Testing and Code Review

1. Change History

Change Date	Modified Sections	Rationale
Nov 12	All	Initial document creation.

2. Back-end Test Specification: APIs

2.1. Locations of Back-end Tests and Instructions to Run Them

2.1.1. Tests

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
POST /api/auth/signup	`backend/tests/no-mocks/auth.routes.test.ts#L76`	`backend/tests/with-mocks/auth.routes.test.ts#L98`	None (uses real OAuth2Client and DB)
POST /api/auth/signin	`backend/tests/no-mocks/auth.routes.test.ts#L115`	`backend/tests/with-mocks/auth.routes.test.ts#L576`	None (uses real OAuth2Client and DB)
POST /api/auth/select-role	`backend/tests/no-mocks/auth.routes.test.ts#L154`	`backend/tests/with-mocks/auth.routes.test.ts#L821`	None (uses real DB)
POST /api/jobs	`backend/tests/no-mocks/jobs.routes.test.ts#L92`	`backend/tests/with-mocks/jobs.routes.test.ts#L156`	Database Connection, Auth Middleware, Validation Middleware, Job Model, Order Service, Notification Service, Event Emitter, User Model
GET /api/jobs	`backend/tests/no-mocks/jobs.routes.test.ts#L329`	`backend/tests/with-mocks/jobs.routes.test.ts#L285`	Database Connection, Auth Middleware, Job Model
GET /api/jobs/available	`backend/tests/no-mocks/jobs.routes.test.ts#L390`	`backend/tests/with-mocks/jobs.routes.test.ts#L309`	Database Connection, Auth Middleware, Job Model
GET /api/jobs/mover	`backend/tests/no-mocks/jobs.routes.test.ts#L439`	`backend/tests/with-mocks/jobs.routes.test.ts#L333`	Database Connection, Auth Middleware, Job Model
GET /api/jobs/student	`backend/tests/no-mocks/jobs.routes.test.ts#L513`	`backend/tests/with-mocks/jobs.routes.test.ts#L378`	Database Connection, Auth Middleware, Job Model

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
GET /api/jobs/:id	`backend/tests/no-mocks/jobs.routes.test.ts#L585`	`backend/tests/with-mocks/jobs.routes.test.ts#L423`	Database Connection, Auth Middleware, Job Model
PATCH /api/jobs/:id/status	`backend/tests/no-mocks/jobs.routes.test.ts#L640`	`backend/tests/with-mocks/jobs.routes.test.ts#L602`	Database Connection, Auth Middleware, Validation Middleware, Job Model, Order Service, Notification Service, Event Emitter, User Model
POST /api/jobs/:id/arrived	`backend/tests/no-mocks/jobs.routes.test.ts#L902`	`backend/tests/with-mocks/jobs.routes.test.ts#L1242`	Database Connection, Auth Middleware, Job Model, Order Service, Notification Service, Event Emitter, User Model
POST /api/jobs/:id/confirm-pickup	`backend/tests/no-mocks/jobs.routes.test.ts#L1024`	`backend/tests/with-mocks/jobs.routes.test.ts#L1305`	Database Connection, Auth Middleware, Job Model, Order Service, Notification Service, Event Emitter, User Model
POST /api/jobs/:id/delivered	`backend/tests/no-mocks/jobs.routes.test.ts#L1215`	`backend/tests/with-mocks/jobs.routes.test.ts#L1367`	Database Connection, Auth Middleware, Job Model, Order Service, Notification Service, Event Emitter, User Model

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
POST /api/jobs/:id/confirm-delivery	`backend/tests/no-mocks/jobs.routes.test.ts#L1384`	`backend/tests/with-mocks/jobs.routes.test.ts#L1479`	Database Connection, Auth Middleware, Job Model, Order Service, Notification Service, Event Emitter, User Model
POST /api/order/quote	`backend/tests/no-mocks/order.routes.test.ts#L68`	`backend/tests/with-mocks/order.routes.test.ts#L150`	Socket.io, Job Service, Payment Service, Stripe Service, Order Model, Job Model, Event Emitter
POST /api/order	`backend/tests/no-mocks/order.routes.test.ts#L138`	`backend/tests/with-mocks/order.routes.test.ts#L270`	Socket.io, Job Service, Payment Service, Stripe Service, Order Model, Job Model, Event Emitter
POST /api/order/create-return-Job	`backend/tests/no-mocks/order.routes.test.ts#L262`	`backend/tests/with-mocks/order.routes.test.ts#L957`	Socket.io, Job Service, Payment Service, Stripe Service, Order Model, Job Model, Event Emitter
GET /api/order/all-orders	`backend/tests/no-mocks/order.routes.test.ts#L499`	`backend/tests/with-mocks/order.routes.test.ts#L1596`	Socket.io, Job Service, Payment Service, Stripe Service, Order Model, Job Model, Event Emitter
GET /api/order/active-order	`backend/tests/no-mocks/order.routes.test.ts#L616`	`backend/tests/with-mocks/order.routes.test.ts#L1878`	Socket.io, Job Service, Payment Service, Stripe Service, Order Model, Job Model, Event Emitter

Interface	Describe Group Location, No Mocks	Describe Group Location, With Mocks	Mocked Components
DELETE /api/order/cancel-order	`backend/tests/no-mocks/order.routes.test.ts#L761`	`backend/tests/with-mocks/order.routes.test.ts#L2032`	Socket.io, Job Service, Payment Service, Stripe Service, Order Model, Job Model, Event Emitter
POST /api/payment/create-intent	`backend/tests/no-mocks/payment.routes.test.ts#L85`	`backend/tests/with-mocks/payment.routes.test.ts#L97`	Socket.io, Stripe Service
POST /api/payment/process	`backend/tests/no-mocks/payment.routes.test.ts#L173`	`backend/tests/with-mocks/payment.routes.test.ts#L250`	Socket.io, Stripe Service
GET /api/payment/status/:paymentIntentId	`backend/tests/no-mocks/payment.routes.test.ts#L206`	N/A	N/A
GET /api/routePlanner/smart	`backend/tests/no-mocks/routePlanner.routes.test.ts#L82`	`backend/tests/with-mocks/routePlanner.routes.test.ts#L55`	Database Connection, Auth Middleware, User Model, Job Service, Mongoose Utils
GET /api/user/profile	`backend/tests/no-mocks/user.routes.test.ts#L82`	`backend/tests/with-mocks/user.routes.test.ts#L100`	None (uses real DB)
POST /api/user/profile	`backend/tests/no-mocks/user.routes.test.ts#L122`	`backend/tests/with-mocks/user.routes.test.ts#L131`	None (uses real DB)
POST /api/user/cash-out	`backend/tests/no-mocks/user.routes.test.ts#L275`	`backend/tests/with-mocks/user.routes.test.ts#L283`	None (uses real DB)
DELETE /api/user/profile	`backend/tests/no-mocks/user.routes.test.ts#L362`	`backend/tests/with-mocks/user.routes.test.ts#L443`	None (uses real DB)
Socket.IO - Order Status Updates	N/A	`backend/tests/with-mocks/socket.test.ts#L171`	None (integration test with real DB)
Socket.IO - Job Status Updates	N/A	`backend/tests/with-mocks/socket.test.ts#L532`	None (integration test with real DB)

2.1.2. Commit Hash Where Tests Run

`deb761996260ef597203cf65d2f5bd474ca74bc4`

2.1.3. Explanation on How to Run the Tests

Prerequisites:

- Ensure you have Node.js (version 14 or higher) and npm installed on your system. You can check by running `node --version` and `npm --version` in your terminal.

1. **Clone the Repository**:
    - Open your terminal and run:

    ```
    git clone https://github.com/Dormdash-CPEN321/DormDash.git CPEN321-Project
    ```

2. **Navigate to the Backend Directory**:

   - Change to the backend directory:

     ```
     cd CPEN321-Project/backend
     ```

3. **Install Dependencies**:

   - Install the required npm packages:

     ```
     npm install
     ```

4. **Update Test Scripts**:

   - Update all the test packages:

     ```
     npm run update-test-scripts
     ```

5. **Run the Tests**:

   - To run all the tests:

     ```
     npm run test
     ```

   - To run all tests with coverage:

     ```
     npm run test:coverage
     ```

   - To run specific tests (e.g., order tests with mocks and coverage):

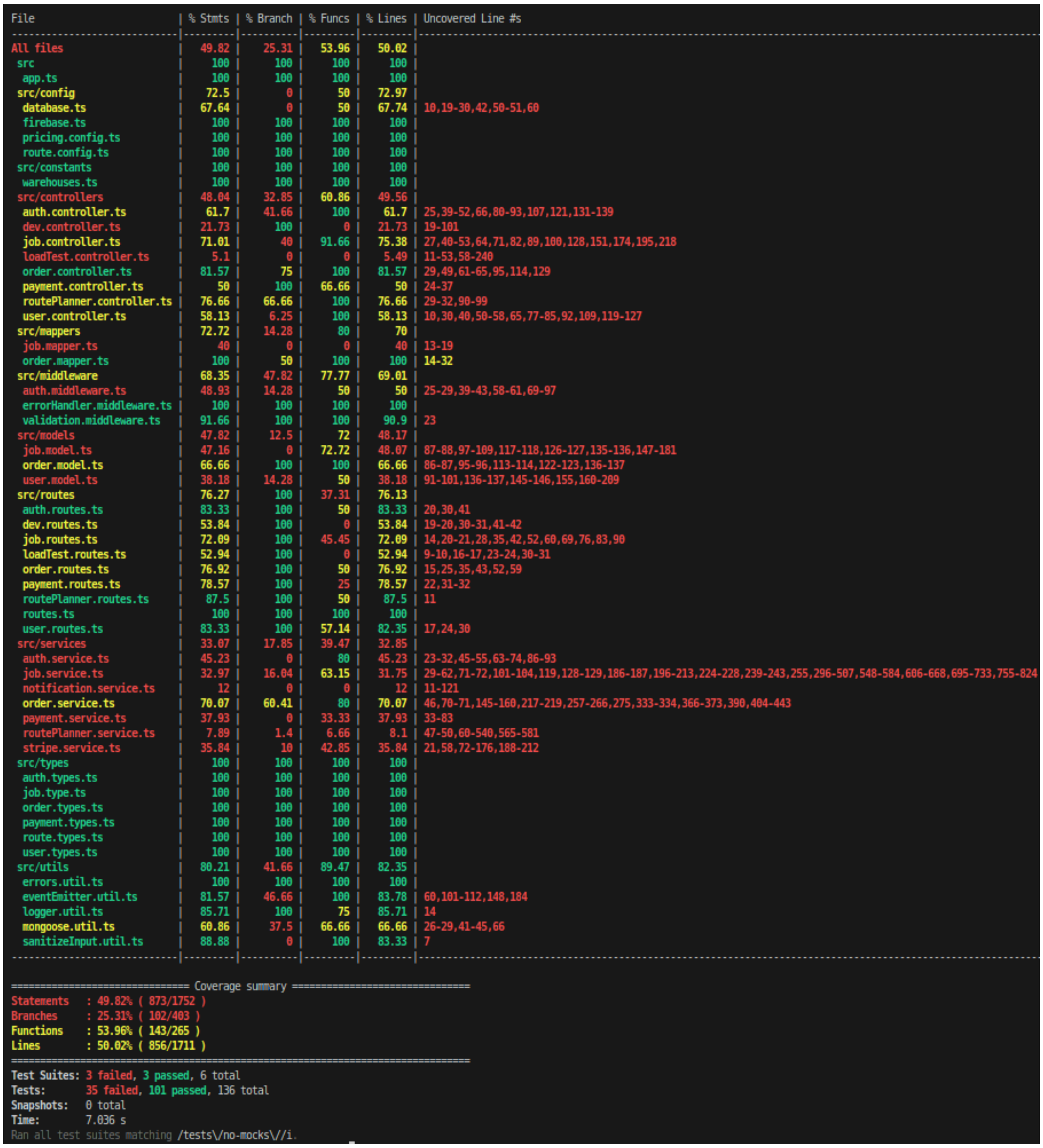
     ```
     npm run test:mock:order:coverage
     ```

     Replace `order` with the desired module (e.g., `auth`, `jobs`, `payment`, `routePlanner`, `user`) and choose `mock` or `nomock` as needed. For coverage, append `:coverage` to the script name.

2.2. GitHub Actions Configuration Location

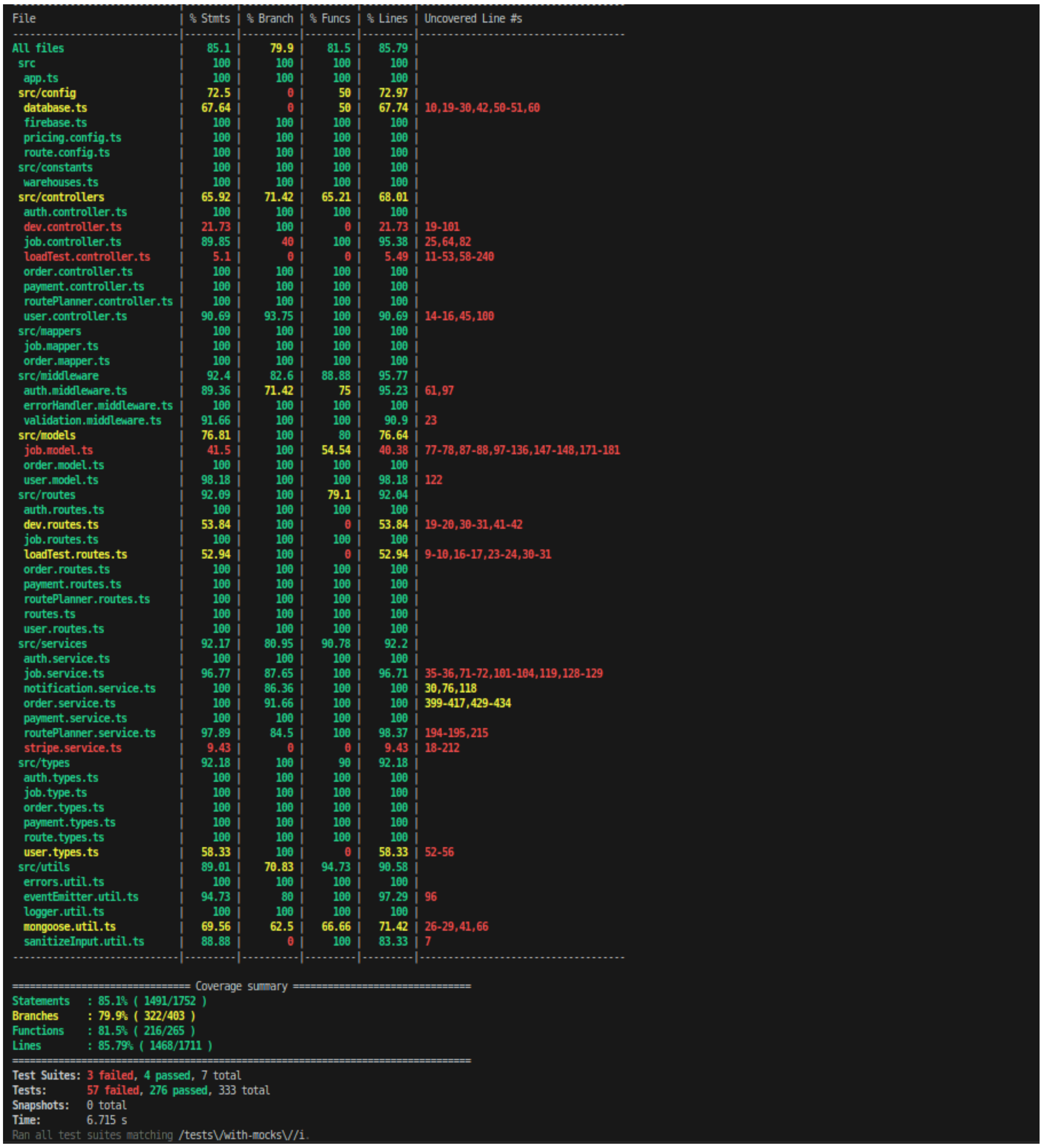
`~/.github/workflows/backend-tests.yml`

2.3. Jest Coverage Report Screenshots for Tests Without Mocking



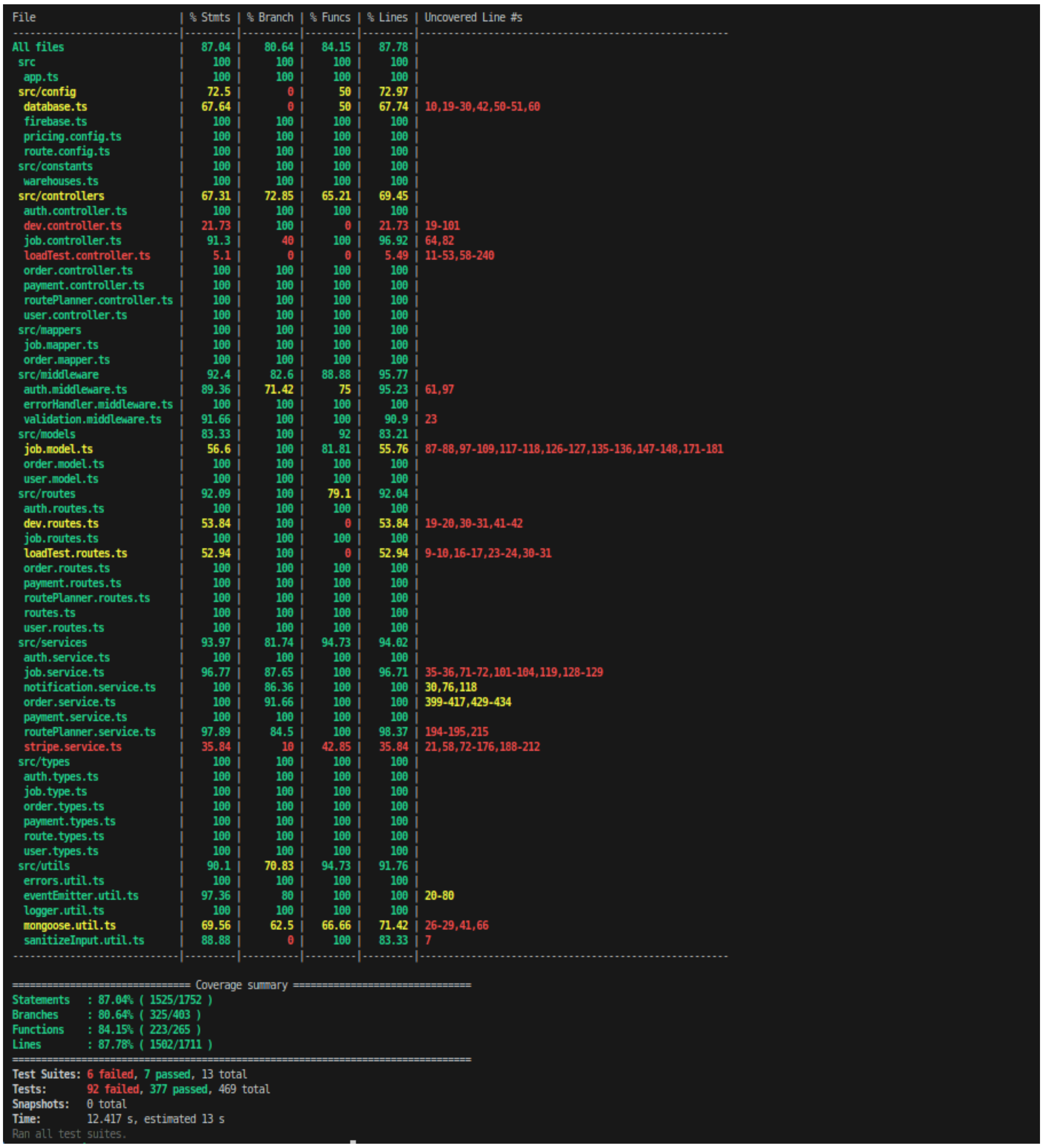
```
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|-------------------
All files                    |   49.82 |    25.31 |   53.96 |   50.02 |
 src                         |     100 |      100 |     100 |     100 |
  app.ts                     |     100 |      100 |     100 |     100 |
 src/config                  |    72.5 |        0 |      50 |   72.97 |
  database.ts                |   67.64 |        0 |      50 |   67.74 | 10,19-30,42,50-51,60
  firebase.ts                |     100 |      100 |     100 |     100 |
  pricing.config.ts          |     100 |      100 |     100 |     100 |
  route.config.ts            |     100 |      100 |     100 |     100 |
 src/constants               |     100 |      100 |     100 |     100 |
  warehouses.ts              |     100 |      100 |     100 |     100 |
 src/controllers             |   48.04 |    32.85 |   60.86 |   49.56 |
  auth.controller.ts         |    61.7 |    41.66 |     100 |    61.7 | 25,39-52,66,80-93,107,121,131-139
  dev.controller.ts          |   21.73 |      100 |       0 |   21.73 | 19-101
  job.controller.ts          |   71.01 |       40 |   91.66 |   75.38 | 27,40-53,64,71,82,89,100,128,151,174,195,218
  loadTest.controller.ts     |     5.1 |        0 |       0 |    5.49 | 11-53,58-240
  order.controller.ts        |   81.57 |       75 |     100 |   81.57 | 29,49,61-65,95,114,129
  payment.controller.ts      |      50 |      100 |   66.66 |      50 | 24-37
  routePlanner.controller.ts |   76.66 |    66.66 |     100 |   76.66 | 29-32,90-99
  user.controller.ts         |   58.13 |     6.25 |     100 |   58.13 | 10,30,40,50-58,65,77-85,92,109,119-127
 src/mappers                 |   72.72 |    14.28 |      80 |      70 |
  job.mapper.ts              |      40 |        0 |       0 |      40 | 13-19
  order.mapper.ts            |     100 |       50 |     100 |     100 | 14-32
 src/middleware              |   68.35 |    47.82 |   77.77 |   69.01 |
  auth.middleware.ts         |   48.93 |    14.28 |      50 |      50 | 25-29,39-43,58-61,69-97
  errorHandler.middleware.ts |     100 |      100 |     100 |     100 |
  validation.middleware.ts   |   91.66 |      100 |     100 |    90.9 | 23
 src/models                  |   47.82 |     12.5 |      72 |   48.17 |
  job.model.ts               |   47.16 |        0 |   72.72 |   48.07 | 87-88,97-109,117-118,126-127,135-136,147-181
  order.model.ts             |   66.66 |      100 |     100 |   66.66 | 86-87,95-96,113-114,122-123,136-137
  user.model.ts              |   38.18 |    14.28 |      50 |   38.18 | 91-101,136-137,145-146,155,160-209
 src/routes                  |   76.27 |      100 |   37.31 |   76.13 |
  auth.routes.ts             |   83.33 |      100 |      50 |   83.33 | 20,30,41
  dev.routes.ts              |   53.84 |      100 |       0 |   53.84 | 19-20,30-31,41-42
  job.routes.ts              |   72.09 |      100 |   45.45 |   72.09 | 14,20-21,28,35,42,52,60,69,76,83,90
  loadTest.routes.ts         |   52.94 |      100 |       0 |   52.94 | 9-10,16-17,23-24,30-31
  order.routes.ts            |   76.92 |      100 |      50 |   76.92 | 15,25,35,43,52,59
  payment.routes.ts          |   78.57 |      100 |      25 |   78.57 | 22,31-32
  routePlanner.routes.ts     |    87.5 |      100 |      50 |    87.5 | 11
  routes.ts                  |     100 |      100 |     100 |     100 |
  user.routes.ts             |   83.33 |      100 |   57.14 |   82.35 | 17,24,30
 src/services                |   33.07 |    17.85 |   39.47 |   32.85 |
  auth.service.ts            |   45.23 |        0 |      80 |   45.23 | 23-32,45-55,63-74,86-93
  job.service.ts             |   32.97 |    16.04 |   63.15 |   31.75 | 29-62,71-72,101-104,119,128-129,186-187,196-213,224-228,239-243,255,296-507,548-584,606-668,695-733,755-824
  notification.service.ts    |      12 |        0 |       0 |      12 | 11-121
  order.service.ts           |   70.07 |    60.41 |      80 |   70.07 | 46,70-71,145-160,217-219,257-266,275,333-334,366-373,390,404-443
  payment.service.ts         |   37.93 |        0 |   33.33 |   37.93 | 33-83
  routePlanner.service.ts    |    7.89 |      1.4 |    6.66 |     8.1 | 47-50,60-540,565-581
  stripe.service.ts          |   35.84 |       10 |   42.85 |   35.84 | 21,58,72-176,188-212
 src/types                   |     100 |      100 |     100 |     100 |
  auth.types.ts              |     100 |      100 |     100 |     100 |
  job.type.ts                |     100 |      100 |     100 |     100 |
  order.types.ts             |     100 |      100 |     100 |     100 |
  payment.types.ts           |     100 |      100 |     100 |     100 |
  route.types.ts             |     100 |      100 |     100 |     100 |
  user.types.ts              |     100 |      100 |     100 |     100 |
 src/utils                   |   80.21 |    41.66 |   89.47 |   82.35 |
  errors.util.ts             |     100 |      100 |     100 |     100 |
  eventEmitter.util.ts       |   81.57 |    46.66 |     100 |   83.78 | 60,101-112,148,184
  logger.util.ts             |   85.71 |      100 |      75 |   85.71 | 14
  mongoose.util.ts           |   60.86 |     37.5 |   66.66 |   66.66 | 26-29,41-45,66
  sanitizeInput.util.ts      |   88.88 |        0 |     100 |   83.33 | 7
-----------------------------|---------|----------|---------|---------|-------------------

=============================== Coverage summary ===============================
Statements   : 49.82% ( 873/1752 )
Branches     : 25.31% ( 102/403 )
Functions    : 53.96% ( 143/265 )
Lines        : 50.02% ( 856/1711 )
================================================================================
Test Suites: 3 failed, 3 passed, 6 total
Tests:       35 failed, 101 passed, 136 total
Snapshots:   0 total
Time:        7.036 s
Ran all test suites matching /tests\/no-mocks\//i.
```

2.4. Jest Coverage Report Screenshots for Tests With Mocking



```
-----------------------------|---------|----------|---------|---------|-----------------------------------
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|-----------------------------------
All files                    |    85.1 |     79.9 |    81.5 |   85.79 |
 src                         |     100 |      100 |     100 |     100 |
  app.ts                     |     100 |      100 |     100 |     100 |
 src/config                  |    72.5 |        0 |      50 |   72.97 |
  database.ts                |   67.64 |        0 |      50 |   67.74 | 10,19-30,42,50-51,60
  firebase.ts                |     100 |      100 |     100 |     100 |
  pricing.config.ts          |     100 |      100 |     100 |     100 |
  route.config.ts            |     100 |      100 |     100 |     100 |
 src/constants               |     100 |      100 |     100 |     100 |
  warehouses.ts              |     100 |      100 |     100 |     100 |
 src/controllers             |   65.92 |    71.42 |   65.21 |   68.01 |
  auth.controller.ts         |     100 |      100 |     100 |     100 |
  dev.controller.ts          |   21.73 |      100 |       0 |   21.73 | 19-101
  job.controller.ts          |   89.85 |       40 |     100 |   95.38 | 25,64,82
  loadTest.controller.ts     |     5.1 |        0 |       0 |    5.49 | 11-53,58-240
  order.controller.ts        |     100 |      100 |     100 |     100 |
  payment.controller.ts      |     100 |      100 |     100 |     100 |
  routePlanner.controller.ts |     100 |      100 |     100 |     100 |
  user.controller.ts         |   90.69 |    93.75 |     100 |   90.69 | 14-16,45,100
 src/mappers                 |     100 |      100 |     100 |     100 |
  job.mapper.ts              |     100 |      100 |     100 |     100 |
  order.mapper.ts            |     100 |      100 |     100 |     100 |
 src/middleware              |    92.4 |     82.6 |   88.88 |   95.77 |
  auth.middleware.ts         |   89.36 |    71.42 |      75 |   95.23 | 61,97
  errorHandler.middleware.ts |     100 |      100 |     100 |     100 |
  validation.middleware.ts   |   91.66 |      100 |     100 |    90.9 | 23
 src/models                  |   76.81 |      100 |      80 |   76.64 |
  job.model.ts               |    41.5 |      100 |   54.54 |   40.38 | 77-78,87-88,97-136,147-148,171-181
  order.model.ts             |     100 |      100 |     100 |     100 |
  user.model.ts              |   98.18 |      100 |     100 |   98.18 | 122
 src/routes                  |   92.09 |      100 |    79.1 |   92.04 |
  auth.routes.ts             |     100 |      100 |     100 |     100 |
  dev.routes.ts              |   53.84 |      100 |       0 |   53.84 | 19-20,30-31,41-42
  job.routes.ts              |     100 |      100 |     100 |     100 |
  loadTest.routes.ts         |   52.94 |      100 |       0 |   52.94 | 9-10,16-17,23-24,30-31
  order.routes.ts            |     100 |      100 |     100 |     100 |
  payment.routes.ts          |     100 |      100 |     100 |     100 |
  routePlanner.routes.ts     |     100 |      100 |     100 |     100 |
  routes.ts                  |     100 |      100 |     100 |     100 |
  user.routes.ts             |     100 |      100 |     100 |     100 |
 src/services                |   92.17 |    80.95 |   90.78 |    92.2 |
  auth.service.ts            |     100 |      100 |     100 |     100 |
  job.service.ts             |   96.77 |    87.65 |     100 |   96.71 | 35-36,71-72,101-104,119,128-129
  notification.service.ts    |     100 |    86.36 |     100 |     100 | 30,76,118
  order.service.ts           |     100 |    91.66 |     100 |     100 | 399-417,429-434
  payment.service.ts         |     100 |      100 |     100 |     100 |
  routePlanner.service.ts    |   97.89 |     84.5 |     100 |   98.37 | 194-195,215
  stripe.service.ts          |    9.43 |        0 |       0 |    9.43 | 18-212
 src/types                   |   92.18 |      100 |      90 |   92.18 |
  auth.types.ts              |     100 |      100 |     100 |     100 |
  job.type.ts                |     100 |      100 |     100 |     100 |
  order.types.ts             |     100 |      100 |     100 |     100 |
  payment.types.ts           |     100 |      100 |     100 |     100 |
  route.types.ts             |     100 |      100 |     100 |     100 |
  user.types.ts              |   58.33 |      100 |       0 |   58.33 | 52-56
 src/utils                   |   89.01 |    70.83 |   94.73 |   90.58 |
  errors.util.ts             |     100 |      100 |     100 |     100 |
  eventEmitter.util.ts       |   94.73 |       80 |     100 |   97.29 | 96
  logger.util.ts             |     100 |      100 |     100 |     100 |
  mongoose.util.ts           |   69.56 |     62.5 |   66.66 |   71.42 | 26-29,41,66
  sanitizeInput.util.ts      |   88.88 |        0 |     100 |   83.33 | 7
-----------------------------|---------|----------|---------|---------|-----------------------------------

=============================== Coverage summary ===============================
Statements   : 85.1% ( 1491/1752 )
Branches     : 79.9% ( 322/403 )
Functions    : 81.5% ( 216/265 )
Lines        : 85.79% ( 1468/1711 )
================================================================================
Test Suites: 3 failed, 4 passed, 7 total
Tests:       57 failed, 276 passed, 333 total
Snapshots:   0 total
Time:        6.715 s
Ran all test suites matching /tests\/with-mocks\//i.
```

2.5. Jest Coverage Report Screenshots for Both Tests With and Without Mocking



```
File                         | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------------|---------|----------|---------|---------|---------------------------------------------------
All files                    |   87.04 |    80.64 |   84.15 |   87.78 |
 src                         |     100 |      100 |     100 |     100 |
  app.ts                     |     100 |      100 |     100 |     100 |
 src/config                  |    72.5 |        0 |      50 |   72.97 |
  database.ts                |   67.64 |        0 |      50 |   67.74 | 10,19-30,42,50-51,60
  firebase.ts                |     100 |      100 |     100 |     100 |
  pricing.config.ts          |     100 |      100 |     100 |     100 |
  route.config.ts            |     100 |      100 |     100 |     100 |
 src/constants               |     100 |      100 |     100 |     100 |
  warehouses.ts              |     100 |      100 |     100 |     100 |
 src/controllers             |   67.31 |    72.85 |   65.21 |   69.45 |
  auth.controller.ts         |     100 |      100 |     100 |     100 |
  dev.controller.ts          |   21.73 |      100 |       0 |   21.73 | 19-101
  job.controller.ts          |    91.3 |       40 |     100 |   96.92 | 64,82
  loadTest.controller.ts     |     5.1 |        0 |       0 |    5.49 | 11-53,58-240
  order.controller.ts        |     100 |      100 |     100 |     100 |
  payment.controller.ts      |     100 |      100 |     100 |     100 |
  routePlanner.controller.ts |     100 |      100 |     100 |     100 |
  user.controller.ts         |     100 |      100 |     100 |     100 |
 src/mappers                 |     100 |      100 |     100 |     100 |
  job.mapper.ts              |     100 |      100 |     100 |     100 |
  order.mapper.ts            |     100 |      100 |     100 |     100 |
 src/middleware              |    92.4 |     82.6 |   88.88 |   95.77 |
  auth.middleware.ts         |   89.36 |    71.42 |      75 |   95.23 | 61,97
  errorHandler.middleware.ts |     100 |      100 |     100 |     100 |
  validation.middleware.ts   |   91.66 |      100 |     100 |    90.9 | 23
 src/models                  |   83.33 |      100 |      92 |   83.21 |
  job.model.ts               |    56.6 |      100 |   81.81 |   55.76 | 87-88,97-109,117-118,126-127,135-136,147-148,171-181
  order.model.ts             |     100 |      100 |     100 |     100 |
  user.model.ts              |     100 |      100 |     100 |     100 |
 src/routes                  |   92.09 |      100 |    79.1 |   92.04 |
  auth.routes.ts             |     100 |      100 |     100 |     100 |
  dev.routes.ts              |   53.84 |      100 |       0 |   53.84 | 19-20,30-31,41-42
  job.routes.ts              |     100 |      100 |     100 |     100 |
  loadTest.routes.ts         |   52.94 |      100 |       0 |   52.94 | 9-10,16-17,23-24,30-31
  order.routes.ts            |     100 |      100 |     100 |     100 |
  payment.routes.ts          |     100 |      100 |     100 |     100 |
  routePlanner.routes.ts     |     100 |      100 |     100 |     100 |
  routes.ts                  |     100 |      100 |     100 |     100 |
  user.routes.ts             |     100 |      100 |     100 |     100 |
 src/services                |   93.97 |    81.74 |   94.73 |   94.02 |
  auth.service.ts            |     100 |      100 |     100 |     100 |
  job.service.ts             |   96.77 |    87.65 |     100 |   96.71 | 35-36,71-72,101-104,119,128-129
  notification.service.ts    |     100 |    86.36 |     100 |     100 | 30,76,118
  order.service.ts           |     100 |    91.66 |     100 |     100 | 399-417,429-434
  payment.service.ts         |     100 |      100 |     100 |     100 |
  routePlanner.service.ts    |   97.89 |     84.5 |     100 |   98.37 | 194-195,215
  stripe.service.ts          |   35.84 |       10 |   42.85 |   35.84 | 21,58,72-176,188-212
 src/types                   |     100 |      100 |     100 |     100 |
  auth.types.ts              |     100 |      100 |     100 |     100 |
  job.type.ts                |     100 |      100 |     100 |     100 |
  order.types.ts             |     100 |      100 |     100 |     100 |
  payment.types.ts           |     100 |      100 |     100 |     100 |
  route.types.ts             |     100 |      100 |     100 |     100 |
  user.types.ts              |     100 |      100 |     100 |     100 |
 src/utils                   |    90.1 |    70.83 |   94.73 |   91.76 |
  errors.util.ts             |     100 |      100 |     100 |     100 |
  eventEmitter.util.ts       |   97.36 |       80 |     100 |     100 | 20-80
  logger.util.ts             |     100 |      100 |     100 |     100 |
  mongoose.util.ts           |   69.56 |     62.5 |   66.66 |   71.42 | 26-29,41,66
  sanitizeInput.util.ts      |   88.88 |        0 |     100 |   83.33 | 7
-----------------------------|---------|----------|---------|---------|---------------------------------------------------

=============================== Coverage summary ===============================
Statements   : 87.04% ( 1525/1752 )
Branches     : 80.64% ( 325/403 )
Functions    : 84.15% ( 223/265 )
Lines        : 87.78% ( 1502/1711 )
================================================================================
Test Suites: 6 failed, 7 passed, 13 total
Tests:       92 failed, 377 passed, 469 total
Snapshots:   0 total
Time:        12.417 s, estimated 13 s
Ran all test suites.
```

Justification: dev.controller.ts: This is used for E2E testing the FindJob feature, so is exempt from coverage. loadTest.controller.ts: This is used for testing non-functional requirements, so is exempt from coverage. database.ts: These lines are used as error handlers, for shutting down the server of the mongo db connection, so are not executable through endpoints. auth.middleware.ts: The try-catch block inside the IIFE catches ALL errors that occur within the async function Notification. Service: The branches that are not covered are defensive checks to ensure that unexpected errors are handled correctly

3. Back-end Test Specification: Tests of Non-Functional Requirements

3.1. Test Locations in Git

Non-Functional Requirement	**Location in Git**
Scalability	`backend/tests/load-test/load-test.js`
UI Responsivness	`frontend/app/src/androidTest/java/com/cpen321/usermanagement/performance/UIResponsivness/MoverUIResponseTimeTes` `frontend/app/src/androidTest/java/com/cpen321/usermanagement/performance/UIResponsivness/StudentUIResponseTimeT`

3.2. Test Verification and Logs

- **Scalability**

- **Verification:** This scalability test evaluates the system's ability to handle concurrent load and maintain performance stability under stress using K6. It simulates real-world traffic by continuously sending requests to two critical API endpoints: `POST /api/order`, representing students placing storage or moving orders, and `GET /api/jobs/available`, representing movers retrieving available job listings. The test ramps up to 200 virtual users, sustains peak load for two minutes, and then scales down to assess elasticity and resilience. Approximately 70% of the traffic targets the order creation endpoint (write-heavy), while 30% targets job retrieval (read-heavy). Each virtual user uses a valid JWT token generated from real user IDs fetched from the deployed API, ensuring realistic authentication behavior. Key performance metrics—such as error rate, 90th percentile latency, and throughput—are tracked throughout the test, with thresholds requiring latency below 20 seconds and error rates below 1%. The goal is to validate that the backend remains stable, responsive, and scalable under high concurrency while identifying any performance bottlenecks or degradation patterns.
- **Log Output**

```
Scalability Overview:
Total Requests: 4401
Failed Requests: 0.05%
Average Response Time: 5203.29ms
90th Percentile Latency: 11686.70ms
Requests per Second: 18.23

Threshold Results:
Error Rate: 0.05% ✓
90th Percentile Latency: 11686.70ms ✓

Scalability Metrics:
Peak Virtual Users (VUs): N/A
Test Duration: N/A
Total Data Transferred: 985.01 MB
Data Sent: 2.82 MB

Check Results:
0: 100.00% (1338/1338)
1: 100.00% (1338/1338)
2: 99.93% (3061/3063)
3: 99.93% (3061/3063)
4: 100.00% (3061/3061)

Summary:
✅ Scalability test completed successfully.
System remained stable under high concurrency.
Performance degradation observed is within acceptable range.
```

- **UI Responsivness**
  - **Verification:** This test suite evaluates the responsiveness of core UI components that operate independently of API calls, focusing purely on client-side interactions. The goal is to ensure that user actions trigger visible state changes within 0.1 seconds, maintaining a seamless experience. For the Mover interface, tests cover interactions such as navigating to the Find Jobs screen, updating Availability, and clicking the Get Optimal Route button. For the Student interface, tests include actions like clicking Create New Order or editing the profile, verifying that UI feedback occurs within the defined responsiveness threshold.
  - **Running the Tests:**

1. Create a **Mover** or **Student** account.
2. When prompted, enter text for the **bio** field and click **Save**.
3. Verify that only **one account** appears when clicking **Sign In**.
4. Before switching between **Mover** and **Student** tests, **delete the existing account** and create a new one by repeating the steps above.
   - Avoid keeping both Mover and Student accounts saved simultaneously, as this may cause the test to sign in with the wrong account type (e.g., selecting a Mover account during a Student test), leading to test failures.

Note: Due to occasional UI flakiness, the test may need to be run up to three times before passing.

- **Log Output**

MoverUIResponseTimeTest.kt

```
Pixel 4 - 13 Tests 3/3 completed. (0 skipped) (0 failed)
Finished 3 tests on Pixel 4 - 13

BUILD SUCCESSFUL in 50s
```

StudentUIResponseTimeTest.kt

```
Pixel 4 - 13 Tests 2/2 completed. (0 skipped) (0 failed)
Finished 2 tests on Pixel 4 - 13
BUILD SUCCESSFUL in 3m 27s
```

4. Front-end Test Specification

Instructions to run the tests:

- Clone the repository
- Create a test.properties file under DormDash/frontend/app/src/androidTest/resources
- Add the following to the file, and fill in 2 test accounts (1 for student and 1 for mover): STUDENT_EMAIL= STUDENT_PASSWORD= MOVER_EMAIL= MOVER_PASSWORD= -Optionally set up those accounts with google, but our tests do that anyways
- Run tests

4.1. Location in Git of Front-end Test Suite:

`frontend/src/androidTest/java/com/cpen321/usermanagement/features`

4.2. Tests

Feature: Authenticate

- **Use Case: Sign In**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User opens app and clicks on 'Sign In with Google'	Ensure 'Sign in with Google' exists and click it
2. System suggests list of authenticated accounts	Ensure accounts are present
3. User selects their desired account	Click on desired account
4. The system authenticates users and redirects them to the role-based main screen.	Ensure existence of app title 'Dormdash'
4a. User does not exist	Not tested, since we are asked to set up an account for the tester; the user will already be signed up

  - **Test Logs:**

```
> Task :app:connectedDebugAndroidTest
Starting 2 tests on Pixel_7(AVD) - 13
Connected to process 13912 on device 'Pixel_7 [emulator-5554]'.

Pixel_7(AVD) - 13 Tests 1/2 completed. (0 skipped) (0 failed)
Pixel_7(AVD) - 13 Tests 2/2 completed. (0 skipped) (0 failed)
Finished 2 tests on Pixel_7(AVD) - 13
```

-**Use Case: Sign Up**

- **Use Case: Sign Up**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User opens app and clicks on 'Sign Up with Google'	Ensure 'Sign up with Google' exists and click it
2. System suggests list of authenticated accounts	Ensure accounts are present
3. User selects their desired account	Click on desired account
4. The system authenticates users and redirects them to the role selection screen.	Ensure existence of role selection screen identifiers ('I'm a Student' and 'I'm a Mover')
4a. User already exists	Check for 'User already exists, please sign in instead'
5. The user selects their role.	Choose 'I'm a Student' for this account and ensure redirection to role-based main screen; verify app title 'Dormdash'

  - **Test Logs:**

```
> Task :app:connectedDebugAndroidTest
Starting 2 tests on Pixel_7(AVD) - 13
Connected to process 22011 on device 'Pixel_7 [emulator-5554]'.

Pixel_7(AVD) - 13 Tests 1/2 completed. (0 skipped) (0 failed)
Finished 2 tests on Pixel_7(AVD) - 13
```

```
BUILD SUCCESSFUL in 35s
76 actionable tasks: 1 executed, 75 up-to-date
```

- **Use case: Sign Out**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User opens app and executes ‘Sign in’ use case.	Execute Sign in test
2. User clicks on profile icon	Ensure profile icon exists and click it
3. System redirects user to profile screen	Ensure profile screen elements exist like ‘Sign out’
4. The user clicks Sign Out	Click Sign Out
5. The system de-authenticates the account and redirects the user to the authentication screen.	Ensure auth screen identifiers exist, like app title and Google auth buttons

  - **Test Logs:**

```
> Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_7(AVD) - 13

Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13

BUILD SUCCESSFUL in 35s
76 actionable tasks: 1 executed, 75 up-to-date
```

-**Use case: Delete Account**

- **Use case: Delete Account**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User opens app and executes ‘Sign in’ use case.	Execute Sign in test
2. User clicks on profile icon	Ensure profile icon exists and click it
3. System redirects user to profile screen	Ensure profile screen elements exist like ‘Delete Account’
4. The user clicks ‘Delete Account’	Click ‘Delete Account’
5. The system prompts user to confirm	Ensure confirm button existence
6. User clicks ‘Confirm’	Click ‘Confirm’
7. System deletes account and redirects user to authentication screen	Ensure auth screen identifiers exist, like app title and Google auth buttons

    - **Test Logs:**

```
```

> Task :app:connectedDebugAndroidTest Starting 1 tests on Pixel_7(AVD) - 13 Connected to process 17126 on device 'Pixel_7 [emulator-5554]'.

Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed) Finished 1 tests on Pixel_7(AVD) - 13

BUILD SUCCESSFUL in 34s 76 actionable tasks: 1 executed, 75 up-to-date

```
```

Feature: Manage Orders

- **Use case: Create Order**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. Student clicks “Create New Order” button	Ensure its existence and click ‘Create New Order’

Scenario Steps	Test Case Steps
2. System displays auto-complete text field for address	Ensure text field existence
3. The student enters the pick up address.	Enter address in text field
4. System suggests a list of valid addresses matching student input using Google Maps	Ensure suggestion existence
5. Student selects address by clicking on the suggestion and clicks on “Get Base Delivery Charge” button	Select suggestion and click “Get Base Delivery Charge”
5a. Student enters address outside of Greater Vancouver area	Input address outside of Greater Vancouver Area, and select suggestion
5a1 System displays an error message: “We currently only service Greater Vancouver.” and continues to display the address input field.	Ensure System displays error message: “We currently only service Greater Vancouver.” and continues to display the address input field.
6. The system displays two date picker field for pickup/return dates, +/- buttons to adjust the hour and 15 minute time intervals for the pickup/return time, and +/- buttons for the number of small, medium, large boxes.	Ensure fields exist
7. Student enters both time fields, dates and number of boxes.	Keep as default for success scenario
7a Student inputs a return date or time which is before the pickup date or time	Input a return date or time which is before the pickup date or time
7a1 System displays error: “Return date/time must be after pickup date/time” and stays on the form.	Ensure System displays error: “Return date/time must be after pickup date/time” and stays on the form.
8. Student adds boxes to order and clicks ‘Proceed to Payment’	Click on ‘+’ for small box and ‘Proceed to Payment’
9. Execute ‘UC-2 Pay’ use case successfully	Run payment use case test, which is a static function to not lose progress and restart.
10. Order is placed and the system displays the order status, details, and pin showing pick up address using Google Maps.	Ensure ‘Active Order’ text

- **Execution Logs / Notes:**

```
java.lang.AssertionError: Failed to inject touch input.
Reason: Expected exactly '1' node but could not find any node that satisfies: (Text + InputText +
EditableText contains
 '15' (ignoreCase: false))
```

- **Use case: Pay**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. User Enters a Valid Name and Email.	Enter Name and Email.
1a. User Enters invalid information.	Enter nothing, which is considered invalid.
2. System Displays "Please fill in all required fields with valid information."	Ensure error message, "Please fill in all required fields with valid information." is displayed.
3. User clicks 'Process Payment'	Click on 'Process Payment'
4. System asks users to confirm.	Ensure '"Confirm & Pay' button exists, and click on it.
5. System displays active order.	Ensure active order exists.

  - **Execution Logs / Notes:** Not reached, since test fails before is executed.

- **Use case: Create Return Job**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. Student clicks on "Schedule Return Delivery" button	1. Click on "Schedule Return Delivery" button
2. System displays bottom sheet with return job form showing "Select Return Date & Time" with expected return date displayed	2. Ensure the bottom sheet and "Select Return Date & Time" are displayed and the expected return date is visible
3. System displays selected time on the time card and calculates days difference from expected return date	3. Ensure the selected time is shown on the time card and that "Early Return Refund" information is present
4. Student selects early return date by clicking ‘Continue’	4. Click "Continue" to accept the current date as the return date

Scenario Steps	Test Case Steps
5. System displays address selection step with default return address (same as pickup address) and option to use custom address	5. Ensure address selection step appears, default return address is shown, and "Use custom address" option exists
6. Student selects custom address	6. Click on "Use custom address"
7. Student enters custom address in auto-complete field	7. Enter the custom address in the auto-complete field and select a suggestion
8. Student enters address outside of Greater Vancouver area and clicks 'Confirm Address' (failure scenario)	8. Enter an invalid address, select the suggestion, and click "Confirm Address"
8a1. System displays error message: "Invalid address. Please select a valid address within Greater Vancouver."	8a1. Ensure the error message "Invalid address. Please select a valid address within Greater Vancouver." is displayed
9. Student clicks 'Confirm Address' button (success path)	9. Click "Confirm Address"
10. System displays success message with refund/late fee information: "Return job created! Refund of $X.XX has been processed for early return."	10. Ensure the success message with refund information is displayed

- **Execution Logs:**

```
> Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_7(AVD) - 13

Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13

BUILD SUCCESSFUL in 1m 7s
76 actionable tasks: 7 executed, 69 up-to-date
```

- **Use case: Confirm Pickup**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. Student clicks on 'Confirm Pickup' button	1. Ensure 'Confirm Pickup' button exists and click it
2. System displays success message 'Mover has picked up your items'	2. Ensure success message 'Mover has picked up your items' is displayed

  - **Execution Logs:**

```
> Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_7(AVD) - 13
Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13
BUILD SUCCESSFUL in 1m 7s
76 actionable tasks: 7 executed, 69 up-to-date
```

- **Use case: Cancel Order**
  - **Expected Behaviors:**

Scenario Steps	Test Case Steps
1. Student clicks on 'Profile' button	1. Ensure profile icon exists and click it
2. System displays manage profile screen	2. Ensure profile screen elements exist like 'Manage Orders'
3. Student clicks 'Manage Orders'	3. Click 'Manage Orders'
4. System displays all orders	4. Ensure order list exists
5. Student clicks on pending order and clicks 'Cancel'	5. Click on a pending order and click 'Cancel'
6. System prints 'Order cancelled successfully, refund has been processed'	6. Ensure success message 'Order cancelled successfully, refund has been processed' is displayed

  - **Execution Logs:**

```
> Task :app:connectedDebugAndroidTest
Starting 1 tests on Pixel_7(AVD) - 13
Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13
```

```
BUILD SUCCESSFUL in 1m 7s
76 actionable tasks: 7 executed, 69 up-to-date
```

Feature: Find Jobs

- **Use Case: Browse and Filter Jobs**

  Expected Behaviors:

Scenario Steps	**Test Case Steps**
1. Mover clicks on "Find Jobs" on the navigation bar on the bottom of the home page.	Click the "Find Jobs" tab.
2. The system displays all unassigned jobs including the pickup, drop-off addresses, volume of items, date and time, the type of job (storage or return) and the credit they can earn from the job.	Check that the job list is displayed. Check that job cards show pickup address, dropoff address, volume, date and time, job type, and credits.
2a. No unassigned jobs exist	Navigate to Find Jobs screen.
2a1. Display to mover that there are no jobs available	Check that the text "No available jobs" is displayed. Check that 0 job cards are present on screen.
3. Mover can click toggle to switch jobs displayed from "Show All" to "Within Availability"	Navigate to Find Jobs screen. Check that text "Show All" is displayed. Check that 2 job cards are displayed. Click the availability toggle. Wait for filtering to apply.
4. System displays unassigned jobs only within mover's availability time windows	Check that te0xt "Show All" does not exist. Check that text "Within Availability" is displayed. Check that only 1 job card is displayed.
4a. No unassigned jobs exist within mover's availability	Click the availability toggle. Wait for filtering to apply.
4a1. Display to mover that there are no jobs available currently with suggestion to broaden their availability	Check that text "No available jobs within your availability" is displayed. Check that 0 job cards are present on screen.
5. Mover can optionally accept the job (see UC-4 "Accept Job" use case)	[See UC-4 "Accept Job" test case]

- **Test Logs:** (for all tests in file)

```
Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Pixel_7(AVD) - 13 Tests 1/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13
```

- **Use Case: Accept Job**

  Expected Behaviors:

Scenario Steps	**Test Case Steps**
1. Mover clicks on “Accept” for the corresponding job they’d like to accept.	Click the "Find Jobs" tab. Verify the "Accept" button is displayed, enabled. Record the initial number of Jobs. Click the "Accept" button.
2. System assigns the job to the mover, Firebase Cloud Messaging Service notifies the student who created the order that it has been accepted, and a live update occurs for the student’s order status.	After acceptance, navigate to the "Current Jobs" tab.
3. Mover sees the job listed under “Current Jobs”	Confirm the accepted job appears in Current Job List and the job status is shown. Return to "Find Jobs", wait for the list to refresh, and assert that the number of Job entries decreased by 1 compared to the recorded initial count.

- **Note:** Before running this test, manually log in as a student and create an order to ensure there is at least one unassigned job available for acceptance.

- **Test Logs:**

```
Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Pixel_7(AVD) - 13 Tests 1/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13
```

- **Use Case: View Recommended Route**

  Expected Behaviors:

Scenario Steps	Test Case Steps
1. Mover clicks on "Get Optimal Route" button	Navigate to Find Jobs screen. Click the "Get Optimal Route" button.
2. System prompts mover for their desired max shift duration	Check that the duration selection screen is displayed. Check that the duration slider is present.
3. Mover enters their max duration by using the slider and clicks "Find Smart Route" button	Use the slider to select desired shift duration (e.g., 4 hours). Click the "Find Smart Route" button.
4. System prompts mover to allow access to their current location	Check that Android location permission dialog appears.
5. Mover allows access to location by pressing "Only this time" or "While using the app" button	Click "While using the app" or "Only this time" button in the permission dialog.
5a. Mover does not allow access to location	Navigate to Find Jobs screen. Click "Get Optimal Route". Click "Find Smart Route". Click "Don't allow" or "Deny" button in the permission dialog.
5a1. Display that location permission is required with suggestion to grant permission	Check that "Location Permission Required" message is displayed. Check that explanation text "We need your location to calculate the optimal route" is shown.
6. System displays a route summary and a list of jobs within the mover's availability and maximum duration window, with details of the pickup address, travel time and credits	Check that route summary is displayed. Check that total earnings is shown. Check that total duration is shown. Check that number of jobs is shown. Check that the job list is displayed. Check that each job shows pickup address, travel time, and credits.
7. Mover can accept all (using “Accept all jobs” button) or some of the suggested jobs (“Accept job” button)	Check that "Accept all jobs" button is displayed. Click "Accept all jobs" button. Navigate to Current Jobs tab. Check that current jobs list is displayed. Check that multiple jobs now appear in Current Jobs.

- **Test Logs:**

```
Pixel_7(AVD) - 13 Tests 0/1 completed. (0 skipped) (0 failed)
Pixel_7(AVD) - 13 Tests 1/1 completed. (0 skipped) (0 failed)
Finished 1 tests on Pixel_7(AVD) - 13
```

  For deny location test: (fails due to location not being denied properly)

```
   java.lang.IllegalStateException: No compose hierarchies found in the app. Possible reasons include: (1) the
Activity that calls setContent did not launch; (2) setContent was not called; (3) setContent was called before
the ComposeTestRule ran. If setContent is called by the Activity, make sure the Activity is launched after the
ComposeTestRule runs
```

- **Use Case: Set Availability**

  Expected Behaviors:

Scenario Steps	Test Case Steps
1. Mover clicks on "Availability" on the navigation bar	Navigate to Availability screen.
2. System displays a list of days of the week with option to add time slots for each day	Check that the days list is displayed. Check that each day shows a "+" button.
3. Mover clicks "+" button for a specific day	Click the "+" button for desired day (e.g., Monday).

Scenario Steps	Test Case Steps
4. System displays a dialog to enter start time and end time	Check that "Add Time Slot" dialog is displayed. Check that start time input field is present. Check that end time input field is present.
5. Mover enters start time and end time	Enter start time (e.g., "09:00"). Enter end time (e.g., "17:00").
5a. Mover enters invalid time	Enter invalid time (e.g., "25:00").
5a1. System displays error message for invalid format	Check that error message "Use HH:mm format" is displayed.
5b. Mover enters start time that is after end time	Enter start time (e.g., "17:00"). Enter end time (e.g., "09:00").
5b1. System disables the "Add" button	Check that "Add" button is disabled.
6. Mover clicks "Add" to confirm the time slot	Click "Add" button in dialog. Check that the time slot appears in the day's list.
7. Mover repeats steps 3-6 for other days as needed	Add time slots for additional days. Check that all time slots are displayed.
8. Mover can add multiple time slots to the same day	Click "+" button for the same day again. Enter start and end times. Click "Add" button. Check that both time slots appear for the same day.
9. Mover can remove a time slot	Click delete button next to a time slot. Check that the time slot is removed from the list.
10. Mover clicks "Save Availability"	Click "Save Availability" button.
11. System saves the availability and displays success message	Check that "Availability updated successfully!" message is displayed.

- **Test Logs:**

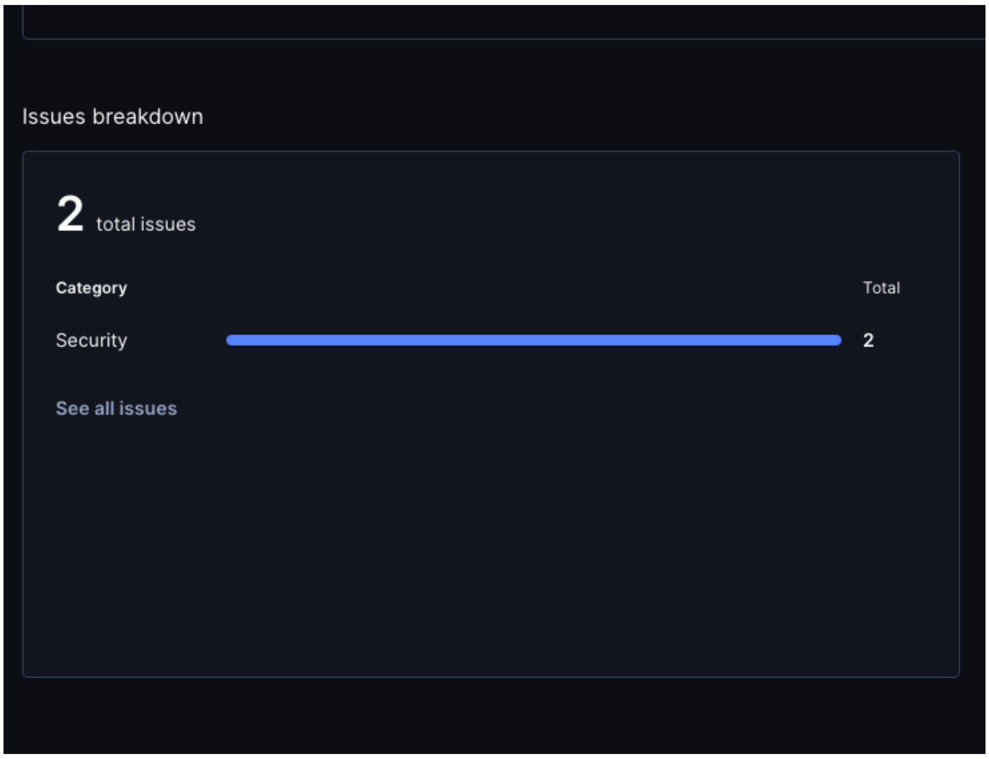
```
    > Task :app:connectedDebugAndroidTest
    Starting 5 tests on Pixel_7(AVD) - 13

    Pixel_7(AVD) - 13 Tests 0/5 completed. (0 skipped) (0 failed)
    Pixel_7(AVD) - 13 Tests 1/5 completed. (0 skipped) (0 failed)
    Pixel_7(AVD) - 13 Tests 2/5 completed. (0 skipped) (0 failed)
    Pixel_7(AVD) - 13 Tests 3/5 completed. (0 skipped) (0 failed)
    Pixel_7(AVD) - 13 Tests 4/5 completed. (0 skipped) (0 failed)
    Finished 5 tests on Pixel_7(AVD) - 13

    BUILD SUCCESSFUL in 1m 35s
    76 actionable tasks: 9 executed, 67 up-to-date

    Build Analyzer results available
```

5. Automated Code Review Results

5.1. Commit Hash Where Codacy Ran

`4a3c52bb44685d50ef99052dcdda729e25068bc2`

5.2. Unfixed Issues per Codacy Category



Issues breakdown

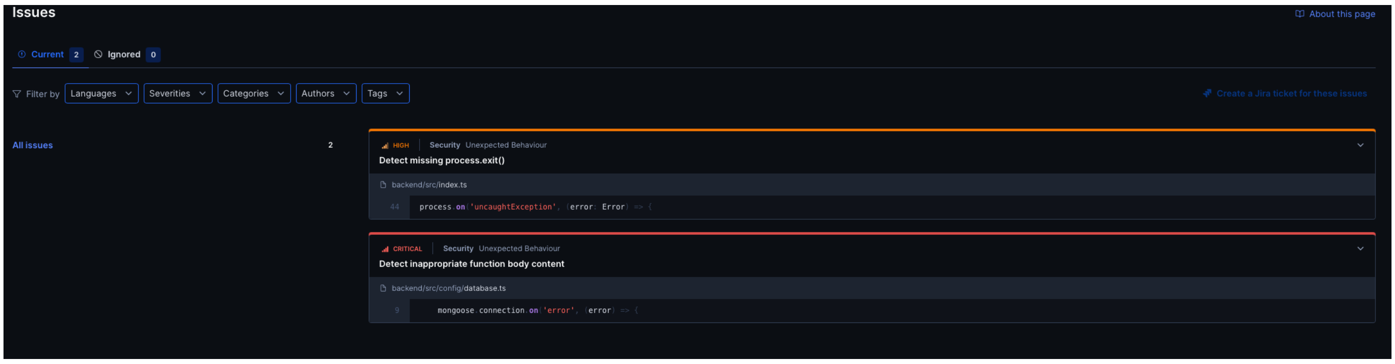
2 total issues

Category		Total
Security		2

See all issues

Figure: Codacy Issues breakdown (Security issues count)

5.3. Unfixed Issues per Codacy Code Pattern



Issues

All issues 2

HIGH | Security Unexpected Behaviour
Detect missing process.exit()
backend/src/index.ts

```
44    process.on('uncaughtException', (error: Error) => {
```

CRITICAL | Security Unexpected Behaviour
Detect inappropriate function body content
backend/src/config/database.ts

```
9        mongoose.connection.on('error', (error) => {
```

Figure: Codacy Issues Page, per category

5.4. Justifications for Unfixed Issues

- **Code Pattern: No pattern, just says 'All issues'**
  1. **'Detect missing process.exit'**
     - **Location in Git:** `backend/src/index.ts#L44`
     - **Justification:** false positive, we call process.exitCode=1. In the Nodejs documentation, It says it will exit when the even loop empties with the exit code specified.
  2. **'Detect missing process.exit'**
     - **Location in Git:** `backend/src/config/database.ts#L9`
     - **Justification:** false positive, the solution says to add an error listener, however the error is on an error handler. We are following guidelines same as GeeksforGeeks - Mongoose Connections guidelines.

  Justification for not pursuing fixes any longer: 'Steve McConnell, Code Complete: "Industry average is about 15 - 50 errors per 1000 lines of delivered code."' Our code has a KLoc of 0.43, which is much higher than industry standard.