**Historical Bicoin**

**Problem Statement (Full Stack – UI, Service + CI/CD)**  
  
Develop, Test and Deploy a service to find the historical Bitcoin price along with the highest & lowest price markers for a user provided Start date, End date and Output Currency along with the default USD value (Hint: Conversions can be done based on the current rates).  
• End user should be able to view results by changing the input parameters  
• The service should be ready to be released to production or live environment  
• The service should be accessible via web browser or postman (using any one of JavaScript frameworks, HTML or JSON)  
• The solution should support offline mode with toggles  
• The service should return relevant results as expected, even while the underlying dependencies (Ex: Public API) are not available!  
(Use your own code/logic/data structures and without 3rd party libraries or DB)  
  
**API Data Sources  
APIs**  
<https://api.coindesk.com/v1/bpi/historical/close.json>  
<https://api.coindesk.com/v1/bpi/supported-currencies.json>  
**Documentation**: <https://www.coindesk.com/coindesk-api>  
  
**Expected output**   
(via an UI mechanism of your choice – Ex: React page)  
Date DD-MM-YYYY e.g. 01-01-2018 for each date in the range  
Price along with text to show  
if it’s the highest in date range i.e. Price (high) e.g. 1234.5678 (high)  
if it’s the lowest in date range i.e. Price (low) e.g. 1001.343 (low)  
  
Currency e.g. INR & USD  
  
**NFRs**   
• Demonstrate SOLID, 12 Factor and HATEOAS principles, Design Patterns in the design and implementation  
• Demonstrate Performance, Optimization & Security aspects  
• Demonstrate Production readiness of the code  
• Demonstrate TDD & BDD & Quality aspects  
• Demonstrate sensitive information used in the Micro Services such as API keys are protected / encrypted  
  
**Documentation**   
• Include the open-API spec./Swagger to be part of the code. Should be able to view API via swagger (Documentation to explain the purpose of the API along with Error codes that the service consumers & client must be aware of!)  
• Create a README.md file in the repository and explain the design and implementation approach  
• In the README, add a sequence diagram or flowchart created using draw.io – https://www.draw.io  
• Mention the design patterns used in the code  
  
**Build & Deploy**   
CI  
• Build CI/CD pipeline for your project(s); Pipeline scripts need to be part of the codebase;  
• Ensure the Jenkins job config., scripts are a part of the project sources  
CD  
• Demonstrate the service deployment using a Docker container (Create a docker image and publish service locally)  
• Ensure the docker files are a part of the project sources