WheelsWell

Batool Shilleh
Manar Jaber

Supervisor: Eng.Dr.Haya Samaana

An-Najah National University

Department of Computer Engineering

Software Graduation Project (GP I)

This dissertation is submitted for the degree of

Computer Engineering

Sep 2023

Acknowledgements

Praise be to Allah, the Lord of the Worlds.

Those who do not thank people do not thank God. I express my gratitude to everyone who contributed to the completion of this project, including our beloved university, professors, teaching assistants, and fellow students who assisted us in its completion.

Special thanks to the esteemed Dr. Haya Samaana for her invaluable efforts.

We also appreciate our families and individuals who supported us mentally to overcome the challenges of the difficult circumstances and move forward despite the hardships.

To the souls of our martyrs, who are more honorable than all of us, we extend our heartfelt gratitude.

DISCLAIMER

Abstract

In the contemporary landscape of the West Bank, the proliferation of cars and transportation methods has led to a scenario where nearly every individual possesses at least one vehicle. This burgeoning automotive presence underscores the undeniable importance of maintenance, cleaning, and other associated services for these vehicles. For those who lack prior experience in navigating the intricate world of automobiles, whether it be sourcing parts, identifying a suitable technician for maintenance, or conducting routine inspections, the prospect can appear overwhelming. However, a transformative solution is at hand in the form of a dedicated mobile application.

This innovative application aims to address the multifaceted needs of car owners, offering a comprehensive repository of services encompassing inspections, maintenance, and detailing, even including painting and aesthetic enhancements. By aggregating all essential automotive crafts in one centralized platform, the application serves as an invaluable resource for users. It goes a step further by tailoring its recommendations based on the user's location, ensuring that each individual is presented with the most suitable solutions.

The application not only simplifies the process of accessing essential services but also fosters interaction with industry experts, providing users with a seamless and informative experience. Moreover, by incorporating advanced location-based filtering, the application extends its reach beyond the user's immediate vicinity, showcasing services and products available in other cities.

One of the standout features of the application is its real-time tracking capabilities, allowing users to monitor the status and location of their received goods or ongoing services. This feature not only enhances convenience but also optimizes time management and overall user comfort. In essence, this application revolutionizes the way car owners in the West Bank approach the upkeep and enhancement of their vehicles, offering an integrated and user-centric solution to meet their diverse automotive needs.

This project has been developed utilizing a technology stack that includes Flutter and Dart for the frontend, complemented by Node.js and MongoDB for the backend. It follows the Model-View-Controller (MVC) architectural pattern to ensure a robust and organized software structure

1. Introduction

1.1 Problem Statement

WheelsWell is an electronic system that manages services related to automobiles:

1. As the automotive sector is vast, and with the majority of people now owning private cars and the widespread use of the internet and electronic services, it has become essential to provide a platform that consolidates the services and products of the automotive sector in one place. This facilitates efficient exchange of services and goods between vehicle owners, shops, and service providers.
2. It provides a secure and conducive environment for the exchange of services and goods.

1.2 Significance

This project aims to facilitate the exchange of goods and services between the supplier and the customer, saving time and effort and making the process easy, quick, and satisfactory for both parties

1.3 Objectives and Scope

The project provides several advantages, including:

- Assisting business owners in creating a workspace where they can store all their services and products, forming a comprehensive documentation of the company's or institution's activities.
- Contributing to a broader dissemination of various services and products.
- Providing users with classification and display of high-quality products or services.
- Offering consultation services to ensure that users receive their requests with high quality

1.4 Report Organization

2. Constraints, Problems and Earlier Coursework

2.1 Constraints

2.1.1 Time Constraints

The project is registered in a semester with academic pressure, and it must be implemented within its scope. Therefore, this project cannot solve some problems and tasks related to the project's scope.

2.1.2 Implementation Constraints

This project is still considered educational for computer engineering students. So to perform it; the contributor should build it from least complex software, and implement the full backend programming with a database connection, and can’t rely on the complete backend solutions like the Firebase or AWS.

2.1.3 Team Constraints

Due to the challenges faced by the country, the geographical distance, and issues with internet connectivity, coupled with the changes experienced during the semester, work processes and meetings have been somewhat difficult and require better pre-organization.

2.2 Challenges

2.2.1 Lack of pre-experience

Dealing with Flutter for the first time was challenging initially, as the front-end, designed using Flutter and Dart, had to seamlessly integrate with the back-end using Node.js. This required studying Flutter widgets, reading their documentation, and learning how to interact with databases and link them to the back-end interface

2.2.2 Coding challenges

The project contributors are simply computer engineering students; so he hasn't a lot of coding problem solving techniques before that project.

2.2.3 Hardware Challenges

Initially, utilizing tools like simulators and application development software posed challenges on the current hardware. This required upgrading the existing hardware, such as increasing RAM and storage capacity.

2.3 Earlier coursework

2.3.1 Computer Programming(C/C++)

These classes covered the fundamentals of programming and introduced the most basic concepts of the programming.

2.3.2 Data Structures and Algorithms

This course has provided us with how to work with the data and organize it in the memory; which is a very important process in programming.

2.3.3 Critical Thinking and Research Skills

This course has taught us how to conduct research and write a report, and it's one of the few non-technical courses that is also lifetime.

2.3.4 Object Oriented Programming

This course has taught us how to deal with Object Oriented Programming, which is the most common programming paradigm nowadays, and it's used to get and leverage high code reusability.

2.3.5 Database Management Systems

This course has provided us with how to deal with databases to store data in a way that it's easily accessed. And provided us with data modeling processes and how the data shape stored. Also provided us how to query data and apply CRUD operations on the data in the needed. It also provided us with how we should connect with the database and work with it in a program that I program with a programming language. It also provided us with the basics of the ORM which is heavily used in almost every framework nowadays.

2.3.6 Web Programming

This course has provided us with the basics of the web, and how the web works. It also gave us a full overview of the server side codes and database usage in the web servers.

2.3.7 Distributed Operating Systems (DOS)

This course has provided us with the basics of communication protocols and the architectural styles

2.3.8 Software Engineering

This course has provided a full overview of how software development processes. It also provided us with tools that can be used in programming like Git and GitHub.

3. Literature Review

When starting the project and given the current country conditions, our primary focus was directed towards incorporating as many features as possible to make it stronger and more efficient. We opted for traditional methods in booking and accessing car services, especially for women. It was essential to have a means of organizing these services efficiently and presenting them in a simple and user-friendly manner. Additionally, there needed to be a straightforward and suitable way for service and product providers to showcase their offerings easily.

With our system, dealing with these matters becomes easy, organized, and accessible, allowing for effortless navigation and us.

4. Methodology

This chapter contains detailed information about the techniques and methods used to develop the project, from choosing the idea and starting the basics, through build the front end and back end control.

4.1 Choosing the idea

We initially started by gathering a large number of ideas based on societal needs or our own desires. We filtered them into three projects to be comprehensive, well-rounded, and containing enough features. However, upon analysis and scrutiny, deficiencies became apparent.

The first project involved the concept of creating a low-featured hosting website, lacking innovation and not sufficient to showcase students' skills. The second idea revolved around a football-related entertainment app, but it was somewhat mundane and needed new and distinctive features.

The third idea centered around an educational language learning app, which was somewhat basic and lacking creativity. We sought the guidance of our professor to help us arrive at a sufficient and comprehensive idea that would effectively showcase the students' abilities and serve as a suitable graduation project.

We eventually settled on the idea of a specialized car system, originally a course project with fewer features. With some additions and improvements, we developed an excellent and beneficial concept for society. This was achieved by considering the target audience and studying the current market for cars, ensuring the project is comprehensive and provides good services that meet users' needs. It became a suitable project to represent graduating students

4.2 Easy Internet Market Analysis

(References for this section are listed at the end of this report)

Following the idea's receipt, it was determined to conduct online market research in order to gauge the idea's demand and make any necessary improvements. The research's findings are as follows:

The progression of internet service usage following the Covid-19 pandemic:

Before the Covid-19 pandemic struck in 2020, the majority of people worldwide viewed online services as unreliable and untrustworthy. However, in 2020, people were forced to use online services due to prevailing health-related circumstances brought on by the Covid-19 virus (Corona pandemic), which resulted in a hundredfold increase in the usage of these services.
Following that, people realized how strong and distinctive these services could be.
Based on the aforementioned, it is evident that the use of online services has increased in an unexpected manner, indicating a genuine need for a system to arrange them.

The need for innovation in the modern world:

With the rapid advancement of technology, nearly 86% of institutions, businesses, and organizations in the world are looking for new ideas and ways to stand out from the competition. The innovators are then granted access to the entire demand, the entrepreneurial culture, and the next generation of technology. The primary demand of the digital age is then met by innovation as well.

4.3 Features and Roles Selection

The features and roles were selected as a result of research and discussion with the supervising doctor, as well as a study of the labor market, and the following agreements were reached:

4.3.1 Features

The application provides a favorable environment for the exchange of goods and services between customers and companies, offering several features that create an efficient platform:

1. Ease of Ordering and Reservation:

- Users can easily reserve a time slot or request specific products.

- The application allows users to control reservation timings and order quantities through user-friendly interfaces for both ordering and booking.

2. Ordering Process:

- Users can place orders for single or multiple items, adding them to the cart and calculating the payment value.

- Payment methods, either cash or card, can be selected by the user.

3. Booking Process:

- Users can choose a convenient time slot based on the company's availability.

- Payment methods, cash, or card can be selected, and users receive notifications as the reserved time approaches.

4. Delivery Tracking:

- Users can track the delivery progress on a map, monitoring the location of the delivery in real-time.

5. User Account Management:

- Users can manage their account information, such as name, location, and password, to enhance their overall experience.

6. Complaints and Feedback:

- Users can submit complaints regarding service issues or inquire about any glitches.

- Feedback can be directed to the application administrator for system-related issues or specific suggestions.

7. Additional Features on the Home Screen:

- The application offers simple services on the main interface, such as traffic signals for studying traffic regulations, useful links, and resources.

- It provides a dynamic meter display according to the user's location.

8. Product Ratings and Reviews:

- Users can rate and review products and services, enabling others to view customer feedback.

- Users can mark favorite services or products for future reference.

9. Company Benefits:

- The application facilitates efficient storage and documentation of the company's services and products.

10. Order and Reservation Management:

- Companies can manage orders and reservations, including details like timing and status (pending or completed).

11. Inventory Management:

- The application features inventory management, providing information on available stock quantities.

12. Company Profile:

- A dedicated company profile feature allows businesses to showcase their services and products effectively.

In conclusion, the application offers a comprehensive set of features for both users and companies, enhancing the overall experience of goods and services exchange.

4.3.2 Roles

To ensure safety and protection, it is essential to distribute roles and grant users specific permissions to prevent interference or unauthorized alterations to critical information. However, increasing the number of roles can lead to a complex and extensive application, requiring more precise and extensive permission management. Therefore, we have decided to categorize roles into just three roles:

1. Administrator:
   - Responsible for crucial tasks such as accepting company membership requests and handling user complaints.
   - Has overarching authority and control over the entire application.

2. Companies:
   - Tasked with uploading products and services, ensuring their integration, and managing their workflow.
   - Cannot access pages for adding products unless approved by the Administrator.

3. Users:
   - Individuals who place orders and browse services and products.
   - Do not have access to product addition pages.

This simplified role structure is designed to streamline the application's functionality and avoid unnecessary complexity. The Administrator plays a central role in managing and overseeing both companies and users. Companies, in turn, focus on product/service management, while users are primarily engaged in placing orders and exploring available offerings.

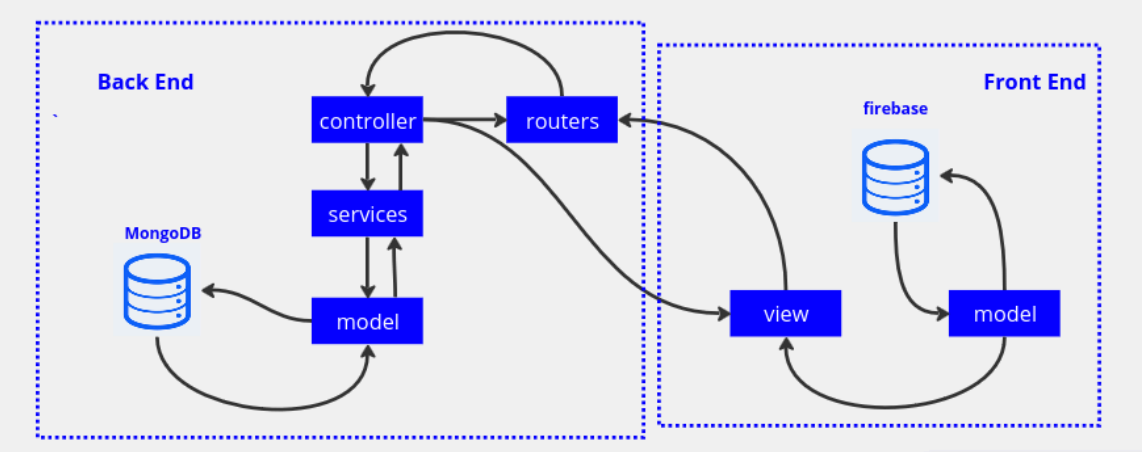
The Administrator's role is pivotal in maintaining a secure and organized environment, and companies can only join the platform upon approval from the Administrator. This

approach aims to strike a balance between security and user-friendly functionality.

4.4 Technologies Selection

The technologies used in the project were chosen based on the nature of the project, the experience of the implementers, and the project's scale.

System Design and Architectural Style:
Given the nature and purpose of the application, we opted to build it using a cross-platform language to make the application work on multiple platforms as needed by the user. Regarding the architectural aspect, we initially considered Microservices, but after studying the application's edition, workflow, and division, it became apparent that using this architecture would result in a complex application with a vast number of services. Therefore, we shifted towards the Model-View-Controller (MVC) architecture and divided the application into server and front-end components, with communication between them using RESTful APIs.



Backend:
We chose to use Node.js due to our familiarity with it and its significance in the job market. It was implemented with the Express.js framework.

Frontend:
For the frontend, we chose to build it using the Flutter framework for the following reasons:

- Cross-Platform Development:

Flutter allows writing applications for both Android and iOS using a single codebase, reducing the effort required to develop for multiple platforms.

- Beautiful and Consistent User Interface:

Flutter enables the creation of a beautiful and engaging user interface using a variety of widgets and animations. It ensures a consistent user experience across different systems.

- High Performance:

Flutter utilizes the embedded graphics engine (Skia), contributing to high and smooth performance for mobile applications. Graphics can be executed quickly at the system

level without the need for a Native bridge.

- Rapid Development:

Flutter provides a fast development experience with Hot Reload, allowing developers to see the impact of changes in real-time without restarting the application.

- Active Community:

Flutter has an active and diverse community of developers, providing many online resources and support to help developers solve problems and effectively use Flutter.

- Good Documentation Support:

Flutter offers comprehensive and detailed documentation, making it easy for developers to understand how to use the framework and troubleshoot issues.

- Advanced Feature Support:

Flutter supports features like Firebase for adding cloud services and tools for performance analysis and continuity.

IF YOU NEED	FLUTTER	REACT NATIVE
Quick prototype of the app	✓	✓
MVP of the app	✓	✓
App with many screens and complex logic	✓	✓
App that looks great even on old devices	✓	✓
Desktop application	✓	✗
Web application	✓	✓
Game application	✓	✗
App with AR elements	✗	✗

Database:

For the database, we chose MongoDB, a document-oriented database, due to its advantages such as flexibility in storage, integration with programming languages, scalability, good performance, and suitability for web and mobile applications.

4.5 Coding Bases Preparation

This project uses two distinct code bases—one for the front end and another for the back end—to write its code. Each code base is hosted on a private Github repository and has its own Git repository. However, the database needs to be correctly installed and configured before you can begin working on them. This is an example of how to prepare the database and work with the two code bases.

4.5.1 Database Installation and Preparation

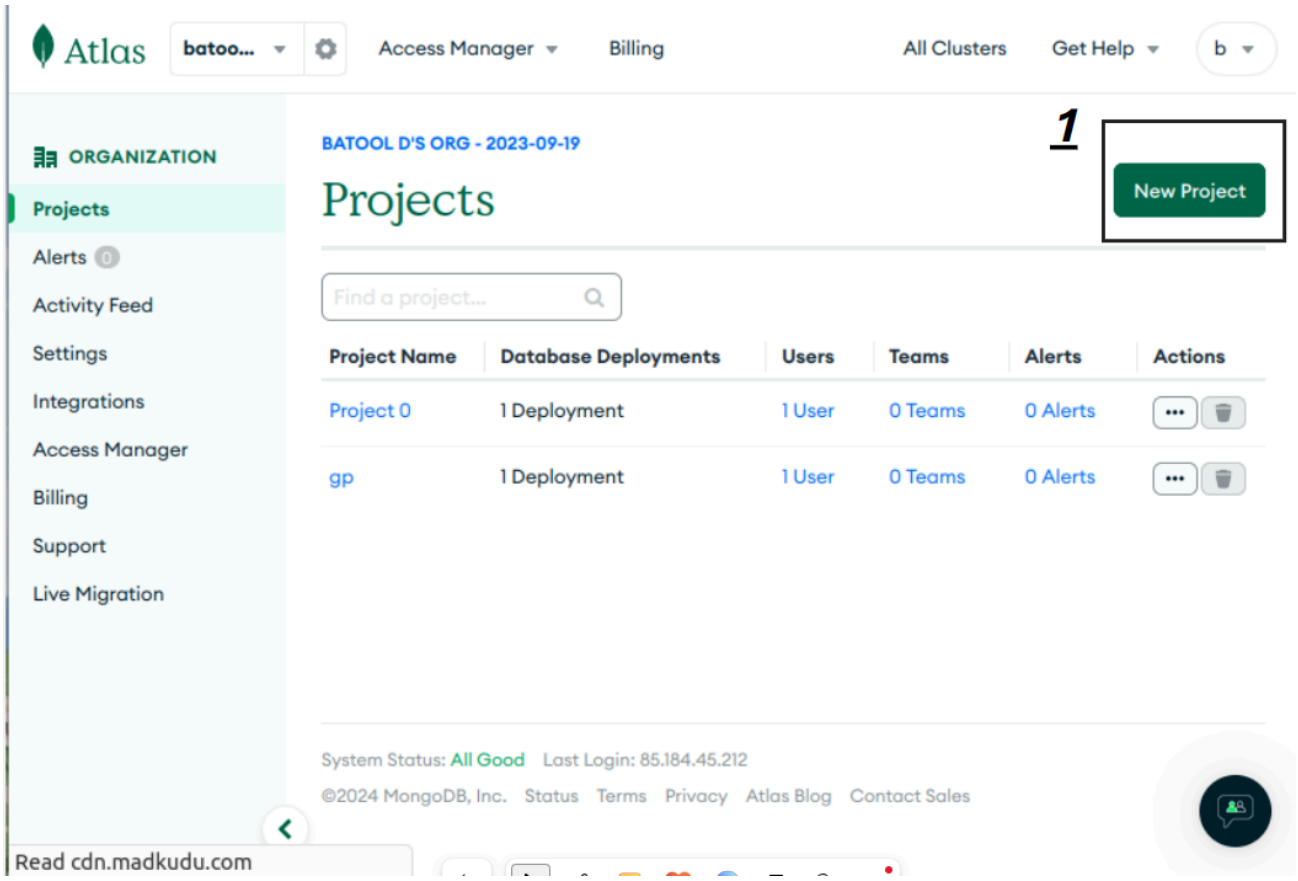
To ensure the use of a shared database between the two students, making the data synchronized and organized, we will utilize the 'Atlas' service. This service provides cloud storage for data, and the accounts of both students have been added to it. The working environment in Node.js has been set up to guarantee the storage of data and access to it from any device. Here are these steps in detail:

Setting up Atlas Environment:

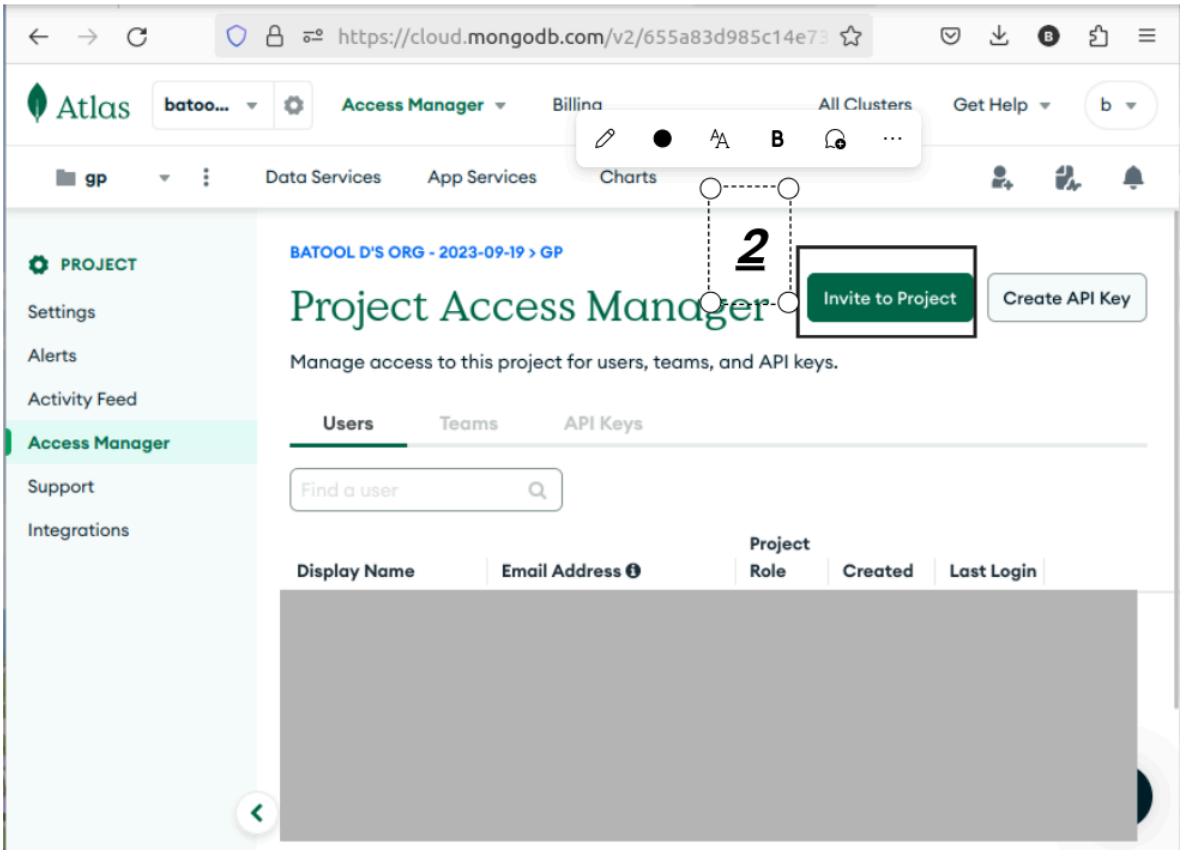
By visiting the Atlas website (link is available in the references section at the end of this report).

Create an Account

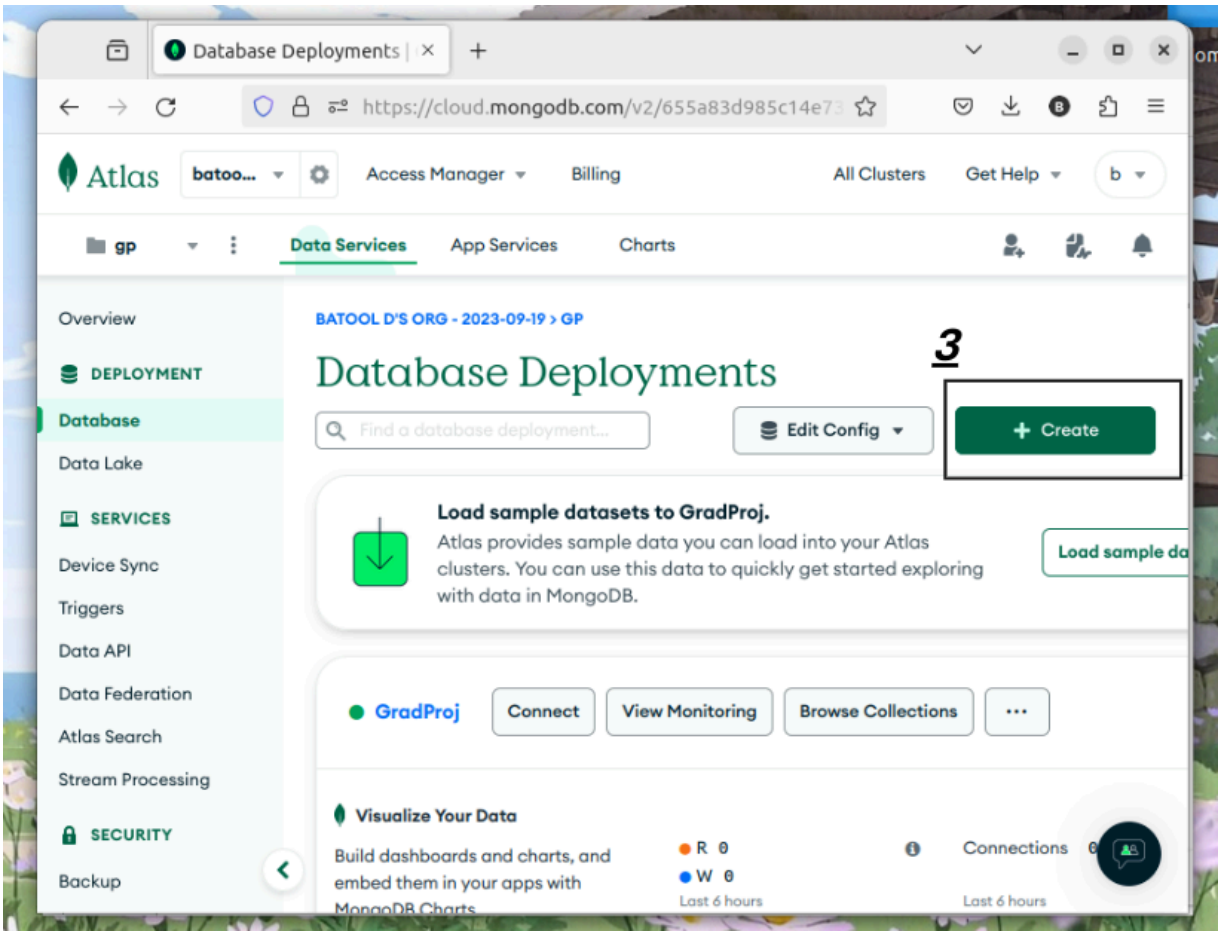
Create a Project



Configure Security and Access



Create Database

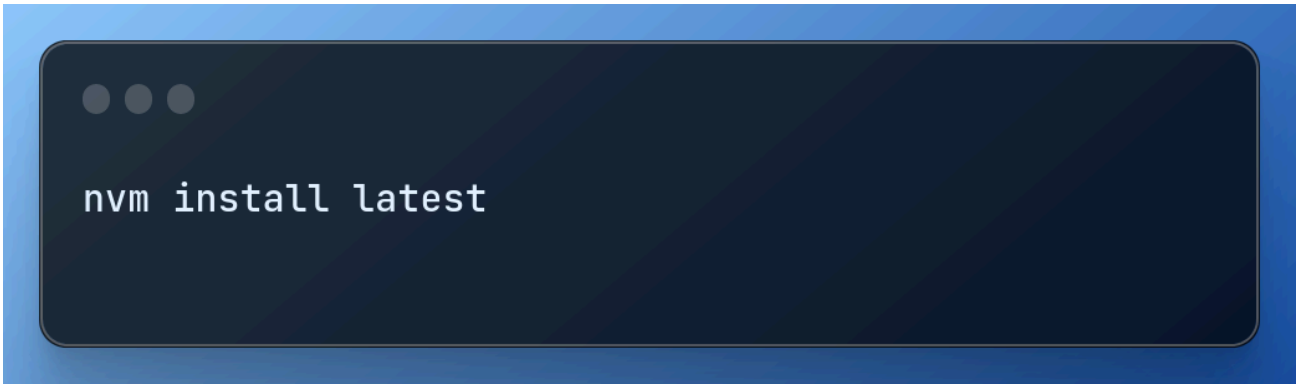


Add Collections:

The process of adding a table is done using the backend interface, and we will discuss that in its section

4.5.2 Back-end Code Base Preparation

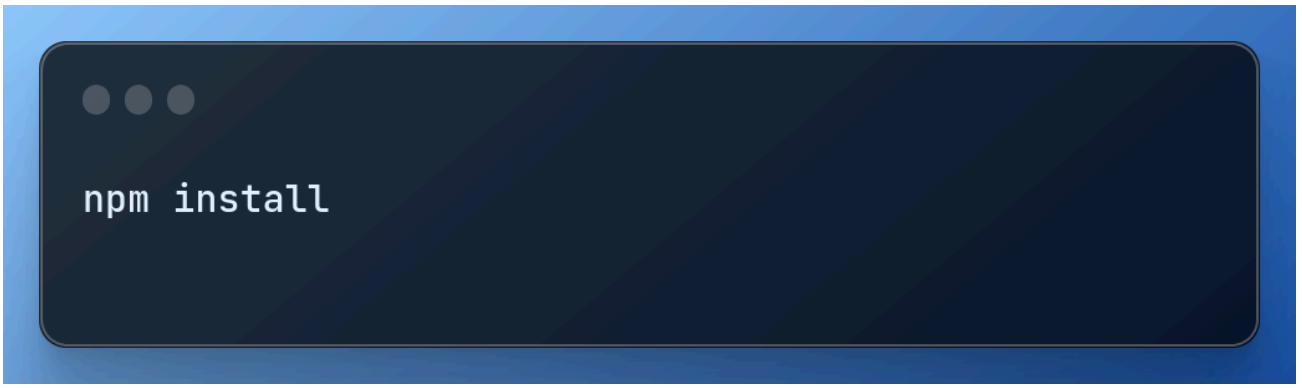
Before starting the encoding process, Node.js must be installed and nvm must be set up.



Setting up Visual Studio Code to start the project:

1- Install Dependencies:

Running 'npm install' installs the project dependencies listed in the 'package.json' file, ensuring the accurate versioning and updating, if necessary, of the modules. It also generates or updates the 'package-lock.json' file for version consistency within the project.

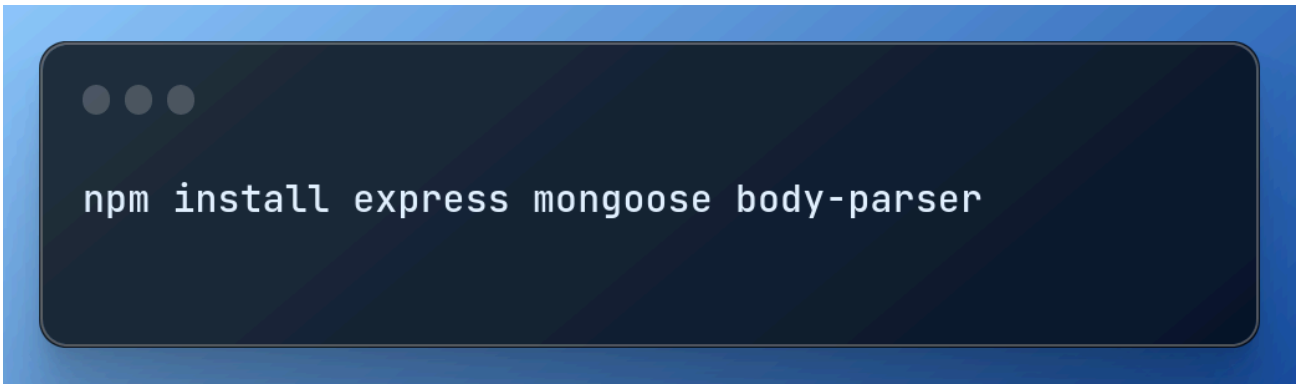


2- Enhancing Node.js Project Capabilities with npm Dependencies:

1. Express:
   - Express is a robust web framework that simplifies web application development, providing essential tools and features for building scalable and efficient servers.

2. Mongoose:
   - Mongoose serves as an interactive module designed for seamless interaction with MongoDB databases. It facilitates the creation of data models, execution of queries, and control over validation processes.

3. Body-Parser:
   - Body-Parser, a middleware, plays a vital role in processing HTTP request body data. It ensures efficient handling of data, particularly essential when dealing with POST and PUT requests.

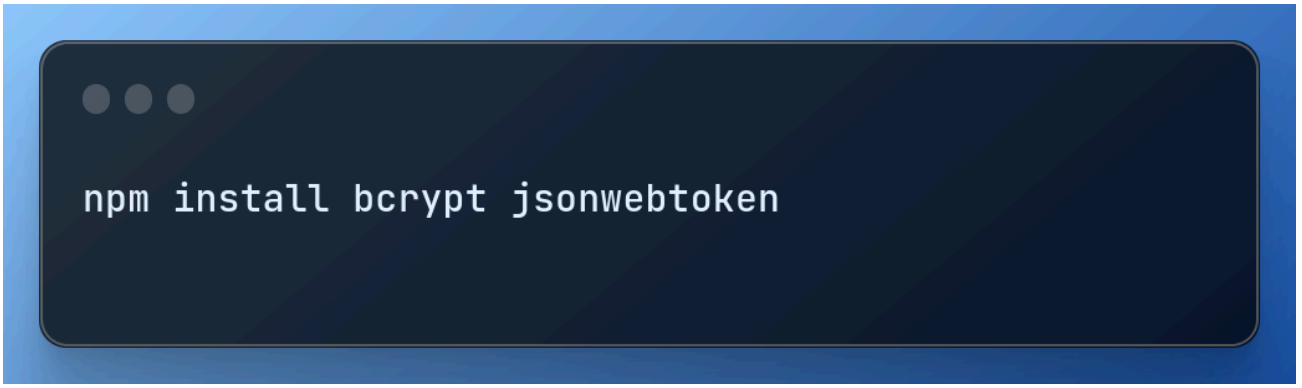


Executing this command not only installs these dependencies but also updates the 'package.json' file, incorporating these modules as dependencies. This integration significantly enhances the functionality and capabilities of the Node.js application, enabling it to handle web development tasks and interact effectively with MongoDB databases.

3- Enhancing Security and Authentication in Node.js Applications:

1. bcrypt:
   - `bcrypt` is a powerful module employed for password hashing. By incorporating bcrypt, the project can implement robust encryption techniques, significantly improving the security of stored passwords in the database.

2. jsonwebtoken:
   - `jsonwebtoken` provides a seamless way to generate and verify signatures for JSON Web Tokens (JWTs). JWTs are widely used for authentication mechanisms, securely exchanging information between parties.



Executing 'npm install bcrypt jsonwebtoken' not only installs these modules but also updates the 'package.json' file, adding them as dependencies. These modules are commonly utilized in Node.js applications handling identity verification and data encryption, such as login functionalities and session management. The integration of bcrypt and jsonwebtoken enhances the overall security posture of the Node.js application, ensuring a robust and secure environment for user authentication and data protection.

4- Optimizing Node.js Development Workflow with DevDependencies

Nodemon as a DevDependency:

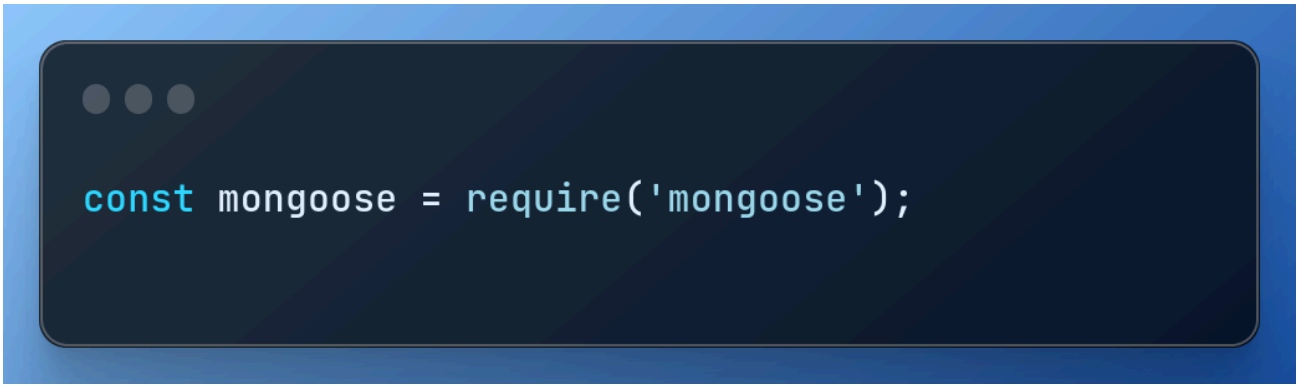
- Nodemon, a prominent development tool, is seamlessly integrated into the project as a DevDependency. By appending '--save-dev' to the installation command, nodemon is exclusively categorized under 'devDependencies' in the 'package.json' file. This designates it as a crucial component for the development and testing phases but excludes it from the essential dependencies required for production.

Executing 'npm install nodemon --save-dev' not only installs nodemon but also updates 'package.json', reflecting its status as a development dependency. Leveraging nodemon in this manner streamlines the development workflow, automating server restarts upon code changes, and contributing to a more efficient and agile development lifecycle.

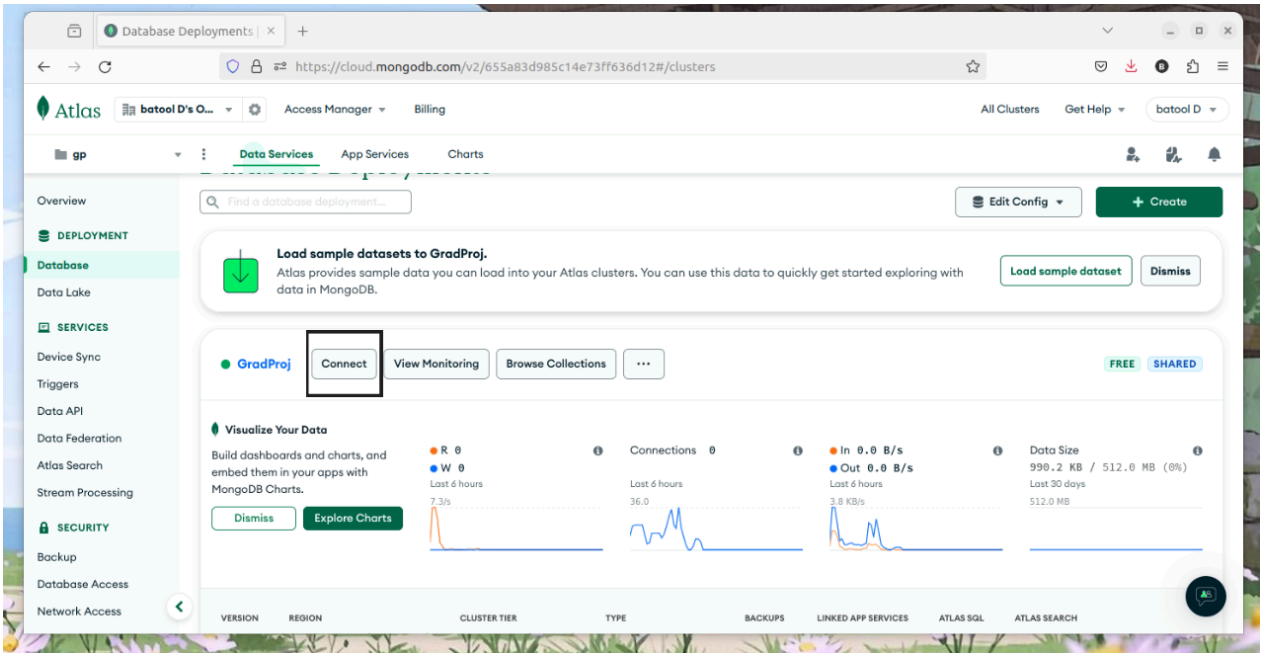
This strategic use of DevDependencies ensures that tools like nodemon, specifically tailored for development convenience, are excluded from production deployments, thereby optimizing the overall project structure and reducing unnecessary bloat in production environments.

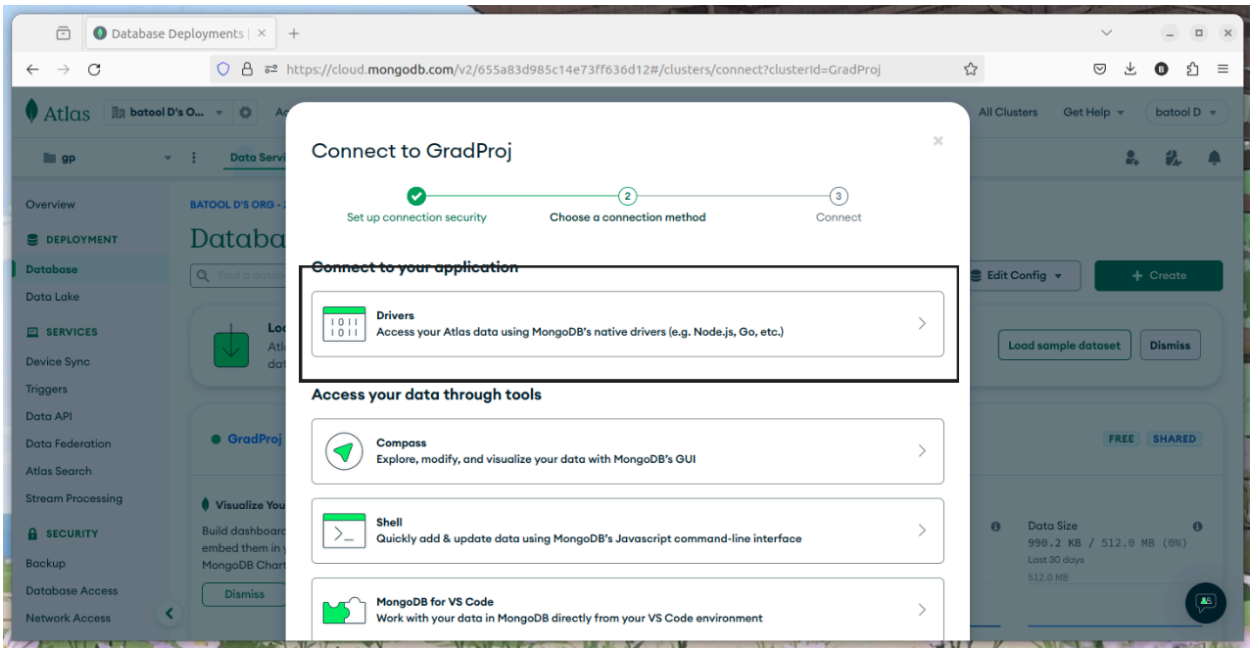
Connect with Database:

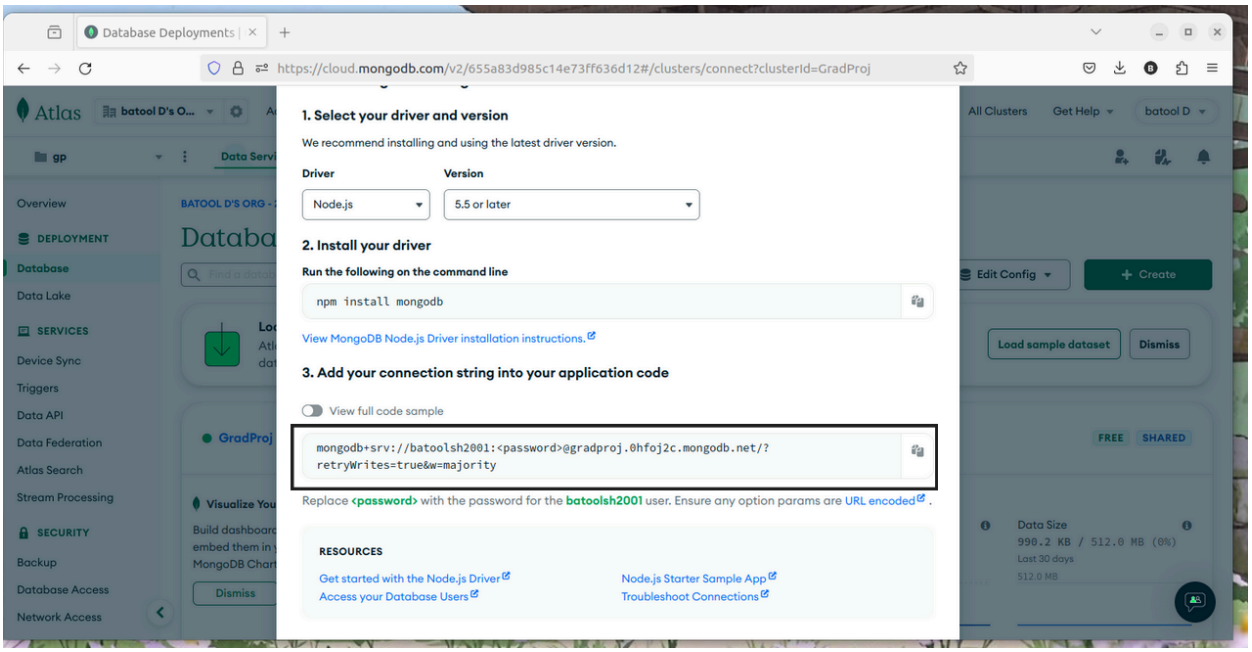
1. Importing Mongoose Library:
   The Mongoose library is imported and stored in the `mongoose` variable for use in the code.



2.Database URI:
   A variable named `uri` is defined, containing the URI (Uniform Resource Identifier) address used to connect to the MongoDB Atlas server, which is a cloud-based MongoDB database hosting service.


Database Deployments
https://cloud.mongodb.com/v2/655a83d985c14e73ff636d12#/clusters
Atlas
batool D's O...
Access Manager
Billing
All Clusters
Get Help
batool D
gp
Data Services
App Services
Charts
Overview
DEPLOYMENT
Database
Data Lake
SERVICES
Device Sync
Triggers
Data API
Data Federation
Atlas Search
Stream Processing
SECURITY
Backup
Database Access
Network Access
Edit Config
Create
Load sample datasets to GradProj.
Atlas provides sample data you can load into your Atlas clusters. You can use this data to quickly get started exploring with data in MongoDB.
Load sample dataset
Dismiss
GradProj
Connect
View Monitoring
Browse Collections
FREE
SHARED
Visualize Your Data
Build dashboards and charts, and embed them in your apps with MongoDB Charts.
Dismiss
Explore Charts
R 0
W 0
Last 6 hours
Connections 0
Last 6 hours
In 0.0 B/s
Out 0.0 B/s
Last 6 hours
Data Size
990.2 KB / 512.0 MB (0%)
Last 30 days
VERSION
REGION
CLUSTER TIER
TYPE
BACKUPS
LINKED APP SERVICES
ATLAS SQL
ATLAS SEARCH


Database Deployments
https://cloud.mongodb.com/v2/655a83d985c14e73ff636d12#/clusters/connect?clusterId=GradProj
Connect to GradProj
Set up connection security
Choose a connection method
Connect
Connect to your application
Drivers
Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)
Access your data through tools
Compass
Explore, modify, and visualize your data with MongoDB's GUI
Shell
Quickly add & update data using MongoDB's Javascript command-line interface
MongoDB for VS Code
Work with your data in MongoDB directly from your VS Code environment


Database Deployments
https://cloud.mongodb.com/v2/655a83d985c14e73ff636d12#/clusters/connect?clusterId=GradProj
1. Select your driver and version
We recommend installing and using the latest driver version.
Driver
Node.js
Version
5.5 or later
2. Install your driver
Run the following on the command line
npm install mongodb
View MongoDB Node.js Driver installation instructions.
3. Add your connection string into your application code
View full code sample
mongodb+srv://batoolsh2001:<password>@gradproj.0hfoj2c.mongodb.net/?
retryWrites=true&w=majority
Replace <password> with the password for the batoolsh2001 user. Ensure any option params are URL encoded.
RESOURCES
Get started with the Node.js Driver
Access your Database Users
Node.js Starter Sample App
Troubleshoot Connections

```
const uri = 'mongodb+srv://batoolsh2001:<password>@gradproj.0hfoj2c.mongodb.net/?
retryWrites=true&w=majority';
```

3. Creating Database Connection:

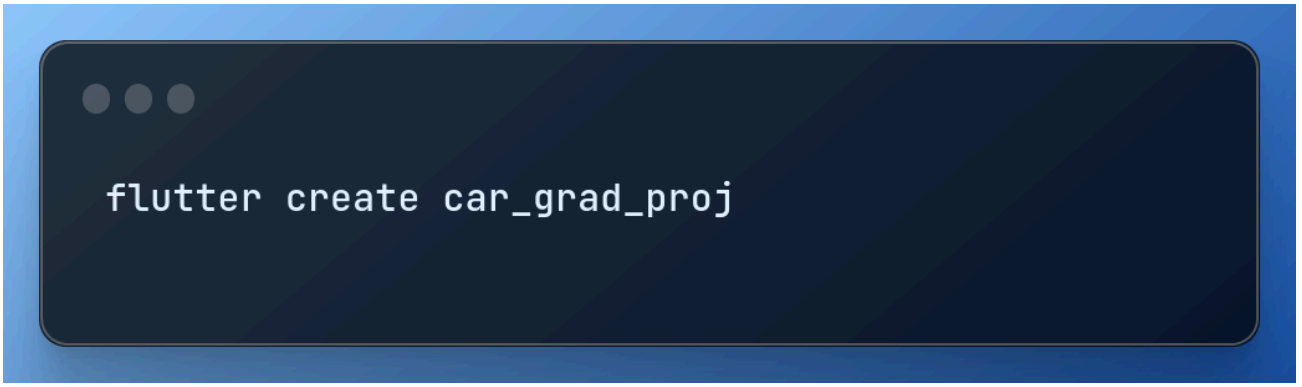
mongoose.createConnection` is used to establish a connection to the database using the provided URI and additional options. In this example, `useNewUrlParser` is used to handle updates to the MongoDB interface, and `useUnifiedTopology` is used to employ a single topology for connecting to the server.

4.5.3 Front-end Code Base Preparation

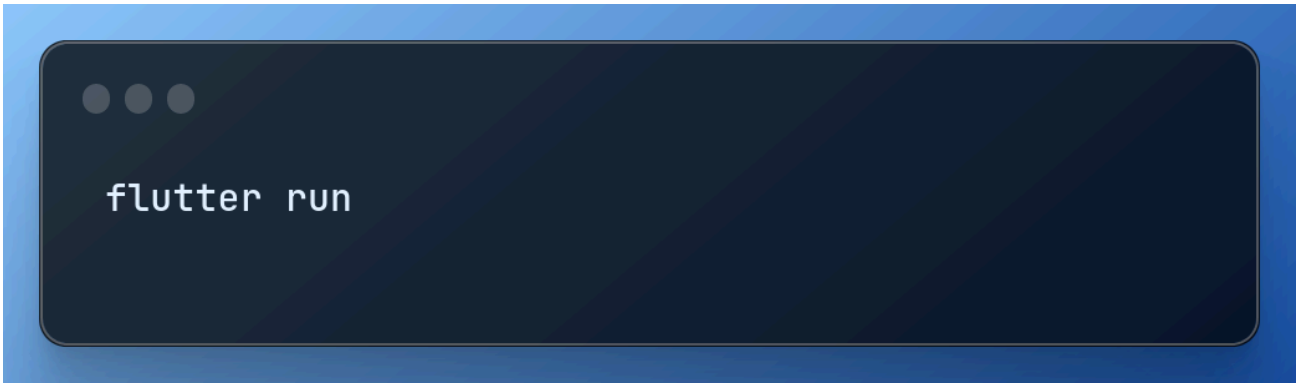
1. Install Flutter SDK:
   from Flutter website then Follow the instructions to download and install the Flutter SDK

2. Use VSCode:
   install the Flutter and Dart extensions for VSCode

3. Create a New Flutter Project:



3. Run the App for the First Time:

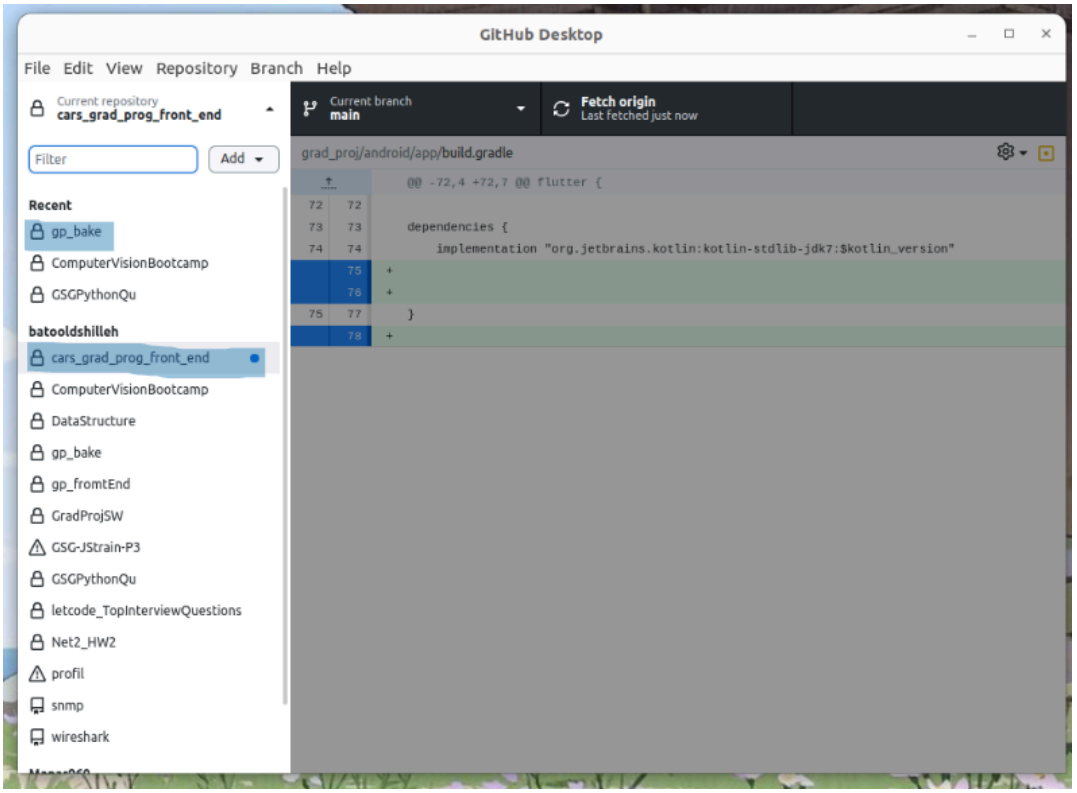


4.5.4 Git VCS and Github Hosting

The frontend and backend projects each used a different version control system, called "Git," to keep track of changes made to the project and preserve the history of code editing.

To go in the easiest and fastest direction, we decided to use the desktop version of GitHub.

As we advance each new piece of code, we commit each change once the repository has been created. Since there were two collaborators, the project's entire work was divided into two branches.
The opportunity to go back and examine and retrieve the previous codes is one advantage of this step.
Additionally, the "Github" service was used to host the project repositories online. The following actions were taken to accomplish this:
Establish a front-end repository and a back-end repository on the "Github" platform.
Connect every local repository to its matching remote repository.



Push should be used to synchronize newly created local commits with the remote repository.

This method has several advantages that could be utilized:

- The capacity to distribute the code to others.
- The capability to recover the code in the event that the code machine becomes inoperable or corrupted.

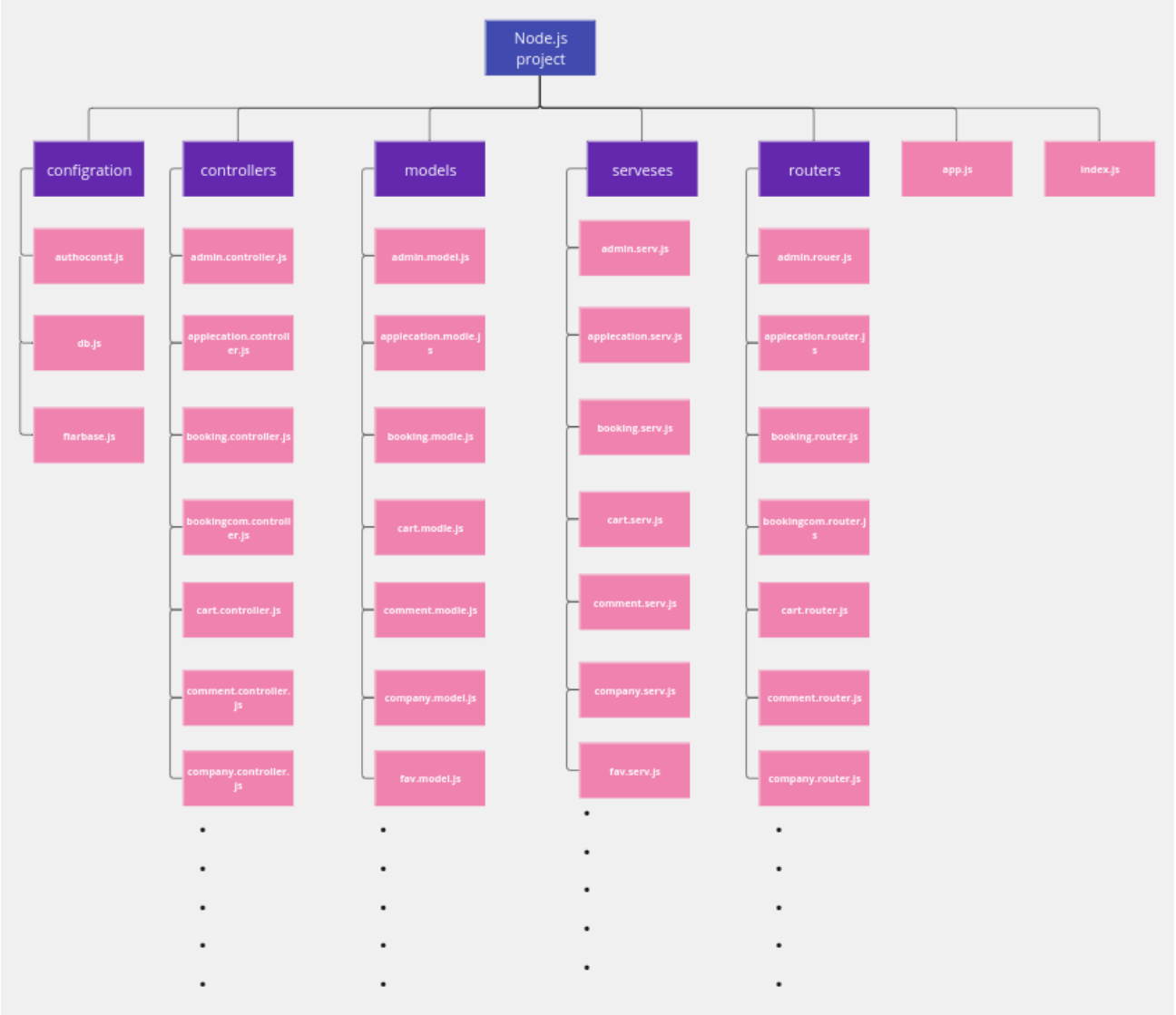
4.6 Backend Control Design and Implementation

This chapter contains comprehensive information on the implementation of server codes, handling of the payment transition, use of security techniques, and Postman backend testing.

4.6.1 Server Application Structure and Apps

We use MVC (Model-View-Controller) on the server-side, which includes the Model and Controller, while the View is on the front-end or client side. Utilizing this architecture leads to:

- Code Organization: Effectively helps organize the code by separating concerns related to presentation, model, and control.
- Ease of Maintenance:Separation between the model, presentation, and controller makes maintenance easier, allowing changes to each part without affecting the others.
- Code Reusability: Parts of the MVC pattern can be reused in other projects, especially when the separation between components is well-implemented.
- Scalability: Facilitates the development of scalable and extensible applications, allowing the addition of new components without impacting the existing parts.



And here’s a thorough look about each thing in the project folder, and explain how it used in this project:

4.7 System Administration

In this section, we will discuss the two key measures in the management process."

4.7.1 Server Management

We utilized Rendr as a primary tool for server management in our project. Rendr is a framework that can be used with Node.js to achieve high-performance and responsive applications. Additionally, the logging module in Rendr played a vital role in monitoring events and errors during the application's runtime. This logging feature allows us to record vital activities and review logs to identify any security issues or technical errors that may arise. The integration of Rendr with the logging module makes it easy for us to monitor and analyze server performance, a crucial part of our strategy to ensure the system operates efficiently and effectively

4.7.2 Django REPL

In our ongoing efforts to achieve continuous security and protection for our project, we have implemented a set of measures and practices. Firstly and foremost, we integrated the HTTPS protocol into our interaction with servers to secure data transmission between the application and servers, protecting information from manipulation or interception.

We also ensured regular updates of the packages and libraries used to get the latest security fixes and updates. This includes updates for Node.js, Flutter, and any other relevant packages.

We carefully implemented the principle of permission management, granting each component of the application only the necessary permissions to ensure secure and organized access. To provide an additional layer of protection, we implemented measures to protect against harmful data injection attacks by thoroughly checking input data.

On the other hand, we included a robust event logging system and continuous log monitoring. This helps us identify and evaluate unexpected events or unfamiliar activities, allowing us to react effectively in case of issues.

Ultimately, securing the entire project is an ongoing process, and we are committed to continuous improvement and regular inspection to ensure the continuity and efficiency of security measures.

4.8 Frontend Control Design and Implementation

To provide an optimal user experience and build an excellent front-end interface, we took the following steps

4.8.1 Designing the interface

Before building any interface, we used to watch several videos and explore various UX/UI designs. Then, we searched for widgets that met our needs. In the absence of design software, we drew rough sketches using paper and pen, engaging in discussions about them and how to build them

4.8.2 Components and control elements

Flutter contains an extensive set of widgets and libraries that provide excellent components, enhancing the user experience based on community reviews. It's not feasible to mention all the widgets here, but I'll highlight some of them and why they were used:

1. **SizedBox:** Used to create spaces between other widgets, either horizontally, vertically, or in both directions.
2. **Buttons:** Utilized in various parts of the project, with control over their color, press method, size, and font.
3. **App Bar:** We programmed our App Bar, relying on Flutter's built-in App Bar. Our App Bar was added to ensure it doesn't interfere with the safe area.
4. **Navigation Bar:** This part was manually programmed and designed.
5. **Container:** Acts as the main container that groups a set of widgets together, allowing for their usage and control.
6. **Rows and Columns:** Used to arrange elements either horizontally or vertically.

There are many other widgets and libraries used to improve the user experience and facilitate the development process.

4.8.2.1 User interaction

User interaction in our program occurs through various methods, each tailored to the interface and widget used. The main interaction methods include:

1. **Tap:** When the user clicks on a button or element in the interface.
2. **Input:** Involves user input of data, such as typing text in a text box or selecting a value from a dropdown menu.
3. **Navigation:** Interacting with navigation elements to move between pages or items within the application.
4. **Interaction with Notifications:** The user's response to notifications received from the application, either by clicking on them or taking a specific action.

All these methods contribute to providing a seamless and comfortable interactive experience for the user while using the application.

4.8.2.2 Responsiveness and Compatibility:

Flutter is a cross-platform language, so there's no need to worry about its compatibility with browsers and devices. However, attention must be given when configuring dependencies, libraries, or databases, even on the backend, to ensure compatibility with all platforms.

To control what should or should not be displayed on a particular platform, we used the `foundation.dart` package.

Regarding responsiveness, we built a custom class to ensure responsiveness across all screens excellently. We added calculations and ratios for screen dimensions, how to handle them, and how to calculate or control widget sizes and switch between rows and columns effectively.

4.8.2.3 Security:

1. Input Validation:

We implemented thorough validation of user inputs to prevent improper input attacks by verifying the accuracy of entered data.

2. Security Updates:

Ensured the use of the latest versions of packages and libraries to receive the latest security updates and guarantee the application's safety.

3. HTTPS Usage:

Utilized a secure connection (HTTPS) to secure the data transmitted between the application and servers.

4. Permission Management:

Organized permission models for users to control access and prevent unauthorized access.

5. Security Check in Firebase:

Verified the correct security configuration for databases and Cloud Functions in Firebase.

4.8.2.4 Performance:

1. Optimizing Image Size:

Improved the size of used images to reduce loading time and enhance the application's performance.

2. Lazy Loading:

Deferred the loading of non-essential data to speed up the initial user interface loading time.

4.8.2.5 Testing and Improvement

We used to add features and test them directly, presenting them to individuals around us to

gather their opinions and receive valuable feedback.

5. Results And Problems Discussion

5.1 Experimenting with the Android application using the emulator

The emulators require relatively powerful hardware, especially in terms of RAM, which should be 32 GB or higher for excellent performance. This issue made the development process slow, as previewing changes and running the emulator took a considerable amount of time. The solution was to use physical devices - mobile phones.

5.2 Dynamic data transfer between interfaces

At the beginning of the development phase, I used to manage state using GetX, and later I switched to using Provider, due to the advantages and good performance that Provider offers compared to GetX in our application's context.

1. Flexibility and Customization:
   - Provider offers a high level of flexibility and customization. If you need precise control over the state management and want to separate concerns within your application, Provider might be a good choice.

2. Integration with Specialized Containers:
   - If you plan to integrate with specialized containers for more advanced state management capabilities, Provider can easily be integrated for additional control over the scope of state updates.

3. Fine-Grained Control over Component State:
   - Provider allows for fine-grained control over the state of individual components. You can specify which updates to the state should trigger a rebuild within a component, providing more granular control.

4. Unidirectional Data Flow:
   - Provider follows a unidirectional data flow pattern, making it suitable for projects where a clear and predictable flow of data is a priority.

5.3 Payment using a credit card

We had to wait for approval to use these tools and APIs, and some of these services were paid, which led us to use a ready-made API to simulate the electronic payment process

5.4 Tracking on the map

To track more than one request on the map, we needed to register and wait for approval, which involved both time and fees. As a time-saving measure, we presented the user's requests with descriptions on the map instead

5.5 Using the web with Firebase

Firebase does not directly and comprehensively support all web services, as it can lead to conflicts and inconsistencies. This compelled us to use the Command Line Interface (CLI) to facilitate the efficient handling of web-related tasks

5.6 Library conflicts

After the release of Dart 2.12, the term 'null safety' emerged, where libraries must support this feature to function efficiently and successfully in the application. Some libraries did not initially support this feature, forcing us to replace them. However, downgrading the version was not a viable option due to security and functionality concerns.Moreover, when certain libraries encountered others, it led to conflicts, prompting us to change some versions to resolve compatibility issue

5.7 Gradle file issues

This file is responsible for the build process of Android applications and how they operate. It has been a source of many issues due to continuous updates, often conflicting with library versions, SDKs, Java, and other components.

5.8 Compatibility with all screens

Yes, initially, it was a significant challenge to create a responsive application that would be compatible with all screens. Libraries often struggled to work precisely in this regard. Therefore, we built a class to handle this process.

5.9 Running the server on a non-local machine

Initially, testing the application on a separate Android device was challenging, involving changing the IP to the network address. If the work network changed or the device was not connected to the same network, testing became impossible. However, the solution was simple; we searched for free websites that provide hosting services for our server, and yes, we succeeded in doing that using Render

5.10 Spaghetti code

Due to the large size of the project and the multitude of functions, we had some classes written in a spaghetti code approach. Refactoring and restructuring these classes to make them more organized and concise was a laborious process. Continuing with this approach made adding, removing, or modifying anything a very challenging task.

6. Conclusion

6.1 Summary

The experience was unique and filled with learning and trying new things. It was a fertile opportunity to enhance our knowledge. We learned cross-platform programming, the process of adding dependencies for different platforms, dealing with Google services, Firebase, and document-oriented databases. It also provided a chance to experience mobile app development. Our skills in Node.js evolved, and we dealt with various libraries and features to achieve optimal results. Attempting to write code professionally following a specific standard as much as possible to ensure ease of updating and fixing was also part of the learning process.

6.2 Improvements

Certainly, there are many improvements that should be added to the project. In the end, we are university students learning and not highly professional in creating a comprehensive project. Some of these improvements include:

1. Performance enhancements to address occasional slow response times.
2. Updating the hosting service, as the current service used is free and may not be suitable for optimal use.
3. Code refinement and making it clean and adhering to coding standards (we attempted to do this as much as possible).
4. Enhancing the user interface to make it more user-friendly.

This is not an exhaustive list, but these are some of the key points.

6.3 Future Work

These features were supposed to be added to the project, but due to time constraints, we were unable to do so:

1. Adding a theoretical testing section.
2. Adding a section that allows users to customize the cat and accessories on a virtual car.
3. Artificial Intelligence intelligence question service and obtaining answers from it.
4. Paint pages service and using a color palette to obtain distinctive colors.

Bibliography

Resources for the online simple market research in section 4.2:

1 - The Ultimate List of Marketing Statistics for 2022 (Link: https://www.hubspot.com/marketing-statistics)

Resources utilized while implementation:

Appendix A

Attachment A

Proposed Disclaimer Statement Format The report is a document written by the student(s). It should reflect expertise in research methodology and technical writing skills. The supervisor's job is to guide the student so that she/he can achieve the objectives in an efficient way while gaining the skills sought. While maintaining credit the disclaimer statement is simply a statement protecting the Department and the University from any legal liability claims associated with the use of the results and the methods presented. Its format is as follows: DISCLAIMER This report was written by Batool Shilleh and Manar Jaber at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of the assessment and it may contain language as well as content errors.