**PayFast Security:**

* For our payment gateway, we make use of PayFast – an online payment platform that allows users to make secure payments.
* In particular we make use of the sandbox feature on PayFast which is an isolated test setting that lets users make payments without the actual transfer of funds.
* The reason why we make use of the sandbox feature and not use the actual PayFast system, is because PayFast charges a percentage of each transaction plus R2, every time a person makes a transaction.
* The sandbox feature is currently free, and is implemented the same way as the actual PayFast system would be implemented. Hence very little changes need to be made to our code when transferring from the sandbox environment to the actual PayFast system.
* PayFast ensures that the payment gateway is secure by being PCI-DSS Level 1 Compliant.
* PCI-DSS Level 1 Compliant:
  + There are 4 levels of PCI compliance, with level 1 being the highest and strictest.
  + Through a set of specifications established by the PCI SSC, PCI compliance certification ensures the security of card data.
  + Organizations must take a number of steps to verify their adherence to the PCI DSS. Which are:
    - A yearly report on compliance (ROC) from a certified internal security assessor or security assessor;
    - A yearly network scan performed by an authorized scan vendor (ASV);
* PayFast makes use of extended Validation SSL
* They perform AVS scans on a weekly basis – weekly network scans to search for fresh potential vulnerabilities, and quarterly certification.
* They make use of BIN/IN Validation - checks the card-issuing bank locations, and compares it with the user’s current location. Some businesses may decide to reject/accept payments if the locations don’t match or if payment is being made from a specific nation.
* To identify and prevent malicious activity before it reaches their systems, PayFast uses Web application firewall technologies.

**Hashing of Passwords:**

* To hash passwords we make use of the bcrypt library, which is based on the Blowfish cipher that uses symmetric-key block cipher (same key is used for encryption and decryption, and it is not encrypted byte by byte but rather block by block).
* bcrypt is an adaptive function that, in addition to including a salt to prevent rainbow table attacks, it keeps its resistance to brute-force search attacks even as computing power increases.

**Encrypting Local Storage:**

* We make use of Expo Secure Store when encrypting the local storage.
* Key-value pairs may be encrypted and safely stored locally on the device with the help of expo-secure-store. Each Expo project has its own storage system and is not permitted to access the storage of other Expo projects.
* The majority of the information kept by Expo, including credentials, is secured at rest by their cloud service provider, Google Cloud. Additionally, Key Management Service is used to encrypt credentials.