1. Setting up the database
   1. In MySQL Workbench, initialize the database by running GenerateDatabase.sql
   2. To edit the name of the database, replace Line 1 with Create Schema <Database Name> and Line 2 with use <Database Name>
2. Setting up the application
   1. Download and Open the project folder in an IDE (IntelliJ or VSCode)
   2. In VSCode
      1. Ensure Java Extensions Pack is installed
      2. Navigate to the project folder in VSCode
      3. Open AhoyApplication.java
      4. For initial build, press F5 to debug and build the project
      5. After debugging, launch the project with F5
   3. In IntelliJ
      1. click Open Project: “your directory”/cs102-main/ahoy/pom.xml  
         \*pom.xml is the project file that helps IntelliJ recognise it as a project
      2. In ahoy/src/main/java/com.ahoy.ahoy/AhoyApplication click a green arrow that appears on the left side of the code
3. Configuring the application
   1. To configure the port the application is run on, open application.properties and replace Line 1 with server.port = <port number>
   2. To configure the database
      1. Replace line 4 with: spring.datasource.url=jdbc:mysql://localhost:<Database Port>/<Database Name>
      2. If there is an error with the timezone, replace line 4 with: spring.datasource.url=jdbc:mysql://localhost:<Database Port>/<Database Name>?useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTimezone=UTC
      3. Enter your MySQL Username and Password under spring.datasource.username and spring.datasource.password respectively
   3. To configure the timings of API Calls and Emails:
      1. By default, the first API is called once a day at 8am. To change the timing it is called, edit the CRON timing in application.properties named post.timing
      2. The second API is called twice a day at 8am and 8pm. To change the timing, edit get.timing
      3. The alerts for each user are emailed out to them twice a day at 9am and 9pm. To edit the timing , edit email.timing
   4. To configure email (Gmail account)
      1. Set your email account: spring.mail.username = “Your Email Address”
      2. Set your email application password: spring.mail.password = “Your Email Application Password”  
         \*Please note, this is not your email login password, but a generated app password (read [here](https://support.google.com/accounts/answer/185833))
4. Open “http://localhost:8080/” in your web browser to access the Application