// Types for post tracking

interface LinkedInPost {

id: string;

postUrl: string;

content: string;

postedAt: Date;

performance?: PostPerformance[];

}

interface PostPerformance {

impressions: number;

reactions: number;

comments: number;

shares: number;

clicks: number;

measuredAt: Date;

}

// Post creation and tracking

class LinkedInPostManager {

async createPost(content: string): Promise<LinkedInPost> {

// Create post on LinkedIn

const linkedInResponse = await this.linkedInAPI.createPost(content);

// Store reference

const post = await this.db.from('linkedin\_posts').insert({

linkedin\_post\_id: linkedInResponse.id,

post\_url: linkedInResponse.postUrl,

post\_content: content

});

return post;

}

async updatePerformance(postId: string, metrics: PostPerformance): Promise<void> {

await this.db.from('post\_performance').insert({

post\_id: postId,

...metrics,

measured\_at: new Date()

});

}

async getPostAnalytics(postId: string): Promise<PostPerformance[]> {

return await this.db

.from('post\_performance')

.select('\*')

.where('post\_id', postId)

.orderBy('measured\_at', 'asc');

}

}