

users.schemas.js

```
import { number, object, string } from 'yup'
import GlobalConst from '../constants/globals.constants.js'

const { nameRegx, emailRegx, passwordRegx } = GlobalConst.regEx

const UsersSchema = {}

UsersSchema.create = object()
  .shape({
    name: string()
      .matches(nameRegx, 'Invalid Name Provided ')
      .required('Name Is a Required'),
    email: string()
      .matches(emailRegx, 'Invalid Mail Address')
      .required('Email Is Required'),
    password: string()
      .matches(passwordRegx, 'Inavlid Password')
      .required('Password Is Required'),
    gender: string()
      .oneOf(['male', 'female'], 'invalid gender')
      .required('Gender Is Required'),
    bloodGroup: string()
      .oneOf(
        ['a+', 'a-', 'b+', 'b-', 'ab+', 'ab', 'ab-', 'o+', 'o-'],
        'Invalid Blood Group'
      )
      .required('Blood Group Is Required'),
    age: number()
      .min(1, 'Age must be above one')
      .max(80, 'Age Must not be above 80')
      .required('Age Is Required'),
    city: string(),
```

```

        postcode: number()
        .min(1000, 'Postcode must be in Four Digit Number')
        .max(9999, 'Postcode must not be Above Four Digit Number')
        .required('PostCode Is Required')
    })
    .strict()
    .noUnknown()

```

```

UsersSchema.update = object()
    .shape({
        name: string().optional().matches(nameRegx, 'Invalid Name Pl
        password: string().optional().matches(passwordRegx, 'Inavli
        gender: string().oneOf(['male', 'female'], 'invalid gender'
        bloodGroup: string().oneOf(
            ['a+', 'a-', 'b+', 'b-', 'ab+', 'ab', 'ab-', 'o+', 'o-']
            'Invalid Blood Group'),
        age: number().min(1, 'Age must be above one').max(80, 'Age I
        city: string(),
        postcode: number().min(1000, 'Postcode must be in Four Digi
    })
    .strict()
    .noUnknown()

```

```

UsersSchema.getAll = object().shape({}).strict().noUnknown()

```

```

UsersSchema.getOne = object()
    .shape({})
    .strict()
    .noUnknown()

```

```

UsersSchema.deleteOne = object()
    .shape({})
    .strict()
    .noUnknown()

```

```
export default UsersSchema
```