Problem Statement:

You are working on a project that analyzes crypto currencies data and the requirement is to process historical price list of currencies provided by a currency exchange. A sample data set below shows prices of BTC, ETH and LTC crypto currencies at various times on 7th May:

Table 1:

Currency	Date	Time	Price
BTC	20180507	0915	34.98
BTC	20180507	1045	36.13
BTC	20180507	1230	37.01
BTC	20180507	1400	35.98
BTC	20180507	1530	33.56
ETC	20180507	0900	1.45
ETC	20180507	1030	1.87
ETC	20180507	1245	1.55
ETC	20180507	1515	2.01
ETC	20180507	1700	2.15
LTC	20180507	0930	14.32
LTC	20180507	1115	14.87
LTC	20180507	1245	15.03
LTC	20180507	1400	14.76
LTC	20180507	1700	14.15

You need to read the data and find out the best profit that would have been possibly made by buying a currency at a given price and selling it later on the same day. Display the profit for each currency separately. For the sample data above, following could be an example output for the 3 currencies:

07-May-18	
BTC	
Buy	**Sell**
$34.98	$37.01
09:15AM	12:30PM
Profit: $2.03	

07-May-18	
ETC	
Buy	**Sell**
$1.45	$2.15
09:00AM	05:00PM
Profit: $0.7	

07-May-18	
LTC	
Buy	**Sell**
$14.32	$15.03
09:30AM	12:45PM
Profit: $0.71	

Preferred Technologies to be used:

The position for which this challenge is being presented has a mix of these tech stack: ReactJS, NodeJS, TypeScript/ES6, MongoDB, Oracle, Java, AWS etc.

Development Notes:

- The goal of this coding challenge is to provide the candidate an opportunity to showcase their expertise in following areas:
  - Ability to retrieve data from a database or by calling an API.

- Ability to process the data or build efficient computational logic using a library/framework best suited for the task OR just using the inherent language features.
- Ability to display the processed data in the required format using a suitable frontend library/framework.

- Depending on your expertise and experience in certain areas you can choose to put more effort there. For example, if you are more comfortable in backend/database/API side, you can choose to simplify the presentation logic in favor of an elaborate backend design.
- You can come up with your own UX for displaying the output, while ensuring the key details are not missed out. Recommended output would be a web page using React components, but you can choose another framework as well.
- Feel free to make assumptions if something is not clear and specify those assumptions in code comments.
- For backend part, you can read the data from a database OR a JSON file OR even build an API. Use the following details as suggestions, but not requirements:
  - If you are working with Relational Database, you can assume it to have similar schema as Table 1.
  - If you are working API, NoSQL DB or JSON file, the JSON can be assumed to be structured as:


```
[
  {
    "currency":"BTC",
    "date": "20180507",
    "quotes":
      [{"time":"0915", "price":"34.98"},
       {"time":"1045", "price":"36.13"},
       {"time":"1230", "price":"37.01"},
       {"time":"1400", "price":"35.98"},
       {"time":"1530", "price":"33.56"}]
  },
  {
    "currency":"ETC",
    "date": "20180507",
    "quotes":
      [{"time":"0900", "price":"1.45"},
       {"time":"1030", "price":"1.87"},
       {"time":"1245", "price":"1.55"},
       {"time":"1515", "price":"2.01"},
       {"time":"1700", "price":"2.15"}]
  },
  {
    "currency":"LTC",
    "date": "20180507",
    "quotes":
      [{"time":"0930", "price":"14.32"},
       {"time":"1115", "price":"14.87"},
       {"time":"1245", "price":"15.03"},
       {"time":"1400", "price":"14.76"},
       {"time":"1700", "price":"14.15"}]
```

```
  }
]
```

- Ensure the prevalent best coding practices are followed and write clean code.
- Demonstrate TDD approach was taken during the development.
- Commit the solution to Github and share the URL. Make sure the project is not private in git and is accessible to reviewers.
- If you are successful in obtaining a 1st round interview, please come prepared to discuss your solution/implementation as this will form a part of the interview process.

Stretch Objectives:

If the challenge is simple enough for you, you are encouraged to optionally implement additional features or elaborate on your solution. Few suggestions are:

- Assume you have historical data for more currencies and for more than one day, then implement a form that allows filtering based on date and/or currency.
- Implement/enhance the API that retrieves filtered data based on the parameters passed while calling the API.
- Incorporating AWS skills in the solution will be highly favored.

All the best.