

# HACK-IT!

AT OCBC

## Things to Prepare:

- Please rename yourself on Zoom:  
**Track\_Name**
- Please stay on Zoom throughout the session
- Turn on your camera throughout the session
- You can reach out to us on Zoom chat if you are experiencing any issues

```
Program [Calling OCBC Digital Experts!]
```

```
{  
  If ["Digital" + "Expert"]  
  Output: ["You are the CODE we are hiring for!"]  
}
```

# HACK-IT!

AT OCBC

Welcome Address By

**Kenneth Tan**

**OCBC Bank**  
Head of Resourcing Centre

```
Program [Calling OCBC Digital Experts!]  
{  
  If ["Digital" + "Expert"]  
  Output: ["You are the CODE we are hiring for!"]  
}
```

# HACK-IT!

AT OCBC

## Agenda

- 5.00pm Welcome address by Kenneth Tan,  
Head of Resourcing Centre
- 5.10pm Next steps into final assessment
- 5.15pm Assignment Briefing - Developer Track
- 5.25pm Assignment Briefing - UI/UX Track
- 5.35pm Assignment Briefing - Data & AI Track
- 5.45pm Q&A Session
- 5.55pm Closing
- 6.00pm End

Output: ["You are the CODE we are hiring for!"]

# Final Assessment



## Data & AI

---

### RELEVANT POSITIONS

Data Scientist

Software Developer

Digital Product/Project Management

Data Analyst



## Developer

---

### RELEVANT POSITIONS

Full Stack Developer

Mobile Developer

Technology Analyst



## UI/UX

---

### RELEVANT POSITIONS

User Experience Designer

Product Developer

# Final Assessment

## Things To Take Note



**01**

You are given **14 days** window to complete the assessment, submission will close on **13 September, 11.59pm**

**02**

The submission requirements and judging criteria will be unique to tracks attempted

**03**

You are required to complete the assessment **only on your own effort**

**04**

You may attempt more than one assessment if time allows, priority will be given to assigned track

**05**

Every submission will be assessed by **3-4 technical experts** under Indorse.io network

**06**

**Assessment report** will be shared at the end of the HACK-IT! AT OCBC Singapore campaign

**07**

Top candidates from the assessment will move forward to the **final job interview stage**

**08**

Top submissions ranked 1, 2 and 3 from each track will be rewarded with **SGD3000, SGD2000 and SGD1000**



Developer

## Assessment Details



### Scope of Assignment

- Create a web application that allows users to book a seat in a movie theatre - no authentication required.
- This assignment aims to create a fully functional web application with Backend in Java and Front-end in related technologies such as ReactJS.
- Alternatively, you could use a Java MVC framework for creating both the Backend and Frontend.

### App Functionality

- Display all the seats in a theatre and allow users to book them.
- If a user clicks a seat that another user booked, they should get an error. You must handle the concurrency scenarios and avoid data inconsistency.
- If a seat is available, the user should be asked for their details like name, email ID, etc. and email them with a confirmation.
- You do not need to gather the payment details.
- The solution should have a single web page with the seats displayed in a grid (you could start by a smaller number of seats, maybe 20-30)



Developer

## Assessment Details

### For Mobile Developer Applicants

Instead of using ReactJS for the front-end, you will need to create a Mobile front-end using C#, Kotlin or Swift. Backend APIs need to be done in Java.

SCREEN 3

SEATING KEY:

- Your seat
- Occupied seat
- Standard seat
- Not Bookable seat
- VIP seat
- The Recliner
- Wheelchair space



### WHERE TO SIT?

[Add/remove tickets](#)

J9

1 of 1 seats selected

Total (incl. £0.79/€0.79 booking fee per ticket): £4.99

DONE



Developer

## Assessment Details



### Requirements

- Hosting a live solution is encouraged and will increase the chance of a higher score for this assignment.
- Test coverage is essential.
- Complete installation instructions for your application must be available via README.md.
- Please create a new Git repo where you will upload the relevant project files. Please properly document your code.





Developer

## Assessment Details

### Evaluation Criteria



#### 1. Code Quality (20%)

Code Quality, also known as Software Quality, is generally defined in two ways:

- How well does the code conform to the functional specifications and requirements of a project?
  - Structural quality, which relates to the maintainability and robustness of the code
- Please make sure that your assignment code conforms to these standards.

#### 2. Documentation (20%)

Documentation is essential for your role in the banking environment because you would be working on projects and codebase, which is quite large and extensible. Since hundreds of other professionals work on the same project, documentation becomes the bedrock of collaboration and teamwork.

#### 3. Security (20%)

As a part of the OCBC team, you will be working on mission-critical banking projects. You will be dealing not just with the customers' data but also with their financial data. This means that software security becomes a first-class citizen in whatever you are designing and building. Your assignment submissions should reflect that and focus on building secure applications - right from infrastructure security to ensuring that your code does not inadvertently release or leak any sensitive information.

#### 4. Extensibility (20%)

The applications and the software you develop will be worked upon in the future by other colleagues within the bank. That means the design of the software has to be extensible from the ground up. While completing this assignment, you need to make sure that adding new functionality or modifying existing functionality is straightforward and does not impair the application's working.

#### 5. Test Coverage (20%)

Since software security and extensibility are the cornerstones of your application, test coverage becomes extremely important to ensure that the security is not compromised when adding new features to the application. While striving for 100% test coverage would be ambitious, at least try to cover the most critical aspects of your application with end to end test cases.



UI/UX

## Assessment Details



### Scope of Assignment

- Create Hi-fi mockups of a mobile application that allows users to book a seat in a movie theatre, with authentication required (email and password) before finalising the booking
- This assignment aims to create a fully functional mobile app mockup

### App Functionality (expected but not limited to)

A work in progress design is what we are looking for here; however, the idea of this challenge is to get some insight on how you work, your design ability and see how you realise your ideas.

#### User Flow Steps:

- A user selects a movie to book
- A user chooses a seat(s)
- A user finalises the purchases
- A user provides personal/ booking details ( You don't need to gather payment details)
- A user completes the transaction and gets confirmation

#### Points to Consider:

- You can assume that the movie theatre has 20-30 seats
- Edge case - another user tries to book the same seat that a different user is trying to secure.



UI/UX

## Assessment Details



### Requirements

- Please produce and send a high-fidelity mockup of your design in Sketch (or your preferred program if you don't currently use Sketch)
- [Bonus Points] Please provide comments about any assumptions/shortcuts you make
- [Bonus Points] If you manage to conduct preliminary user research or test your mocks with actual users



## Assessment Details

### Evaluation Criteria



#### 1. Problem Analysis (25%)

As a UX designer, you need a deep understanding of the user problem.

You need to think from a user's perspective, empathise with their problems, and then come up with the best possible solution.

#### 2. Interface Design (25%)

You need to be aware of the design principles and design trends to work in a competitive environment like that of a renowned bank.

The interface should look modern, sleek while at the same time also consider the various security requirements of a banking institution and its clients.

#### 3. User Experience (25%)

You should demonstrate a keen interest in maintaining an easy to use and intuitive experience from an end-user perspective while keeping accessibility and usability at the forefront.

#### 4. Presentation (25%)

The aim is to see how you approach a user problem and come up with ideas and solutions.

There aren't any right answers; you will be evaluated on your thought process rather than the quantity of your output.



### Scope of Assignment

- Use the [US Traffic 2015 Dataset](#), publicly available on Kaggle, to visualise the traffic patterns.
- This assignment aims to clean and analyse the dataset, create appropriate models and visualise them using the proper software.

### Functionality

- Use the appropriate algorithms and models to find out the top 5 most obvious patterns from this data.
- Support your hypotheses with appropriate data

### Requirements

- Submit a Jupyter Notebook with the data models
- Good commenting and Documentation is essential
- Explain why you chose this particular model for solving the problem
- Create a new Git repo where you will upload the relevant project files.



## Assessment Details

### Evaluation Criteria



#### 1. Code Quality (20%)

Code Quality, also known as Software Quality, is generally defined in two ways:

- How well does the code conform to the functional specifications and requirements of a project?
- Structural quality, which relates to the maintainability and robustness of the code

#### 2. Creativity in solving the problem statement (20%)

Creativity speaks volumes about your capability to make sense of given data, derive tangible results relevant to an organisation's business needs, and present the findings. All this while keeping in mind the problem statements.

#### 3. Feature Engineering (20%)

Feature Engineering refers to the process of selecting and transforming variables when creating a data model for a given problem statement. While you will be given a public dataset related to the problem statement, you need to create “features” that make the models and algorithms work as intended. You can use standard features, including open-source implementations, or create your features -- or learn features automatically during training.

Note that your code needs to be self-contained, i.e. it should automatically create your desired features, which can be used in evaluating the Hold-out test set.

#### 4. Model Performance (20%)

Model performance determines how a model represents the data and how well the chosen model will work. Please make sure that your models are universally compatible with various machines and operating systems.

#### 5. Documentation (20%)

Documentation is essential for your job in the banking environment because you would be working on projects and codebases that are pretty large and extensible. Since hundreds of other professionals work on the same project, documentation becomes the bedrock of collaboration and teamwork.

# Submission Guidelines



## Steps to submit:

- We will provide you with a form to submit your solution. The form will include
  - Relevant personal details (Name, Email Address)
  - Chosen track
  - Link to the hosted solution - Please make sure to have the view (GitHub / Google Drive) set to public so that anyone with the link can access the solution.

## Important:

- For all the tracks - Make sure your submission folder or Git Repo has accompanying document or README that entails how to navigate your code / mocks / Jupyter notebook.
- For Developer track - If you've hosted your solution somewhere (Netlify / Heroku / Your own server), make sure to add the publicly accessible link to README.

## Next Steps



### **Following this session :**

- Padang & Co will be sharing across the session recording link and the briefing deck to all the candidates latest by today.

### **Assessment :**

- You are given 14 days window to complete the assessment, submission will close on **13 September, 11.59pm**

### **For any questions :**

- Please email Divakar, [divakar@padang.co](mailto:divakar@padang.co)

**All the best!!**