**“Farmeasy”**

**An Online Farm Equipment Renting Platform**

**PG – DAC**

**200250120082 – Sagar Nishad**

**200250120110 – Vishal Gupta**

**Project in brief-**

* ‘Farmeasy’ provides farm equipment renting platform to the farmers who can not buy or afford them.
* Through this platform one who have equipments can put it on rent and others who need it can rent it from here.
* This will provide users easy registration, booking and avail services at doorstep from service providers near them.
* To use this platform, one needs to have an account over here. The user will signup first to get started.
* If he’s any equipments to put on rent, he can register by providing all the necessary information.
* Once the service provider is registered, the needy people can search for the services they want for their location.
* Then it can be booked by searching for slot availability. If one is just going through, wishlisting can also be done.
* If slots are available, simply by making payment, the booking will be done.
* Service provider will look into it and reach you on time.
* If some service provider has more than one service, he can put them all and if wanted a service, can get here.
* ‘Farmeasy’ would bring tooling cost down by **20%** and increase its utility, so the profit, by **3** times than usual.
* **0.1%** of farming population(**150mn**) can get employed with this, would contribute the IT sector as well, and keep **‘Make in India’**, **‘Digital India’** get going! Farm’easy!!!

**Working of the Project-**

Let us go through booking scenario of project-

1. **Presentation Layer-**
   1. When you search for a service, on click of a button, the values in textbox will be read and the object will be passed to service layer through an ajax call.
   2. Service returns the list if available. While booking, you need to check for availability by putting dates.
   3. If available then go to payment and by making payment, get your service booked.
   4. If not available, then you have to search for another dates, to book the service.
2. **Service Layer-** 
   1. If URI matches, you’ll get into controller which will call service method ‘checkavailability’.
   2. List of bookings will be checked for slot availability for that service provider.
   3. If available, user will go to payment. On payment, ‘dobooking’ method will be called.
   4. On successful payment, user booking will be done.
3. **DAO Layer-**
   1. In booking repository, we will query to get bookings already present. Send this list to service for checking and avoiding overlapping of bookings.
   2. If slot is available, booking will be allowed.
   3. On payment, this booking will be saved into bookings.

**Problem solving-**

1. Composite key-
   1. We wanted composite keys to avoid duplicacy, which was a little tedious.
   2. We searched a lot, found ‘embeddedId’ way and ‘Idclass’ way out of which we went for ‘Idclass’ way.
   3. Idclass is a class of all the attributes that form composite key.
   4. While mapping, ‘unsupported media type’ error stuck us a lot. It was coming due to mapping issues which we solved with the help of our project guide.
2. Spring security-
   1. It secures the service layer by filtering all the request coming to it.
   2. We needed to handle cross origin, and antmatchers so that only desired requests will be allowed.
   3. Password encryption was sometimes creating issues, so used online tools to decrypt it and see if our password matches the encrypted password.
3. Session handling-
   1. We were using history state to pass data along with route, which was on refresh losing the data from memory.
   2. We then used session storage to get the functionality.
   3. Session expires on logout so the security issues were solved as well.
4. Unhandled things-
   1. We logged the request and response data into logfile, then we came to know some null values, illegal values that were causing issues.
   2. Sometimes, ‘concurrentModificationException’ came, we changed the code accordingly to avoid concurrent modification.
5. Session expiry handling-
   1. Whenever session gets expired, we wanted user to get logged out.
   2. We did it by checking session data. If session has expired, data in it would also be cleared.
6. Other errors-
   1. java.util.Optional was putting an error in ‘get()’ method for null values. To handle this null, we used ‘isPresent()’ method which on null returns false.

**References helped to rectify these issues-**

1. [https://stackoverflow.com/](https://stackoverflow.com/%20)
2. <https://dzone.com/>
3. <https://www.baeldung.com/>
4. <https://vladmihalcea.com/>
5. <https://www.techgeeknext.com/>
6. <https://www.c-sharpcorner.com/>
7. <https://angular.io/>
8. <https://www.npmjs.com/>
9. <https://itnext.io/>

**Learnings during the project-**

1. During the entire project, we came across a lot of problems, that taught us how to be calm and patient, to solve the problems effectively.
2. We created high level design plan, and tried to go through it. It minimized confusions to almost zero.
3. Before going through any scenario, we did small proof of concepts, that taught us how should be the approach.
4. We did market study, that helped us design the presentation that catches the attention of the customer. So, we learnt various aspects of design and developing.
5. We visualized the scenarios on which we are going to handle the events, that led to the betterment of the solution.
6. We burnt the fingers on coding a lot of scenarios, thinking on logic, improving it, that helped us upgrade our technical skills.
7. We learnt, how to hunt down the solution for various problems, that came across, while doing the project.
8. We used github for version control, followed agile way of doing the project, by maintaining scrum board. This brought us exposure to industrial standards.