const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

name: { type: String, required: true },

email: { type: String, required: true, unique: true },

username: { type: String, required: true, unique: true },

phone\_number: { type: String, required: true, unique: true },

user\_type: { type: String, enum: ['Farmer', 'Veterinarians'], required: true },

location: { type: String },

total\_animals: { type: Number, default: 0 },

profile\_picture: { type: String },

bio: { type: String },

dob: { type: Date },

gender: { type: String, enum: ['Male', 'Female', 'Other'] },

verification\_status: { type: Boolean, default: false },

any\_website: { type: String },

facebook: { type: String },

twitter: { type: String },

instagram: { type: String },

total\_posts: { type: Number, default: 0 },

total\_likes: { type: Number, default: 0 },

total\_reposts: { type: Number, default: 0 },

is\_active: { type: Boolean, default: true },

last\_login: { type: Date },

created\_at: { type: Date, default: Date.now },

updated\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('User', userSchema);

const postSchema = new mongoose.Schema({

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

post\_type: { type: String, enum: ['farm\_update', 'disease\_question'], required: true },

title: { type: String },

content: { type: String },

images: [{ type: String }], // Array of image URLs

videos: [{ type: String }], // Array of video URLs

is\_repostable: { type: Boolean, default: true },

like\_count: { type: Number, default: 0 },

repost\_count: { type: Number, default: 0 },

reply\_count: { type: Number, default: 0 },

share\_count: { type: Number, default: 0 },

save\_count: { type: Number, default: 0 },

report\_count: { type: Number, default: 0 },

created\_at: { type: Date, default: Date.now },

updated\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('Post', postSchema);

const likeSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

created\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('Like', likeSchema);

const viewSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

created\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('View', viewSchema);

const replySchema = new mongoose.Schema({

comment\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Comment', required: true },

parent\_reply\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Reply', default: null },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

content: { type: String, required: true },

created\_at: { type: Date, default: Date.now },

updated\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('Reply', replySchema);

const shareSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

created\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('Share', shareSchema);

const questionRaisedSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

created\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('QuestionRaised', questionRaisedSchema);

const repostSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

created\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('Repost', repostSchema);

const savedPostSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

created\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('SavedPost', savedPostSchema);

const downloadedPostSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

created\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('DownloadedPost', downloadedPostSchema);

const commentSchema = new mongoose.Schema({

post\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'Post', required: true },

user\_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

content: { type: String, required: true },

created\_at: { type: Date, default: Date.now },

updated\_at: { type: Date, default: Date.now },

});

module.exports = mongoose.model('Comment', commentSchema);