## 1. User Schema HIGH PRIORITY

```
const userSchema = new mongoose.Schema({
 user_id: { type: String, unique: true, required: true },
 email: { type: String, unique: true, required: true },
 password: { type: String, required: true },
 phone: { type: String, unique: true, required: true },
 verified: { type: Boolean, default: false },
 banned: { type: Boolean, default: false },
 profile_image: { type: String, default: null },
 bio: { type: String, default: null },
 crime_reports: [{ type: mongoose.Schema.Types.ObjectId, ref:
'Crime' }],
});
```

---

## 2. Crime Report Schema HIGH PRIORITY

```
const crimeSchema = new mongoose.Schema({
 crime_id: { type: String, unique: true, required: true },
 title: { type: String, required: true },
 description: { type: String, required: true },
 division: { type: String, required: true },
 district: { type: String, required: true },
 crime_time: { type: Date, required: true },
 images: [{ type: String }], // URLs of uploaded images
 video: { type: String, default: null }, // URL of uploaded
video
 post_time: { type: Date, default: Date.now },
 user_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User',
required: true },
 upvotes: { type: Number, default: 0 },
 downvotes: { type: Number, default: 0 },
 verification_score: { type: Number, default: 0 },
 comments: [{ type: mongoose.Schema.Types.ObjectId, ref:
'Comment' }],
});
```

---

## 3. Comment Schema HIGH PRIORITY

```
const commentSchema = new mongoose.Schema({
 comment_id: { type: String, unique: true, required: true },
 user_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User',
required: true },
 crime_id: { type: mongoose.Schema.Types.ObjectId, ref:
'Crime', required: true },
 comment: { type: String, required: true },
 proof_image: { type: String, default: null },
 proof_video: { type: String, default: null },
 created_at: { type: Date, default: Date.now },
});
```

---

## 4. Heatmap Data Schema

```
const heatmapSchema = new mongoose.Schema({
 location: { type: String, required: true },
 count: { type: Number, required: true },
 date_range: { type: String, required: true },
});
```

---

## 5. Leaderboard Schema

```
const leaderboardSchema = new mongoose.Schema({
 user_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User',
required: true },
 username: { type: String, required: true },
 score: { type: Number, required: true },
 type: { type: String, enum: ['top_contributors',
'most_helpful_comments'], required: true },
});
```

---

## 6. Division and District Schema NOT NECESSARY NOW

```
const divisionDistrictSchema = new mongoose.Schema({
 division_name: { type: String, required: true },
 districts: [{ type: String, required: true }],
});
```

---

## 7. OTP Schema HIGH PRIORITY

```
const otpSchema = new mongoose.Schema({
 phone_number: { type: String, required: true },
 otp: { type: String, required: true },
 expires_at: { type: Date, required: true },
});
```

---

## 8. Token Schema HIGH PRIORITY

```
const tokenSchema = new mongoose.Schema({
 user_id: { type: mongoose.Schema.Types.ObjectId, ref: 'User',
required: true },
 access_token: { type: String, required: true },
 refresh_token: { type: String, required: true },
 expires_at: { type: Date, required: true },
});
```

---

## 9. AI Description Schema HIGH PRIORITY

```
const aiDescriptionSchema = new mongoose.Schema({
 image_url: { type: String, required: true },
 description: { type: String, required: true },
 generated_at: { type: Date, default: Date.now },
});
```

---

## 10. Admin Action Schema

```
const adminActionSchema = new mongoose.Schema({
 admin_id: { type: mongoose.Schema.Types.ObjectId, ref:
'User', required: true },
 action: { type: String, enum: ['ban_user'], required: true },
 target_user_id: { type: mongoose.Schema.Types.ObjectId, ref:
'User', required: true },
 action_taken_at: { type: Date, default: Date.now },
});
```

## Summary of Schemas

1. User Schema - For user registration, login, and profile management.
2. Crime Report Schema - For storing crime reports.
3. Comment Schema - For storing comments on crime reports.
4. Heatmap Data Schema - For storing heatmap data.
5. Leaderboard Schema - For storing leaderboard rankings.
6. Division and District Schema - For storing geographical divisions and districts.
7. OTP Schema - For storing OTPs for verification.
8. Token Schema - For storing access and refresh tokens.
9. AI Description Schema - For storing AI-generated descriptions.
10. Admin Action Schema - For logging admin actions.