

**Pre-Interview Selection Task for Full Stack Developer (Intern) for Meydit**

Our client, Meydit, is a double-sided online marketplace based in Sydney that brings together consumers looking for tailor-made clothes with thousands of independent tailors or makers, primarily those who do not have a shop.

This internship is 99% hands-on, intending to create and refine user stories and acceptance criteria to be part of a team to launch an MVP (Minimum Viable Product) within the next 3 months.

As an intern, your priorities will be to learn the following but not limited to:

- Steps and methods in developing web-based applications using a modern full-stack architecture based on micro-services, React/NextJS front-end and Node/AdonisJS backend and in a real-world project.
- Develop fluency in operating JIRA (and Trello) software
- Operational fluency within Agile / SCRUM methods; and processes involved in transitioning from the KANBAN method to the SCRUM Method.
- Fluency in the business processes related to double-sided digital marketplaces and proprietary methods and processes of Meydit.
- Gain feedback through pairing with a senior experienced Business Analyst and get regular feedback on your approach to each task, review your deliverables, and ask for critical feedback on your overall progress.
- A first-hand account of Development Operations of the development of an MVP, a critical stage in the overall approach to modern SDLC

As an intern, you would be engaging with the

- Founder and CEO Susan Hansen,
- Product Owner(s), Agile Business Analyst
- Collaborate with a full-stack development team of the modern technology stack of React/NodeJS and AWS Environment.
- Business stakeholders, including creatives, designers, and consumers (Limited primarily to BAs)

Essential Tools, Frameworks,

- React JS, Next JS, Axios, Material UI
- NodeJS, Adonis JS Framework including Lucid ORM and MVC
- AWS EC2, S2 Buckets, Elastic Beanstalk,
- AWS RDS Postgres DB
- Jira (new) and Trello (getting decommissioned) for User Stories
- Figma (UI UX Prototypes)
- Visual Studio Code and extensions such as ThunderClient.
- Github
- Vercel Cloud (frontnd)

**Submission Details:**

- Use the mini-project brief to create and deploy a full-stack application that meets the following criteria.
  1. Use Node JS / Adonis JS framework to create several Rest APIs,
  2. Use React UI for the front end.
  3. Publish the app in GitHub.
  4. Deploy the app in any of the Cloud platforms (e.g. AWS, Vercel, Azure)
- Create a document with the following.
  - Links to GitHub (Code needs to be accessible)
  - Link to the URL for the front-end
  - List of URL, routes for the Rest API and the expected JSON input
  - Write down any assumptions to make
  - Write any questions you have.
  - Write a paragraph about your key learning from this exercise. Keep this in bullet points.
  - Write a paragraph about the areas you want to learn further and focus on during this internship.
- **Submit your document to Jess Kaur, Recruitment Specialist, LVT People, through LinkedIn Direct Message, or email on [jess@lvtpeople.com.au](mailto:jess@lvtpeople.com.au)**

*TIP: You may need to send a connection request before sending the message.*

<https://www.linkedin.com/in/jessykaur/>

- You may also contact Susan Hansen, Founder / CEO of Meydit, for questions related to the business requirements <https://www.linkedin.com/in/suhansen/>

*Nothing below represents the actual situation or requirements. The platform, or minimum viable product, is still under development. The beta release is expected to be launched within the next few months.*

## **Situation**

Meyd.it is a digital, double-sided marketplace that connects consumers looking for tailor-made clothes from makers who often independently work from home and not necessarily full-time.

Thousands of skilled tailors and designers in Sydney alone are willing to take on gig-based work.

Consumers (may also be called buyers or clients) can register for an account for free, create a “job” along with inspirations, publish the job (or may also be called “making”), get quotations, accept quotations, make payments upfront for the entire project, acknowledge receipt of goods, and provide feedback on the markers or the outcome of their makings.

The Markers can create an account for free, update their Ateliers, find and bid for jobs, update the accepted job status, and provide delivery details for shipped items.

The business, Meydit, would take 15% of the total cost of the job.

## **Programming Task**

### **User-Story #1**

As a Consumer, I want to go to a page, post a job and get notified with the quotes.

### **Acceptance Criteria**

1. The consumer must be able to specify their first name, last name, phone number, and email address, address (including postcode and state), select the types of clothing (e.g. Dress, Ethnic Wear - Sari / Blouse), upload several images representing the type of clothing they want to get made, enter a description of their making in plain text, and specify a budget (optional).

### **User Story #2**

As a Maker, I want to list all the available jobs, filter jobs by location and types of clothing, see the total count of quotations for jobs, and send a quote.

### **Acceptance Criteria:**

1. The maker must be able to list all jobs and view information such as the types of making, making, status, count of quotations, location, etc.

## **LVT People**

2. The maker must be able to select a specific job and see all the details the Customer provides.
3. The maker must be able to send a quotation, including a price and any other comments; the system would notify the customer by email.

## **Considerations for Implementation**

### **DevOps**

1. Consider installing the VS Code extension for Adonis  
<https://marketplace.visualstudio.com/items?itemName=jripouteau.adonis-vscode-extension>

### **AWS Cloud**

You may need to create either a trial account or a free-tiered account. Consider some of the following services:

1. Use AWS RDS Postgres DB
2. Use either EC2 instance (e.g. T2) or Elastic Beanstalk to deploy your backend and front end.

### **Backend**

1. Write in Typescript
2. Use the Lucid ORM.
3. Create Lucid data models <https://docs.adonisjs.com/guides/models/introduction> and use the Adonis JS to implement the CRUD operations.
4. Utilise Validators <https://docs.adonisjs.com/guides/validator/introduction> to enforce data rules.
5. Use any supported database with the Lucid ORM (PostgreSQL, MySQL, MSSQL, MariaDB and SQLite), and leverage the built-in migration system.

### **Frontend**

1. Use React JS
2. Use the Axios library to connect to the Rest APIs
3. Optionally, consider using Material UI