PAYSLIP TECHNICAL DOCUMENT

**Roles:-**

|  |  |
| --- | --- |
| **Role** | **Activities** |
| Admin | Admin can access Employee,user screens |
| USER | User can access only user screen |
| EMPLOYEE | Employee can access only employee screen |

**Employee Login :-**

Method : Post

URL: [http://192.168.1.163:8092/employee/send-otp](http://192.168.1.163:8092/admin/send-otp)

|  |
| --- |
| {  “emailId” : “email@gmail.com”,  “password” :“ password”,  } |

**Logic:**

* After giving a valid emailId it will send OTP to the entered email
* If username is invalid it will throw an error invalid username

Method :Post

|  |
| --- |
| {  “otp” : “ 123456”  } |

URL : [http://192.168.1.163:8092/employee/](http://192.168.1.163:8092/admin/send-otp)login

**Logic** :

* If OTP is invalid throws an error as OTP is invalid
* OTP is valid redirect to dashboard based on the role in response.

**User Login :**

Method : Post

URL: [http://192.168.1.163:8092/user/send-otp](http://192.168.1.163:8092/admin/send-otp)

|  |
| --- |
| {  “emailId”/”userName” : “email@gmail.com”/ ”userName”,  “password” :“ password”,  } |

**Logic:**

* After giving a valid email Id or password it will send OTP to the respective email Id
* If username,email Id and password is invalid it will throw an error invalid username or password or email Id

Method :Post

|  |
| --- |
| {  “otp” : “ 123456”  } |

URL : [http://192.168.1.163:8092/user/](http://192.168.1.163:8092/admin/send-otp)login

**Logic** :

* If OTP is invalid throws an error as OTP is invalid
* OTP is valid redirect to dashboard based on the role in response.

**EMPLOYEE:- registration**

Method : POST

[URL:-](URL:/admin/add) <http://192.168.1.163:8092/employee/registration>

|  |
| --- |
| {  "employeeType": "Permanent",  "employeeId": "EMP001",  "firstName": "keerthi",  "lastName": "sree",  "emailId": "keerthisreev@pathbreakertech.in",  "password": "keerthi",  "designation": " Developer",  "dateOfHiring": "2022-01-25",  "department": "Backend",  "location": "Nasuja",  "manager": "vijay",  "role": "EMPLOYEE",  "status": 1,  "ipAddress": "122.168.1.163"  } |

**Logic:-**

* Employee registration will be done by using the above data given in the form.
* The similar fields which are present in the employee login table will also store. Shown in the below payload

|  |
| --- |
| { "emailId": "keerthisreev@pathbreakertech.in",  "password": "keerthi",  "role": "EMPLOYEE",  "status": 1,  "ipAddress": "122.168.1.163"  "employeeId": "EMP001"  } |

**Employee\_actions:**

Method : GET

URL:[http://192.168.1.163:8092/employee/a](http://192.168.1.163:8092/employee/registration)ll

|  |
| --- |
| [  { "employeeType": "Permanent",  "employeeId": "EMP001",  "firstName": "keerthi",  "lastName": "sree",  "emailId": "keerthisreev@pathbreakertech.in",  "password": "keerthi",  "designation": " Developer",  "dateOfHiring": "2022-01-25T00:00:00.000+00:00",  "department": "Backend",  "location": "Nasuja",  "manager": "vijay",  "role": "EMPLOYEE",  "status": 1,  "ipAddress": "122.168.1.163",  "loginResponse": {  "id": 3,  "otp": 0,  "emailId": "keerthisreev@pathbreakertech.in",  "password": "keerthi",  "status": 1,  "lastLoginTime": null,  "role": "EMPLOYEE",  "ipAddress": "122.168.1.163"  }  }  ] |

**Logic:**

* All the Employee details will populate by using the above URL.
* If it doesn’t get any details in the database it shows an empty.

Method **:**GET BY ID

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/employee/](http://192.168.1.163:8092/employee/registration){employeeId}

Request payload :

|  |
| --- |
| {  "employeeType": "Permanent",  "employeeId": "EMP001",  "firstName": "keerthi",  "lastName": "sree",  "emailId": "keerthisreev@pathbreakertech.in",  "password": "keerthi",  "designation": " Developer",  "dateOfHiring": "2022-01-25T00:00:00.000+00:00",  "department": "Backend",  "location": "Nasuja",  "manager": "vijay",  "role": "EMPLOYEE",  "status": 1,  "ipAddress": "122.168.1.163",  "loginResponse": {  "id": 3,  "otp": 0,  "emailId": "keerthisreev@pathbreakertech.in",  "password": "keerthi",  "status": 1,  "lastLoginTime": null,  "role": "EMPLOYEE",  "ipAddress": "122.168.1.163"  }  } |

**Logic:-**

* Based on the employee id it will check in the database,if found it will get all the details of the particular employee
* If the regarding id not found and it will show “Id not found” error

Method **:** - PUT

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/employee/](http://192.168.1.163:8092/employee/registration){employeeId}

Request payload :

|  |
| --- |
| {  "employeeType": "Permanent",  "firstName": "keerthi",  "lastName": "sree",  "emailId": "keerthisreev@pathbreakertech.in",  "password": "keerthi",  "designation": " Developer",  "dateOfHiring": "2022-01-25T00:00:00.000+00:00",  "department": "Backend",  "location": "Nasuja",  "manager": "vijay",  "role": "EMPLOYEE",  "status": 1,  "ipAddress": "122.168.1.163",  } |

**Logic:-**

* Based on the employee id it will check in the database,if found it will get all the Update the details of the particular employee.
* The update will be done in the employee login table since we are updating the

Employee details.

* If the regarding id not found and it will show “Id not found” error

Method : DELETE

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/employee/](http://192.168.1.163:8092/employee/registration){employeeId}

**Logic:**

Based on the Employee id it will check the database if it is existing or not.

If id is exists then it will deleted.

If it is not existed then it show “id not found ” error

**User :- registration**

Method : POST

[URL:-](URL:/admin/add) [http://192.168.1.163:8092/user/registration](http://192.168.1.163:8092/employee/registration)

|  |
| --- |
| {  "userId": "US005",  "emailId": "narendrab@pathbreakertech.in",  "userName": "narendra",  "password": "narendra",  "role": "ADMIN",  "registrationDate": "2024-01-30"  } |

**Logic:-**

* User registration will be done by using the above data given in the form.
* The similar fields which are present in the userlogin table will also store. Shown in the below payload

|  |
| --- |
| {  "emailId": "narendrab@pathbreakertech.in",  "userName": "narendra",  "password": "narendra",  "role": "ADMIN",  "registrationDate": "2024-01-30"  "userId": "US005",  } |

**User\_actions:**

Method : GET

URL: - [http://192.168.1.163:8092/user/a](http://192.168.1.163:8092/employee/registration)ll

|  |
| --- |
| [  {  "userId": "US001",  "emailId": "anushak@pathbreakertech.in",  "userName": "Anusha",  "password": "Kavali",  "role": "USER",  "registrationDate": "2024-01-28T00:00:00.000+00:00",  "userLoginResponse": {  "id": 1,  "emailId": "anushak@pathbreakertech.in",  "userName": "Anusha",  "password": "Kavali",  "role": "USER",  "lastLoginTime": "2024-01-29T08:27:38.000+00:00"  }  },  {  "userId": "US002",  "emailId": "keerthisreev@pathbreakertech.in",  "userName": "keerthi",  "password": "keerthi",  "role": "ADMIN",  "registrationDate": "2024-01-29T00:00:00.000+00:00",  "userLoginResponse": {  "id": 3,  "emailId": "keerthisreev@pathbreakertech.in",  "userName": "keerthi",  "password": "keerthi",  "role": "ADMIN",  "lastLoginTime": null  }  }  ] |

**Logic:-**

* All the user details will populate by using the above URL.
* If it doesn’t get any details in the database it shows an empty.
* All the userLogin details will also get in the console.

Method **:**GET BY ID

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/user/](http://192.168.1.163:8092/employee/registration){userId}

|  |
| --- |
| {  "userId": "US001",  "emailId": "anushak@pathbreakertech.in",  "userName": "Anusha",  "password": "Kavali",  "role": "USER",  "registrationDate": "2024-01-28T00:00:00.000+00:00",  "userLoginResponse": {  "id": 1,  "emailId": "anushak@pathbreakertech.in",  "userName": "Anusha",  "password": "Kavali",  "role": "USER",  "lastLoginTime": "2024-01-29T08:27:38.000+00:00"  } |

**Logic:-**

* Based on the userid it will check in the database,if found it will get all the details of the particular user
* If the regarding id not found and it will show “Id not found” error

Method **:** - PUT

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/user/](http://192.168.1.163:8092/employee/registration){userId}

Request payload :

|  |
| --- |
| {  "emailId": "anushak@pathbreakertech.in",  "userName": "Anusha",  "password": "Kavali",  "role": "USER",  } |

**Logic:-**

* Based on the userid it will check in the database,if found it will Update the details of the particular user.
* The update will be done in the userlogin table since we are updating the

User details.

* If the regarding id not found and it will show “Id not found” error.

Method : DELETE

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/user/](http://192.168.1.163:8092/employee/registration){userId}

**Logic:**

Based on the id it will check the database if it is existing or not.

If id is exists then it will deleted.

If it is not existed then it show “id not found ” error

**Relieving :- adding the employee to relieving screen**

Method : POST

[URL:-](URL:/admin/add) [http://192.168.1.163:809](http://192.168.1.163:8092/employee/registration)[2/relieving/add](http://192.168.1.163:8092/role/add)

|  |
| --- |
| { "employeeId": "EMP003",  "designation": "Java Developer",  "typeOfEmployement": "Full-time",  "resignationDate": "2024-02-01",  "lastWorkingDate": "2024-02-20" } |

**Logic:-**

* Relieving registration will be done by using the above data given in the form.

**Relieving\_actions:**

Method : GET

URL: - [http://192.168.1.163:8092/relieving/a](http://192.168.1.163:8092/employee/registration)ll

|  |
| --- |
| [  {  "id": 2,  "employeeId": "EMP003",  "designation": "Java Developer",  “firstName” : ”keerthi”,  “lastName” : “sree”,  "typeOfEmployement": "Full-time",  "resignationDate": "2024-02-01",  "lastWorkingDate": "2024-02-20"  },  { "id": 3,  "employeeId": "EMP004",  "designation": “ Developer",  "designation": "Java Developer",  “firstName” : ”keerthi”,  “lastName” : “sree”,  "typeOfEmployement": "contract",  "resignationDate": "2024-08-07",  "lastWorkingDate": "2024--12-09"  }  ] |

**Logic:-**

* All the relieving details will populate by using the above URL and also employee first name and last name will be populated.
* If it doesn’t get any details in the database it shows an empty.

Method **:-** GET BY ID

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/relieving/](http://192.168.1.163:8092/employee/registration){employeeId}

|  |
| --- |
| { "id": 3,  "employeeId": "EMP004",  "designation": “ Developer",  "designation": "Java Developer",  “firstName” : ”keerthi”,  “lastName” : “sree”,  "typeOfEmployement": "contract",  "resignationDate": "2024-08-07",  "lastWorkingDate": "2024--12-09"  } |

**Logic:-**

* Based on the id it will check in the database,if found it will get all the details of the particular relieving employee details and also employee first name and last name will be populated.
* If the regarding id not found and it will show “Id not found” error

Method **:** - PUT

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/relieving/](http://192.168.1.163:8092/employee/registration){employeeId}

Request payload :

|  |
| --- |
| {  "designation": “ Developer",  "typeOfEmployement": "contract",  "resignationDate": "2024-08-07",  "lastWorkingDate": "2024--12-09"  } |

**Logic:-**

* Based on the employeeId it will check in the database,if found it will Update the details of the particular user.
* If the regarding id not found and it will show “employeeId not found” error.

Method : DELETE

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/relieving/](http://192.168.1.163:8092/employee/registration){employeeId}

**Logic:**

Based on the employeeId it will check the database if it is existing or not.

If employeeId is exists then it will deleted.

If it is not existed then it show “employeeId not found ” error

**PaySlips:- add the payslips of the employees**

Method : POST

[URL:-](URL:/admin/add) [http://192.168.1.163:809](http://192.168.1.163:8092/employee/registration)2/[payslip/upload-document](http://192.168.1.163:8092/payslip/upload-document)

|  |
| --- |
| {  "employeeId" : "EMP001"  "month" : "january"  "financialYear" : 2024  file: select file Content-Type : multipart/form-data  } |

**Logic:-**

* Payslip upload will be done by using the above data given in the form.

And we have to select the file of the payslip it will store in the server.

**Relieving\_actions:**

Method : GET

URL: - [http://192.168.1.163:8092/payslip/a](http://192.168.1.163:8092/employee/registration)ll

|  |
| --- |
| [{  "id": 1,  "employeeId": "EMP001",  “firstName” : ”keerthi”,  “lastName” : “sree”,  "month": "January",  "financialYear": 2024,  "filePaths": "D:\\PaySlip\_WS\\Documents\\Screenshot (8).pngEMP001"  }, {  "id": 2,  "employeeId": "EMP001",  “firstName” : ”keerthi”,  “lastName” : “sree”,  "month": "January",  "financialYear": 2024,  "filePaths": "D:\\PaySlip\_WS\\Documents\\Screenshot (3).pngEMP009"  }] |

**Logic:-**

* All the payslip details will populate by using the above URL and also employee first name and last name will be populated.
* If it doesn’t get any details in the database it shows an empty.

Method **:-** GET BY employeeID

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/payslip/](http://192.168.1.163:8092/employee/registration){employeeId}

|  |
| --- |
| [ {  "id": 8,  "employeeId": "EMP001",  “firstName” : ”keerthi”,  “lastName” : “sree”,  "month": "January",  "financialYear": 2024,  "filePaths": "D:\\PaySlip\_WS\\Documents\\d16cb8ce-21cf-4cbd-bde1-fbd239240554\_Screenshot (3).png"  },  {  "id": 13,  "employeeId": "EMP001",  “firstName” : ”keerthi”,  “lastName” : “sree”,  "month": "febuary",  "financialYear": 2024,  "filePaths": "D:/PaySlip\_WS/Payslip\_Documents/EMP001\_Desktop - 4.png"  }] |

**Logic:-**

* Based on the emplyeeId it will check in the database,if found it will get all the details of the particular employee payslips and also employee first name and last name will be populated.
* If the regarding employeeId not found and it will show “employeeId not found” error

Method **:** - PUT

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/relieving/](http://192.168.1.163:8092/employee/registration){employeeId}

Request payload :

|  |
| --- |
| {  "designation": “ Developer",  "typeOfEmployement": "contract",  "resignationDate": "2024-08-07",  "lastWorkingDate": "2024--12-09"  } |

**Logic:-**

* Based on the employeeId it will check in the database,if found it will Update the details of the particular user.based on employeeId
* If the regarding employeeId not found and it will show “employeeId not found” error.

Method : DELETE

[URL:](URL:/admin/%7badminId%7d) - [http://192.168.1.163:8092/relieving/](http://192.168.1.163:8092/employee/registration){employeeId}

**Logic:**

Based on the employeeId it will check the database if it is existing or not.

If employeeId is exists then it will deleted.

If it is not existed then it show “employeeId not found ” error

**Role:**

Method:POST

URL: <http://192.168.1.163:8092/role/add>

Request payload:

|  |
| --- |
| {“role”:”HR”} |

**Logic:**

* Role will added by using the above URL.

Method:GET

URL:<http://192.168.1.163:8092/role/all>

Request payload:

|  |
| --- |
| {  “Id”:1,  “role”:”User”  },  {  “id” : 2,  “role” : “Admin”  } |

**Logic:**

* All the role details will populate by using the above URL.
* If it doesn’t get any details in the database it shows an empty.

Method **:-** GET BY ID

[URL:](URL:/admin/%7badminId%7d) - <http://192.168.1.163:8092/role/>{id}

|  |
| --- |
| {  “Id”:1,  “role”:”User”  } |

**Logic:-**

* Based on the roleId it will check in the database,if found it will get all the details of the particular Id .
* If the regarding Id not found and it will show “Id not found” error

Method **:**PUT

URL: <http://192.168.1.163:8092/role/>{id}

Request payload:

|  |
| --- |
| {  “role”:”User”  } |

**Logic:**

* Based on the id it will check in the database,if it find it will update the field.
* If the regarding id not found and it will show “Id not found”

Method : DELETE

URL:  [http://192.168.1.163:8092/role/{id}](http://192.168.1.163:8092/role/%7Bid%7D)

**Logic:**

* Based on the id it will check the database if it is existing or not.
* If id is exists then it will deleted.
* If it is not existed then it show “id not found ”

**Department:**

Method:POST

URL:http://192.168.1.163:8092/department/add

Request payload:

|  |
| --- |
| {  "departmentTitle": "Development"  } |

**Logic:**

* Department details will added by the end user.

Method:GET

URL: http://192.168.1.163:8092/department/all

Request payload:

|  |  |
| --- | --- |
| {  “Id”:1,  "departmentTitle": "Development"  },  {  “Id”:2,  "departmentTitle": "Testing"  } | |
|  | |

**Logic:**

* All the department details will populate by using the above URL.
* If it doesn’t get any details in the database it shows an empty.

Method **:-** GET BY ID

[URL:](URL:/admin/%7badminId%7d) - http://192.168.1.163:8092/department/{id}

|  |
| --- |
| {  “Id”:1,  "departmentTitle": "Development"  } |

**Logic:-**

* Based on the Id it will check in the database,if found it will get all the details of the particular Id .
* If the regarding Id not found and it will show “Id not found” error

Method **:**PUT

URL: http://192.168.1.163:8092/department/{id}

Request payload:

|  |
| --- |
| {  "departmentTitle": "Development"  } |

**Logic:**

* Based on the id it will check in the database,if it find it will update the field.
* If the regarding id not found and it will show “Id not found”

Method : DELETE

URL:http://192.168.1.163:8092/department/{id}

**Logic:**

* Based on the id it will check the database if it is existing or not.
* If id is exists then it will deleted.
* If it is not existed then it show “id not foun

**Designation:**

Method:POST

URL: http://192.168.1.163:8092/designation/add

Request payload:

|  |
| --- |
| {  "designationTitle": "Associate Engineer"  } |

**Logic:**

* Designation title will be added by the end user.

Method:GET

URL: http://192.168.1.163:8092/designation/all

Request payload:

|  |
| --- |
| { “Id”:1,  "designationTitle": "Associate Engineer"  },  {  “Id”:2,  "designationTitle": "Trainee Engineer" } |

**Logic:**

* All the designation titles will populate by using the above URL.
* If it doesn’t get any details in the database it shows an empty.

Method **:-** GET BY ID

[URL:](URL:/admin/%7badminId%7d) http://192.168.1.163:8092/designation/{id}

|  |
| --- |
| {  “Id”:1,  "designationTitle": "Associate Engineer"  } |

**Logic:-**

* Based on the Id it will check in the database,if found it will get all the details of the particular Id .
* If the regarding Id not found and it will show “Id not found” error

Method **:**PUT

URL: http://192.168.1.163:8092/designation/{id}

Request payload:

|  |
| --- |
| {  "designationTitle": "Associate Engineer"  } |

**Logic:**

* Based on the id it will check in the database,if id found it will update the field.
* If the regarding id not found and it will show “Id not found”

Method : DELETE

URL:http://192.168.1.163:8092/designation/{id}

**Logic:**

* Based on the id it will check the database if it is existing or not.
* If id is exists then it will deleted.
* If it is not existed then it show “id not found ”

**Status:**

Method:POST

URL: http://192.168.1.163:8092/status/add

Request payload:

|  |
| --- |
| {  "statusInfo": "Active"  } |

**Logic:**

* Status details will added by the end user.

Method:GET

URL: http://192.168.1.163:8092/status/all

Request payload:

|  |
| --- |
| {  "status": 1,  "statusInfo": "Active"  },  {  ""status"": 2,  "statusInfo": "Inactive"  } |

**Logic:**

* All the Status details will populate by using the above URL.
* If it doesn’t get any details in the database it shows an empty.

Method **:-** GET BY ID

[URL:](URL:/admin/%7badminId%7d) http://192.168.1.163:8092/status/{id}

|  |
| --- |
| {  "status": 1,  "statusInfo": "Active"  } |

**Logic:-**

* Based on the Id it will check in the database,if found it will get all the details of the particular Id .
* If the regarding Id not found and it will show “Id not found” error

Method **:**PUT

URL: http://192.168.1.163:8092/status/{id}

Request payload:

|  |
| --- |
| {  "statusInfo": "Active"  } |

**Logic:**

* Based on the details will check in the database,if it find it will update the field.
* If the regarding id not found and it will show “Id not found”

Method : DELETE

URL: http://192.168.1.163:8092/status/{id}

**Logic:**

* Based on the id it will check the database if it is existing or not.
* If id is exists then it will deleted.
* If it is not existed then it show “id not found ”

**Dashboard Api’s:**

Method:GET

URL: <http://192.168.1.163:8092/dashboard/employees>

Request payload:

|  |
| --- |
| [9] |

**Logic:**

* All the employees count will be populated by using the above URL.
* If it doesn’t get any employees in the database it shows an empty.

Method **:**GET

URL: http://192.168.1.163:8092/dashboard/relieved

Request payload:

|  |
| --- |
| [2] |

**Logic:**

* All the relieving employees count will be populated by using the above URL.
* If it doesn’t get any relieving employees in the database it shows an empty.

Method : GET

URL:http://192.168.1.163:8092/dashboard/all/employees

Request payload:

|  |
| --- |
| 9+2=[11] |

**Logic:**

* All the relieving employees & actual employees will be added & count will be populated by using the above URL.
* If it doesn’t get any relieving employees & actual employees in the database it shows an empty.