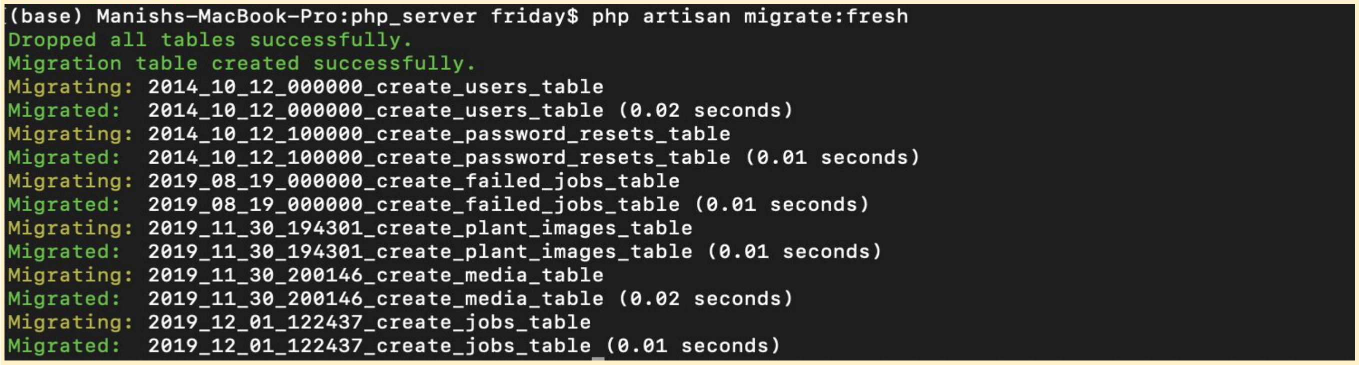
```
$ php artisan key:generate
```



```
(base) Manishs-MacBook-Pro:php_server friday$ php artisan key:generate
Application key set successfully.
```

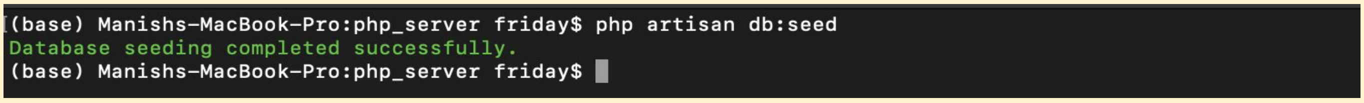
5. Run migrations and seeders

```
$ php artisan migrate
```



```
(base) Manishs-MacBook-Pro:php_server friday$ php artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated:  2014_10_12_000000_create_users_table (0.02 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated:  2014_10_12_100000_create_password_resets_table (0.01 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated:  2019_08_19_000000_create_failed_jobs_table (0.01 seconds)
Migrating: 2019_11_30_194301_create_plant_images_table
Migrated:  2019_11_30_194301_create_plant_images_table (0.01 seconds)
Migrating: 2019_11_30_200146_create_media_table
Migrated:  2019_11_30_200146_create_media_table (0.02 seconds)
Migrating: 2019_12_01_122437_create_jobs_table
Migrated:  2019_12_01_122437_create_jobs_table (0.01 seconds)
```

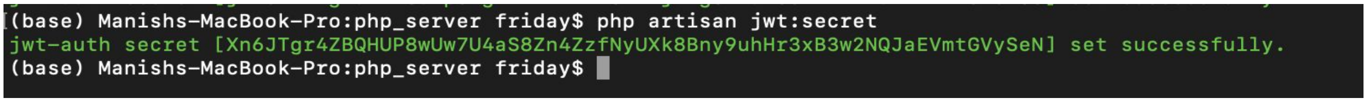
```
$ php artisan db:seed
```



```
(base) Manishs-MacBook-Pro:php_server friday$ php artisan db:seed
Database seeding completed successfully.
(base) Manishs-MacBook-Pro:php_server friday$
```

6. Set JWT secret

```
$ php artisan jwt:secret
```

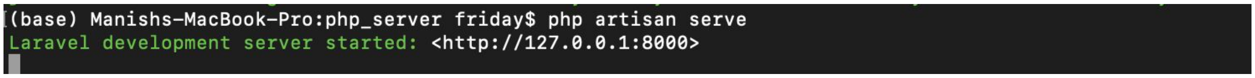


```
(base) Manishs-MacBook-Pro:php_server friday$ php artisan jwt:secret
jwt-auth secret [Xn6JTgr4ZBQHUP8wUw7U4aS8Zn4ZzfNyUXk8Bny9uhHr3xB3w2NQJaEVmtGVySeN] set successfully.
(base) Manishs-MacBook-Pro:php_server friday$
```

7. Run the server and Run the queue in separate consoles

```
$ php artisan serve
```

```
$ php artisan queue:work --tries=2
```



```
(base) Manishs-MacBook-Pro:php_server friday$ php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

Backend(Plantinator) Installation

1. To install dependencies run the following command

   ```
   $ pip3 install -r requirements.txt
   ```

2. Download the following files from the given link and put them under ml_files folder
   - 2.1. File no 1 - cnn_models.pkl
   - 2.2. File no 2 - transformation.pkl

3. To start the server on port 4000 run

   ```
   $ python3 manage.py runserver 4000 --nothreading
   ```

Due to time constraints, we were not able to make our products more manageable, we planned to make the product within the standards of Software specifications, so

1. If the developer is running the PHP server on some other port say 1337, in order to successfully connect PHP API to the react app, the developer must update the port or hostname in src/helpers/api.js
2. If the developer is running the Python server on some other port say 5000, the developer must update the port in App\Jobs\PredictPlantDisease.php. And must restart the queue.
   a. The queue can be restarted by simply causing the interrupt in the terminal where queue:work was executed and executing the cmd again.
   b. Or by the cmd

      ```
      $ php artisan queue:restart
      ```