## horizontal line

# **Backend Services**

## Admin registration

This service takes the user’s email id, first name, last name for creating their account and allows the user to set their password. When the user is successfully registered, it returns a confirmation message “You are registered and can now login”.

In case a user is already registered or the service is unable to complete the registration then appropriate responses are sent to the user.

## Admin authentication

### **Sign in service for admin:**

This service first authenticates the admin. After a successful authentication, a non-expiring JWT token is generated which encodes the email id and the IsAdmin field, which by default is set to **False** in this case it will be **True**. This token is returned and admin dashboard view is loaded

### **Sign out service:**

The service updates the last active date in the user profile, and sends out a JWT token as a success response.

## Admin - Change password

This service validates the admin’s current password and allows admin to change their password after the validation is successful the new password is updated in the admin table. The admin is logged out and **“successfully changed password, please login with new password”** response is sent.

## Admin - Forgot password

This service sends a password reset link to the provided email id, provided the email id is found in the database and once admin provides a new password, through the link provided, the same is updated in the admin table and **“successfully changed password, please login with new password”** response is sent.

## Admin-States

### **Listing:**

This service shows the list of states currently available in the database.

### **Add State:**

This service allows the admin to add a new state providing state name and state code(internal) and the same will be added in the states table. Once done, **“successfully added {state name}”** response is sent.

### **Modify State:**

This service allows the admin to modify the details(name and code) of an already existing state and the same will be updated in the states table. Once done, **“successfully modified {state name} details”** response is sent.

### **Delete State:**

This service allows the admin to completely delete(or we can change status to inactive with updated date time) an already existing state and the same will be updated in the states table. Once done, **“successfully deleted {state name}”** response is sent.

Once the state functionalities are confirmed functionalities for districts, constituencies, etc., will be updated here.

## User registration

This service takes the user’s email id, first name, last name for creating their account and allows the user to set their password. When the user is successfully registered, it returns a confirmation message “You are registered and can now login”.

In case a user is already registered or the service is unable to complete the registration then appropriate responses are sent to the user.

## Admin authentication

### **Sign in service for admin:**

This service first authenticates the admin. After a successful authentication, a non-expiring JWT token is generated which encodes the email id and the IsAdmin field, which by default is set to **False** in this case it will be **True**. This token is returned and admin dashboard view is loaded

### **Sign out service:**

The service updates the last active date in the user profile, and sends out a JWT token as a success response.

## Admin - Change password

This service validates the admin’s current password and allows admin to change their password after the validation is successful the new password is updated in the admin table. The admin is logged out and **“successfully changed password, please login with new password”** response is sent.

## Admin - Forgot password

This service sends a password reset link to the provided email id, provided the email id is found in the database and once admin provides a new password, through the link provided, the same is updated in the admin table and **“successfully changed password, please login with new password”** response is sent.

## User history manager

This service allows the user to view all the actions performed by a them such as edit, create, rename and delete along with the associated timestamp.

## RazorPay integration

Razorpay has been chosen as the payment gateway for subscribers, since it supports multiple payment methods. We have created a Razorpay account with the following gmail id: Techdocs.gl@gmail.com, and have used the gateway in Test Mode only.

The service accepts the user’s preferred payment method which can be using a Card or via UPI. In case of card payments, it takes the card details, the CVV and validates the expiry date entered. To proceed with payment the user needs to click on “Pay”. For UPI payments, the service accepts a UPI ID.

## Future scope