Good morning to all Cotiviti family, I am Saroj Sharma and today I am going to give a presentation about a not fully developed but simple eCommerce project. We named this project as Pasal. The main objective of this application is to provide ease of buying and selling products for both customer and supplier respectively.

Talking about the framework for building this project, we used Angular on the front end and Spring Boot on the backend side. I would like to give a quick introduction about this framework. Angular is an open framework and platform for creating Single Page Applications, written in TypeScript.The framework consists of components, template, services etc.Each component class holds the application logic and data.When data or logic that has to be shared across components, a service class is created.Now talking about the architecture of spring boot it contains controllers, repository and services.Controllers contain application logic and it passes user input data to service.The Service is the middleware between controller and repository. It gathers data from controllers, performs validation and business logic, and calls repositories for data manipulation.Repositories layer is for interaction with models and performing database operations.

Moving on to the project first we need to set up the database, and this can be done easily with Spring Data JPA.Some information is written in application.properties file to set up the connection to the H2 database.In Pom XML file required dependencies are added.

In the security Config file required security configuration is set up.In filter chain class antMatchers() is used to configure the URL paths from which the Springboot application security should permit requests based on the user's roles.

Now moving on to authentication the next step is to create a User Model. They are also called Entities and are annotated by @Entity. We Autowired LoginService for dependency injection and @PostMapping annotation to specify the type of HTTP requests accepted.In login service, the authenticate method of authentication manager Interface returns an instance if it can verify that the input represents a valid principal.Otherwise throw AuthenticationException if the input represents an invalid input. SecurityContext is used to store the details of the currently authenticated user and the SecurityContextHolder is a helper class, which provides access to the security context.Then Login request is encoded to generate a token and it is sent as response back to the client.In client side, this response object is saved on local storage for authentication and authorization.Different role is assigned to the user.Based on this role he/she can or can't perform certain operation.For example customer only can buy products where as supplier can sell his/her products.For communication between client and server, data transfer object(DTO) is used to avoid exposing sensitive information.

Talking about the products,To display a list of products, a product list api is created and used to fetch the data from the product table database and displayed on the home page. Likewise, to display details of product, Detail button is provided and when user click on detail button it navigate user to the detail page where he/she can see detail information of product.Also they need to specify the number of items count and then Order Now button is clicked to place order.

If anyone hits the wrong URL then the not found page is displayed.Thank you this is my part Now I would like to call Sonam for next.