**Complete Code Implementation**

**Project Structure**

src/

├── main.ts

├── app.module.ts

├── preferences/

│ ├── preferences.controller.ts

│ ├── preferences.service.ts

│ ├── preferences.module.ts

│ ├── schemas/

│ │ ├── user-preference.schema.ts

│ │ ├── notification-log.schema.ts

├── notifications/

│ ├── notifications.controller.ts

│ ├── notifications.service.ts

│ ├── notifications.module.ts

├── dto/

│ ├── create-preference.dto.ts

│ ├── update-preference.dto.ts

│ ├── send-notification.dto.ts

├── config/

│ ├── swagger.config.ts

│ ├── mongoose.config.ts

**Code Implementation**

**main.ts**

import { NestFactory } from '@nestjs/core';

import { AppModule } from './app.module';

import { SwaggerModule, DocumentBuilder } from '@nestjs/swagger';

async function bootstrap() {

const app = await NestFactory.create(AppModule);

// Swagger Setup

const config = new DocumentBuilder()

.setTitle('User Notification Preferences API')

.setDescription('API to manage user preferences and send notifications')

.setVersion('1.0')

.build();

const document = SwaggerModule.createDocument(app, config);

SwaggerModule.setup('api-docs', app, document);

await app.listen(3000);

}

bootstrap();

**app.module.ts**

import { Module } from '@nestjs/common';

import { MongooseModule } from '@nestjs/mongoose';

import { PreferencesModule } from './preferences/preferences.module';

import { NotificationsModule } from './notifications/notifications.module';

@Module({

imports: [

MongooseModule.forRoot('mongodb://localhost:27017/user-notifications'), // Replace with your DB URL

PreferencesModule,

NotificationsModule,

],

})

export class AppModule {}

**Preferences Module**

**preferences.module.ts**

import { Module } from '@nestjs/common';

import { MongooseModule } from '@nestjs/mongoose';

import { PreferencesController } from './preferences.controller';

import { PreferencesService } from './preferences.service';

import { UserPreference, UserPreferenceSchema } from './schemas/user-preference.schema';

@Module({

imports: [

MongooseModule.forFeature([{ name: UserPreference.name, schema: UserPreferenceSchema }]),

],

controllers: [PreferencesController],

providers: [PreferencesService],

})

export class PreferencesModule {}

**schemas/user-preference.schema.ts**

import { Schema, Prop, SchemaFactory } from '@nestjs/mongoose';

import { Document } from 'mongoose';

export type UserPreferenceDocument = UserPreference & Document;

@Schema({ timestamps: true })

export class UserPreference {

@Prop({ required: true, unique: true })

userId: string;

@Prop({ required: true, unique: true })

email: string;

@Prop({

type: Object,

required: true,

})

preferences: {

marketing: boolean;

newsletter: boolean;

updates: boolean;

frequency: 'daily' | 'weekly' | 'monthly' | 'never';

channels: {

email: boolean;

sms: boolean;

push: boolean;

};

};

@Prop({ required: true })

timezone: string;

}

export const UserPreferenceSchema = SchemaFactory.createForClass(UserPreference);

**preferences.controller.ts**

import { Controller, Post, Get, Patch, Delete, Body, Param } from '@nestjs/common';

import { PreferencesService } from './preferences.service';

import { CreatePreferenceDto } from '../dto/create-preference.dto';

import { UpdatePreferenceDto } from '../dto/update-preference.dto';

@Controller('api/preferences')

export class PreferencesController {

constructor(private readonly preferencesService: PreferencesService) {}

@Post()

async createPreference(@Body() createPreferenceDto: CreatePreferenceDto) {

return this.preferencesService.create(createPreferenceDto);

}

@Get(':userId')

async getPreference(@Param('userId') userId: string) {

return this.preferencesService.findOne(userId);

}

@Patch(':userId')

async updatePreference(

@Param('userId') userId: string,

@Body() updatePreferenceDto: UpdatePreferenceDto,

) {

return this.preferencesService.update(userId, updatePreferenceDto);

}

@Delete(':userId')

async deletePreference(@Param('userId') userId: string) {

return this.preferencesService.remove(userId);

}

}

**preferences.service.ts**

import { Injectable } from '@nestjs/common';

import { InjectModel } from '@nestjs/mongoose';

import { Model } from 'mongoose';

import { UserPreference, UserPreferenceDocument } from './schemas/user-preference.schema';

import { CreatePreferenceDto } from '../dto/create-preference.dto';

import { UpdatePreferenceDto } from '../dto/update-preference.dto';

@Injectable()

export class PreferencesService {

constructor(

@InjectModel(UserPreference.name) private userPreferenceModel: Model<UserPreferenceDocument>,

) {}

async create(createPreferenceDto: CreatePreferenceDto) {

const createdPreference = new this.userPreferenceModel(createPreferenceDto);

return createdPreference.save();

}

async findOne(userId: string) {

return this.userPreferenceModel.findOne({ userId }).exec();

}

async update(userId: string, updatePreferenceDto: UpdatePreferenceDto) {

return this.userPreferenceModel

.findOneAndUpdate({ userId }, updatePreferenceDto, { new: true })

.exec();

}

async remove(userId: string) {

return this.userPreferenceModel.findOneAndDelete({ userId }).exec();

}

}

**Notifications Module**

**notifications.module.ts**

import { Module } from '@nestjs/common';

import { NotificationsController } from './notifications.controller';

import { NotificationsService } from './notifications.service';

@Module({

controllers: [NotificationsController],

providers: [NotificationsService],

})

export class NotificationsModule {}

**notifications.controller.ts**

import { Controller, Post, Body } from '@nestjs/common';

import { NotificationsService } from './notifications.service';

import { SendNotificationDto } from '../dto/send-notification.dto';

@Controller('api/notifications')

export class NotificationsController {

constructor(private readonly notificationsService: NotificationsService) {}

@Post('send')

async sendNotification(@Body() sendNotificationDto: SendNotificationDto) {

return this.notificationsService.send(sendNotificationDto);

}

}

**notifications.service.ts**

import { Injectable } from '@nestjs/common';

import { SendNotificationDto } from '../dto/send-notification.dto';

@Injectable()

export class NotificationsService {

async send(sendNotificationDto: SendNotificationDto) {

// Simulate sending notification

return { status: 'Notification sent', data: sendNotificationDto };

}

}

**DTOs**

**create-preference.dto.ts**

import { IsString, IsEmail, IsBoolean, IsEnum, IsObject } from 'class-validator';

export class CreatePreferenceDto {

@IsString()

userId: string;

@IsEmail()

email: string;

@IsObject()

preferences: {

@IsBoolean() marketing: boolean;

@IsBoolean() newsletter: boolean;

@IsBoolean() updates: boolean;

@IsEnum(['daily', 'weekly', 'monthly', 'never']) frequency: string;

channels: {

@IsBoolean() email: boolean;

@IsBoolean() sms: boolean;

@IsBoolean() push: boolean;

};

};

@IsString()

timezone: string;

}