**Placement Test for Backend Developer (NestJS)**

Introduction:

Boost Solutions L.L.C., based in Damascus, Syria, is a digital transformation agency specializing in high-quality, cost-effective ICT solutions. Boost focuses on enhancing business efficiency, streamlining organizational processes, and driving profitability through services like web development, e-commerce, digital marketing, and online reputation management. Committed to client success, Boost provides tailored strategies that adapt to diverse industry needs, aiming to deliver impactful and scalable solutions. For more details, visit their official website at [Boost Solutions](https://boost-sy.com/).

Rule:

* This test time starts when the email is sent to you, and you should finish the mini-project in max 48 hours, less is better and qualifies for more points.
* You should make GitHub repo, and take new branch from main and push all your work on this branch before finish the time.
* The project should contain a readme file, you should write all your work and your note.
* It's energetically finish before the last time to work, and this give you more point.
* After you finish you must send the repo link and commit name as a replay to the email

# Introduction to E-Store for Clothes

E-Store, an ultimate destination for fashionable clothing that combines style, comfort, and affordability. it offers a diverse range of apparel for men, women, and children, ensuring everyone can find something to express their unique personality. the collections feature the latest trends and timeless classics, catering to various occasions. With user-friendly navigation and secure checkout.

# Requirements:

1. Build a simple ERD contains the model to build previous app.
2. A product should reference to one category.
3. The product's price change based on Type (Crewneck, hoodie), and color (Black, white).
4. The admin can manage the product, type, color.
5. The customer can search about products base on (name, color, type, price range)
6. Customers can change order of the previous result, based on (name, price, createAt)

## Assignment:

1. Build the previous app, with high quality, and speed of response.
2. Add Auth for user, using JWT.
3. Add pagination.
4. Add all API to manage User.
5. Add all API to manage Variant.
6. Add all API to manage Product.
7. Catch and handle all possible exception.
8. Add Swagger API document
9. Add logger for API.
10. Apply all validation and transformation needed.

## Options:

* Use any ORM framework (justify your choice).
* Use any project architecture (justify your choice).
* Apply the most needed security levels you suggest (justify your choice).

## Extra features:

1. Separation of environments [dev, test and prod]
2. Use of validation.
3. Use of authentication and authorization.
4. Use of global Exception handling.
5. Use of global logger
6. Use of Unit Testing.