Assignment 2: Documentation

Iris Nguyen Student ID: 34065016

Brief:

Due to the many drivers and packages, Monash started to consider the need to store its data persistently. Two more issues have emerged: The Monash IT department cannot host the application locally on-premise. Instead, it prefers to deploy it to a cloud provider that charges per usage. Secondly, Monash mobile (Android and IOS) and desktop applications have requested access to the App data.

Routing table:

Name	HTTP Method	Path Source	Path
Home Page		/	Render the homepage
Add Driver Form	GET	/34065016/Iris/add-driver	Render the form to add a driver
Add Driver	POST	/34065016/Iris/add-driver	Handle adding a new driver to the store
List Drivers	POST	/34065016/Iris/drivers	List all drivers
Delete Driver Form	GET	/34065016/Iris/delete-driver	Render the form to delete a driver
Delete Driver	GET	/34065016/Iris/delete-driver-submit	Handle the deletion of a driver
Add Package Form	GET	/34065016/Iris/add-package	Render the form to add a package
Add Packages	POST	/34065016/Iris/add-package	Handle adding a new package
List Packages	GET	/34065016/Iris/packages	List all packages
Delete Package Form	GET		Render the form to delete a package

		/34065016/Iris/delete-package	
Delete Package	GET	/34065016/Iris/delete-package-submit	Handle the deletion of a package
Department Form	GET	/34065016/Iris/driver-department	Render form to select department
List Drivers By Department	POST	/34065016/Iris/driver-department	List drivers by selected department
Invalid Data	GET	/34065016/Iris/invalid-data	Render invalid data page
404 Not Found	GET	*	Handle 404 errors
Login Page	GET	/login	Handle Login Page
Sign up Page	GET	/signup	Handle Sign up Page
Log out Page	GET	/logout	Handle Logout page
Post Login information	POST	/loginSubmit	Post the login information
CRUD Operations	GET	/34065016/stats	Handle the calculation of CRUD Operations
Post Sign Up Information	POST	/signupSubmit	Post the sign up information
API			
Adding Package	POST	/34065016/api/v1/packages/add	Insert new Package
Listing Package	GET	/34065016/api/v1/packages	List all packages
Deleting Package	DELETE	/34065016/api/v1/packages/removes	Delete package through ID
Update Package	PUT	/34065016/api/v1/packages/:id	Updating package
Adding Driver	POST	/34065016/api/v1/drivers/add	Insert new Driver
Listing Driver	GET	/34065016/api/v1/drivers	List all drivers

Deleting Driver	DELETE	/34065016/api/v1/drivers/removes	Delete Driver
Update driver	PUT	/34065016/api/v1/drivers/:id	Update Driver
Adding Package to driver	PUT	/34065016/api/v1/add-package/:id	Adding package to driver
Sign Up	POST	/34065016/api/v1/packages/signup	Sign Up
Log In	GET	/34065016/api/v1/packages/login	Log In
Log Out	GET	/34065016/api/v1/packages/logout	Log Out

2. Step to run code

1. Initialize the project: npm init -y
2. Install dependencies: npm install express mongoose bcrypt firebase-admin express-session body-parser
3. Ensure Firebase setup: Download service-account.json from Firebase and place it in the project directory.
4. Setup MongoDB: Ensure MongoDB is running locally or via a cloud provider.
5. Start the server: Run node app.js to start the application.
6. Sign up a new user: Visit /signup to create a new account.
7. Log in: Go to /login to log in with the registered account.
8. Access routes: After logging in, access protected routes such as /drivers, /packages, based on the routine table