

Cycle Servicing – at your doorstep/workstation

Objective:

Design the database schema and REST API endpoints which will facilitate the customer to book servicing appointments from cycle mechanic at their doorstep or at the workshop

Problem Definition:

Create an rest application for an Cycle Mechanic that will visit the customer's house and fix the cycle. The mechanic will carry all the spare parts along with him.

Below are the fees charged by the mechanic are as below

Service Cost at station	Cost
Chain Degreasing & Lubrication	200
Bike wash: Pressure washing and drying of bicycle, drivetrain maintenance (chain degreasing, cleaning and lubrication)	300
Standard Service: (14 checkpoints, tuning, alignment and drivetrain maintenance)	500
Advance Service: (Standard Service + Complete strip down of bike and components & service of each component + Bike wash)	1300

Service Cost at doorstep	Cost	Transport Fees
Chain Degreasing & Lubrication	200	200
Standard Service: (14 checkpoints, tuning, alignment and drivetrain maintenance)	500	200

When taking online booking allow user to select either **workstation service** option or at their **doorstep service**.

Once a User requests for a booking for a selected date and time. The user will get a response saying his booking is received and confirmation of the booking will be notified via email or sms. He will also receive a unique booking id, using which he can check his booking status manually

The admin user will check the availability of the service technicians and then approve the service appointment.

For doorstep service an extra charge of Transportation of Rs 200 is levied on the customer.

Timings of workstation:

9:00 AM to 1:00 PM – Full Working hours

2:00 PM to 6:00 PM – Full Working hours

1:00 PM to 2:00 PM – Lunch Break

Entities:

Customer

- FirstName
- LastName
- Mobile
- Instagram Username
- Email
- Password

Expected API Endpoints:

Simple Endpoints:

- 1) API endpoint to get list of available services at workstation.
- 2) API endpoint to get list of available services at doorstep.
- 3) API endpoint to register a new customer
- 4) API endpoint to book appointment
- 5) API endpoint to approve the appointment
- 6) API endpoint to check the status of the appointment
- 7) API endpoint to list all the appointments for the given day at workstation
- 8) API endpoint to list all the appointments for the given day at doorstep

NOTE:

- **Work on Each task individually, don't try to complete all the tasks together.**
- **You are advised to commit your changes once each task is complete and push to GitHub.**

While committing your changes put proper commit messages.

Example: If you are working on "Creating the CRUD Endpoints for customer " and you have completed writing the code for adding customer. Then the commit message should be "TASK-CRUD Customer::Wrote end point to Create customer"

Format is shown below:

"TASK-TASK-NAME>>::<<COMMIT_MSG>>"

Expected output:

- All the expected API Endpoints should be documented in the Postman Collection.
- Invoking the Postman API endpoint should give the expected results.

Tech Stack:

The following tech stack should be used to implement the task.

- **Framework :** NodeJS/NestJS
- **Database :** MongoDB/mysql/Postgres
- **API Documentation :** Postman