Implémentation du Système de Sélection de Modèles AI - ReelGen AI

Date : 26 octobre 2025
Version : 1.0
Status : ✅ Implémenté et testé

🎯 Objectif Atteint

Permettre aux utilisateurs de **choisir les modèles AI** utilisés pour la génération de contenu, avec possibilité d'optimiser selon **3 critères** : Qualité, Coût ou Vitesse.

📊 Résumé de l'Implémentation

✅ Fonctionnalités Ajoutées

1. **Catalogue de modèles** : 9 modèles fal.ai pré-configurés
   - 3 modèles d'images (transformation)
   - 6 modèles vidéo (animation)
2. **Système de préférences utilisateur** : Chaque utilisateur peut sauvegarder ses modèles préférés
3. **Estimation de coût en temps réel** : Calcul automatique avant génération
4. **Interface de sélection** : Panneau dans les paramètres avec tri intelligent
5. **Intégration transparente** : Utilisation automatique des préférences lors de la génération

---

🗂️ Structure de la Base de Données

Nouvelles Tables

`model_preferences`

```
model ModelPreferences {
  id                String   @id @default(cuid())
  userId            String   @unique
  user              User     @relation(fields: [userId], references: [id])

  imageModel        String   @default("fal-ai/flux/dev/image-to-image")
  imageToVideoModel String   @default("fal-ai/luma-dream-machine/image-to-video")

  prioritizeSpeed   Boolean  @default(false)
  prioritizeCost    Boolean  @default(false)
  prioritizeQuality Boolean  @default(true)
}
```

`model_catalog`

```
model ModelCatalog {
  id            String   @id @default(cuid())
  endpoint      String   @unique
  name          String
  category      String   // "image" ou "video"
  provider      String
  pricePerUnit  Float
  priceUnit     String
  qualityRating Int      @default(3) // 1-5 étoiles
  hasAudio      Boolean  @default(false)
  avgSpeed      Int?
  description   String?
  features      Json?
  isActive      Boolean  @default(true)
}
```

🎨 Catalogue de Modèles Implémentés

Modèles Images (Image-to-Image)

Modèle	Prix	Qualité	Vitesse	Description
FLUX.1 [dev]	$0.030/MP	⭐⭐⭐⭐⭐	8s	Excellent rapport qualité/prix (défaut)
FLUX.1 [schnell]	$0.015/MP	⭐⭐⭐⭐	4s	40% moins cher, idéal pour tests
Qwen Image	$0.020/MP	⭐⭐⭐⭐	6s	Excellent pour rendu de texte

Modèles Vidéo (Image-to-Video)

Modèle	Prix	Qualité	Vitesse	Audio	Description
Luma Dream Machine	$0.50/vidéo	⭐⭐⭐⭐⭐	25s	❌	Qualité supérieure (défaut)
Wan 2.5	$0.25/vidéo	⭐⭐⭐⭐	20s	✅	**50% moins cher**, audio natif
Kling 2.5 Turbo Pro	$0.30/vidéo	⭐⭐⭐⭐⭐	18s	❌	Motion fluide, cinématique
LTX-2 Fast	$0.18/vidéo	⭐⭐⭐	12s	❌	**64% moins cher**, très rapide
Seedance 1.0 Pro	$0.62/vidéo	⭐⭐⭐⭐⭐	30s	❌	Mouvement naturel premium
Veo 3.1 (Google)	$0.30/s	⭐⭐⭐⭐⭐	35s	✅	Audio + dialogue, physique réaliste

---

🔧 Architecture Technique

APIs Créées

1. GET /api/models

- Liste tous les modèles d'un catégorie
- Query params : `?category=image|video`
- Filtrage par statut actif
- Tri par qualité et prix

2. GET /api/models/preferences

- Récupère les préférences utilisateur
- Crée des préférences par défaut si inexistantes
- Retourne les modèles sélectionnés

3. PUT /api/models/preferences

- Met à jour les préférences utilisateur
- Validation des données
- Upsert automatique

4. POST /api/models/estimate

- Estime le coût d'une génération
- Params : `{ imageModel, videoModel, variations }`
- Retourne le coût détaillé par étape

Composants UI

1. ModelSelector

`app/dashboard/_components/model-selector.tsx`

- Sélecteur intelligent avec tri automatique
- Affichage du prix et notation en étoiles
- Support du tri par coût/qualité/vitesse
- Chargement asynchrone des modèles

2. ModelPreferencesPanel

`app/dashboard/_components/model-preferences-panel.tsx`

- Panneau complet de configuration
- 3 modes de priorité (Qualité/Coût/Vitesse)
- Estimation de coût en temps réel
- Sauvegarde des préférences

Intégration Backend

Modifications de `lib/media-generator.ts`

```
// Nouvelle fonction : Récupération des préférences
async function getUserModels(userId?: string): Promise<{
  imageModel: string;
  videoModel: string;
}> {
  // Récupère les modèles préférés ou retourne les défauts
}

// Modifié : Utilisation des préférences
export async function generateTransformedImage(options: GenerateImageOptions) {
  const { imageModel } = await getUserModels(options.userId);
  const transformedUrl = await transformImageWithAI(imageUrl, prompt, imageModel);
}

export async function generateAnimatedVideo(options: GenerateVideoOptions) {
  const { videoModel } = await getUserModels(options.userId);
  const videoUrl = await generateVideoFromImage(videoSourceUrl, prompt, duration, vide
oModel);
}
```

Modifications de `lib/fal.ts`

```
// Support de modèles dynamiques
export async function transformImageWithAI(
  imageUrl: string,
  prompt: string,
  modelEndpoint?: string // Nouveau paramètre
): Promise<string> {
  const model = modelEndpoint || 'fal-ai/flux/dev/image-to-image';
  const result = await fal.subscribe(model, { ... });
}

export async function generateVideoFromImage(
  imageUrl: string,
  prompt: string,
  duration: number = 5,
  modelEndpoint?: string // Nouveau paramètre
): Promise<string> {
  const model = modelEndpoint || 'fal-ai/luma-dream-machine/image-to-video';
  const result = await fal.subscribe(model, { ... });
}
```

📈 Scénarios d'Économie Possibles

Configuration Actuelle (Baseline)

- Image : FLUX.1 [dev] ($0.030)
- Vidéo : Luma Dream Machine ($0.50)
- **Coût par génération** : $0.53
- **Projection mensuelle** (100 générations) : $53.00

Configuration Économique (Wan 2.5)

- Image : FLUX.1 [dev] ($0.030)
- Vidéo : **Wan 2.5** ($0.25)
- **Coût par génération** : $0.28 (-47%)
- **Projection mensuelle** : $28.00
- ✅ **Économie** : $25/mois

Configuration Maximale Économie (LTX-2)

- Image : FLUX.1 [schnell] ($0.015)
- Vidéo : **LTX-2 Fast** ($0.18)
- **Coût par génération** : $0.195 (-63%)
- **Projection mensuelle** : $19.50
- ✅ **Économie** : $33.50/mois

Configuration Premium (Veo 3.1)

- Image : FLUX.1 [dev] ($0.030)
- Vidéo : **Veo 3.1** ($0.30/s = ~$1.50 pour 5s)
- **Coût par génération** : $1.53 (+189%)
- **Projection mensuelle** : $153.00

- ➕ **Valeur ajoutée** : Audio natif, dialogue, physique réaliste

---

🎯 Utilisation pour l'Utilisateur

Accès aux Paramètres

1. **Connexion** → Dashboard
2. **Navigation** → Onglet "Settings" (⚙️)
3. **Section** → "Paramètres des Modèles AI"

Sélection de Modèles

Étape 1 : Choisir la Priorité

- **Qualité** : Meilleurs modèles (Luma, Veo 3.1)
- **Économie** : Modèles optimisés coût (Wan 2.5, LTX-2)
- **Vitesse** : Modèles les plus rapides (schnell, LTX-2)

Étape 2 : Sélectionner les Modèles

- **Modèle Image** : Choix parmi 3 options
- **Modèle Vidéo** : Choix parmi 6 options
- Affichage du prix et notation pour chaque modèle

Étape 3 : Voir l'Estimation

- **Coût par génération** (1 image + 3 vidéos)
- **Détail** par composant (image, vidéos)
- **Total** avec projection

Étape 4 : Enregistrer

- **Bouton "Enregistrer"**
- Confirmation du succès
- Application immédiate

Génération de Contenu

Processus automatique :
1. Upload d'une image
2. Saisie des prompts
3. ✨ **Utilisation automatique des modèles préférés**
4. Génération avec le coût optimisé

🔄 Intégration avec le Système Existant

Routes Modifiées

`/api/upload`

```
processJobAsync(jobId, imageUrl, imagePrompt, videoPrompt, session.user.id);
// Ajout du userId pour récupérer les préférences
```

`/api/jobs/[id]/generate-variations`

```
const videoUrl = await generateAnimatedVideo({
  imageUrl: originalJob.transformedImageUrl!,
  prompt: originalJob.videoPrompt!,
  duration: 5,
  userId: session.user.id, // Utilise les préférences utilisateur
});
```

`/api/jobs/[id]/regenerate`

```
const newVideoUrl = await generateAnimatedVideo({
  imageUrl: originalJob.transformedImageUrl,
  prompt: originalJob.videoPrompt,
  duration: 5,
  userId: session.user.id, // Utilise les préférences utilisateur
});
```

📦 Scripts Utiles

Seed du Catalogue de Modèles

```
cd nextjs_space
npx tsx --require dotenv/config scripts/seed-models.ts
```

Peuple la base de données avec les 9 modèles configurés.

Vérifier les Préférences d'un Utilisateur

```
SELECT * FROM model_preferences WHERE "userId" = 'USER_ID';
```

Lister Tous les Modèles Actifs

```
SELECT endpoint, name, category, "pricePerUnit", "qualityRating"
FROM model_catalog
WHERE "isActive" = true
ORDER BY category, "qualityRating" DESC;
```

🎉 Résultats et Bénéfices

Pour l'Utilisateur

✅ **Contrôle total** sur les modèles utilisés
✅ **Optimisation budgétaire** jusqu'à -63% de coût
✅ **Transparence** des prix avant génération
✅ **Flexibilité** selon les besoins (qualité vs coût)
✅ **Simplicité** d'utilisation (sauvegarde automatique)

Pour le Système

✅ **Architecture extensible** pour futurs modèles
✅ **Catalogue centralisé** facile à mettre à jour
✅ **Préférences persistantes** par utilisateur
✅ **Intégration transparente** avec le workflow existant
✅ **Performance** (récupération une seule fois par génération)

📝 Prochaines Étapes Recommandées

Court Terme

1. ⚡ **Monitorer l'utilisation** des différents modèles
2. 📊 **Collecter les retours** utilisateurs sur la qualité
3. 💰 **Analyser l'impact** sur le budget moyen

Moyen Terme

1. 🎨 **Ajouter plus de modèles** (Runway, Stable Video, etc.)
2. 🧪 **Tests A/B** automatiques sur les modèles
3. 📈 **Statistiques détaillées** de consommation par modèle

Long Terme

1. 🤖 **IA de recommandation** selon le type de contenu
2. 🔄 **Optimisation automatique** selon le budget restant
3. 🎯 **Profils prédéfinis** (Économique, Équilibré, Premium)

🛠️ Dépannage

Problème : Modèles non affichés

Solution :

```
cd nextjs_space
npx tsx --require dotenv/config scripts/seed-models.ts
```

Problème : Préférences non sauvegardées

Vérifier :
1. Session utilisateur valide
2. userId présent dans la requête
3. Logs de l'API `/api/models/preferences`

Problème : Estimation incorrecte

Vérifier :
1. Prix des modèles dans `model_catalog`
2. Calcul dans `/api/models/estimate`
3. Nombre de variations par défaut (3)

📚 Ressources

- **Documentation fal.ai** : https://docs.fal.ai/
- **Catalogue complet** : `/instagram_content_generator/ANALYSE_OPTIMISATION_FALAI.md`
- **Pricing officiel** : https://fal.ai/pricing
- **Code source** : `/nextjs_space/app/dashboard/_components/model-*`

Implémentation réalisée avec succès ✅
Checkpoint créé : “Système sélection modèles AI”
Documentation : Complète et à jour
Prêt pour production 🚀